Universidad Autónoma de Aguascalientes

Centro de Ciencias Básicas

Maestría en Ciencias Exactas, Sistemas y de la Información

Redes de Computadoras

DESARROLLO DE CORTAFUEGOS DE APLICACIÓN EN MODO TRANSPARENTE CON SOFTWARE LIBRE (GNU/LINUX) Y EVALUACIÓN DE SU DESEMPEÑO CON: SOLUCIÓN EN WINDOWS® Y SOLUCIÓN EN SISTEMA INTEGRADO.

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias.

Presenta:
I.E.S.C.D. Américo Cuauhtémoc Calzada de Luna.

Dirigido por:
Dr. Juan Manuel Gómez Reynoso.

Comité de Revisión:
M.I.T.C. Francisco Gutiérrez Nájera.
M.I.T.C. Arturo Elías Ramírez.

Aguascalientes Ags. Junio de 2009.

México.

Por este conducto autorizamos al:

I.E.S.C.D. Américo Cuauhtémoc Calzada de Luna

La impresión de su documento final de Tesis, ya que cumple con los requisitos de contenido y forma exigidos en la Universidad Autónoma de Aguascalientes.

Director de Tesis

Juan Manuel Gómez Reynoso, Ph.D.

Sinodal

M.I.T.C. Francisco Gutiérrez Nájera.

Sinodal

M.I.T.C. Arturo Elías Ramírez

DEDICATORIAS.

A mi esposa Cecilia por entenderme y apoyarme siempre de una manera incondicional, gracias por ser el amor de mi vida.

A mis hijos por ser mi inspiración y mi razón de ser.

A mis padres por haberme inculcado sus valores y haber hecho de mí una persona de bien.

A mi tutor Juan Manuel por haber creído en mí.

RESUMEN:

Los cortafuegos que bloquean en base a puertos han quedado obsoletos debido a la gran cantidad de aplicaciones que utilizan los mismos puertos que las aplicaciones criticas, basándose en ello estas nuevas aplicaciones consumen un gran ancho de banda además que algunas de ellas representan riesgos de seguridad para la red interna.

Se presenta un desarrollo o procedimiento para crear un cortafuegos de aplicación en modo transparente y se mide su desempeño contra dos soluciones similares. En el desarrollo de cortafuegos que se propuso con GNU/Linux solo se instalaron 2 servicios indispensables para crearlo los cuales fueron: IPTables con el modulo l7-filter como cortafuegos y brigde-utils para convertir el cortafuegos en un cortafuegos transparente.

El desempeño se estableció en base al menor uso de CPU y menor uso de memoria RAM así como al jitter de un flujo multimedia. Fue un diseño experimental con 3 escenarios, donde se utilizó la herramienta *Generador de Tráfico de Internet Distribuido* para generar 3 flujos de tráfico y medir el jitter del flujo multimedia. Se analizaron los 3 escenarios, primero el cortafuegos en Windows creado con el software WebSenseExpress®, después fue el turno del cortafuegos creado con GNU/Linux y finalmente se analizó la solución en un Sistema Integrado que fue PacketShaper®.

Las hipótesis fueron comprobadas con análisis de varianza y análisis Post Hoc. El mejor desempeño en CPU lo tuvo la solución en sistema integrado, el mejor desempeño en memoria RAM lo tuvo el cortafuegos con GNU/Linux y finalmente el mejor desempeño en jitter fue también para el cortafuegos GNU/Linux aunque sin una diferencia significativa.

INDICE DE CONTENIDO

Página

INDICE DE FIGURAS

Página

INDICE DE TABLAS

Página

I. INTRODUCCION

Los cortafuegos de aplicación son indispensables en las redes actuales debido a la gran cantidad de aplicaciones nuevas que consumen gran parte del ancho de banda y que no funcionan en un puerto en especifico o utilizan puertos usados por aplicaciones críticas, en este nuevo escenario los cortafuegos tradicionales que trabajan bloqueando puertos han quedado obsoletos ya que resulta imposible bloquear los puertos de aplicaciones críticas.

Un cortafuegos de aplicación puede identificar los protocolos usados por las aplicaciones indistintamente del puerto por el que sean transmitidas, permitiendo con esto poder bloquearlos.

Un cortafuegos transparente es aquel que es invisible por los usuarios en la red, este tipo de cortafuegos posee grandes ventajas contra un cortafuegos ordinario, por ejemplo son inmunes a los ataques de denegación de servicio en base a IP, su instalación es inmediata ya que no interfiere en la estructura de la red, otra gran ventaja es que los cortafuegos transparentes en puente son más rápidos para procesar paquetes que los cortafuegos en modo encaminador ya que en si mismo un puente es más rápido que un encaminador investigaciones han demostrado esto.

Existen muy pocos cortafuegos, ya sea en software comercial o de código libre, así como en sistema integrado que incluya las dos cualidades mencionadas, es decir, que sea un cortafuegos de aplicación y además sea transparente.

En la presente investigación se desarrolló un cortafuegos capaz de identificar protocolos. Adicionalmente se configuró dicho cortafuegos en puente para hacerlo transparente en la red. Este cortafuegos fue instalado y configurado usando un sistema operativo GNU/Linux. Se tomó esta decisión debido a que

este sistema operativo permite aprovechar el gran crecimiento y expansión que ha tenido el software GNU dentro del campo de la tecnología.

La presente investigación tuvo dos objetivos:

El primer objetivo fue mostrar un desarrollo o procedimiento para crear el cortafuegos de aplicación en modo transparente usando GNU/Linux. Dicho procedimiento podrá mejorarse o incluso complementarse con modelado de tráfico o calidad en el servicio.

El segundo objetivo fue probar el desempeño del cortafuegos creado contra dos soluciones similares, Websense; solución en software utilizando el sistema operativo Windows y PacketShaper; solución en un sistema integrado. El desempeño se medió en base a tres parámetros: 1) menor consumo de CPU, 2) menor consumo de memoria RAM y 3) menor jitter en un flujo multimedia.

Para probar las hipótesis se realizó un diseño factorial que fue comprobado con análisis de varianza y análisis post hoc. El mejor desempeño en CPU lo tuvo cortafuegos en sistema integrado, el mejor desempeño en memoria RAM lo tuvo el cortafuegos en GNU/Linux y por último el mejor desempeño en el jitter fue también para el cortafuegos en GNU/Linux pero sin una diferencia significativa con la del sistema integrado.

Se deja como trabajo futuro la posibilidad de mejorar el cortafuegos GNU/Linux ya que una vez identificado el protocolo con la herramienta L7-Filter es posible realizar acciones adicionales al bloqueo como lo son el modelado de tráfico o la contabilidad de dichos protocolos.

II MARCO TEÓRICO

1. Historia de los cortafuegos.

La idea de un muro para dejar fuera intrusos data de hace miles de años. Por ejemplo, hace dos mil años los chinos construyeron la gran muralla para protegerse de las tribus del norte. El término cortafuegos fue usado por Lightoler en 1764 para describir los muros que separaban las partes de una construcción de aquellas que tenían fuego. Estas barreras físicas impedían o retrasaban la difusión de fuego por todo el edificio produciendo ahorro de vidas y bienes. Un uso relacionado con los términos se plantea en relación con los trenes de vapor, tal como se describe por Schneier (2000):

> *"El carbón potencia los trenes, habiendo un gran horno en la sala de máquinas, junto con una Pila de carbón. El ingeniero palea el carbón al motor. Este proceso crea polvo de carbón, que es altamente flamable. Ocasionalmente el polvo de carbón se encendía, causando un incendio en la maquina, que algunas veces se difundía a los carros de los pasajeros, como la muerte de pasajeros reducía las ganancias económicas, maquinas de tren se construyeron con paredes de hierro justo detrás del compartimiento del motor. Este detenía la difusión del incendio a los carros de los pasajeros, pero no protegía al ingeniero de entre el montón de carbón y el horno".*

Los predecesores a los cortafuegos para la seguridad en red fueron los dispositivos llamados encaminadores usados al final de los ochentas para separar redes. Un error en la configuración de un segmento de red causaba problemas pero solo de ese lado gracias a la acción del encaminador para separar la red. De manera similar, los protocolos de una red que usan demasiados paquetes de difusión en su configuración no afectan el ancho de banda de las otras redes (Avolio 1999; Schneier 2000).

Para los propósitos de la presente investigación, definiremos cortafuegos como un dispositivo o colección de dispositivos que puede estar entre dos redes o ser parte de una red, tomando en cuenta los siguientes criterios:

- Todo el tráfico entre las dos redes debe pasar a través del cortafuegos;
- Un cortafuegos es un dispositivo que le permite a cierto tipo de tráfico pasar, mientras bloquea otro tráfico;
- Un cortafuegos es un dispositivo de red que refuerza las políticas de seguridad de la organización (Ingham and Forrest 2002).

Adicionalmente, es deseable que contenga los siguientes criterios de seguridad:

- Capaz de ser auditado y contabilizado.
- Poder ser monitorizado en sus recursos.
- No tener cuentas de usuario o acceso directo a usuarios.
- Ser a prueba de fallas de seguridad. Si falla, el sistema protegido es aún seguro, ya que no se pasará tráfico a través del cortafuegos.

De lo anterior, se puede resumir el término cortafuegos como un dispositivo o colección de dispositivos que separa a sus usuarios de ambientes externos potencialmente peligrosos como lo es Internet (Schneier 2000). Un cortafuegos esta diseñado para prevenir o disminuir la difusión de eventos peligrosos.

1.1 Filtrado de Paquetes en Modo encaminador.

Según Zwicky (2000), los sistemas de filtrado de paquetes encaminan a estos entre anfitriones internos y externos, pero lo hacen de manera selectiva. Ellos permiten o bloquean ciertos tipos de paquetes en una manera que refleja la propia política de seguridad de un sitio tal como se muestra en la Figura 1. A este tipo de encaminador utilizado en un cortafuegos de filtrado de paquetes se le conoce como *encaminador selectivo.*

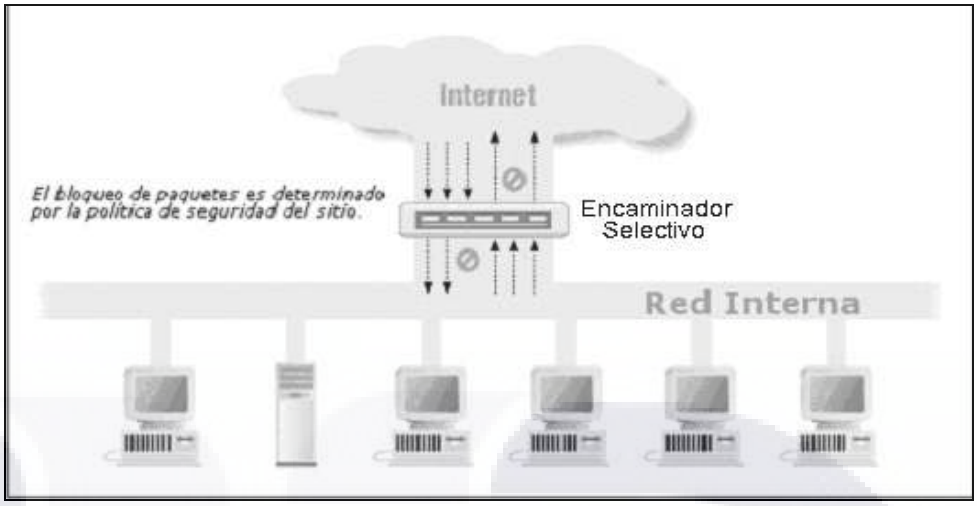


Figura 1: El uso de un encaminador selectivo para hacer filtrado de paquetes (Adaptado de Zwicky, Cooper et al. 2000).

Cada paquete tiene cabeceras que contienen información. La información principal es:

- Dirección IP fuente
- Dirección IP de destino
- Protocolo, si el paquete es un Protocolo de Control de Transmisión (TCP), User Datagram Protocol (UDP), o Protocolo de Mensajes de Control de Internet (ICMP).
- Puerto de origen TCP o UDP
- Puerto de destino TCP o UDP
- Tipo de mensaje ICMP

Además, el encaminador puede determinar información sobre el paquete que no se refleja en la cabecera del paquete, tales como:

- La interfaz de red por la que el paquete llega.
- La interfaz de red por la que el paquete se enviará.

El hecho de que los servidores para un particular servicio de Internet residen en determinados números de puerto permite al encaminador bloquear o permitir ciertos tipos de conexiones con sólo especificar el número de puerto apropiado (por ejemplo, el puerto TCP 23 para conexiones Telnet) en el conjunto de reglas especificadas para filtrado de paquetes.

Para entender cómo funciona el filtrado de paquetes, es importante entender a diferencia entre un encaminador ordinario y un encaminador selectivo. Un encaminador ordinario simplemente mira la dirección de destino de cada paquete y elige el mejor modo que sabe que para enviar paquetes hacia ese destino. La decisión acerca de cómo manejar el paquete se basa únicamente en su destino. Existen dos posibilidades: el encaminador sabe cómo enviar el paquete hacia su destino, y lo hace, o el encaminador no sabe cómo enviar el paquete hacia su destino, y devuelve el paquete, a través de un mensaje ICMP "destino inalcanzable" a su fuente.

Por el contrario, un encaminador selectivo, trata más a fondo los paquetes. Además de determinar si *puede* o no un paquete ser encaminado hacia su destino, este también determina si se debería o no encaminar. La decisión está determinada por la política de seguridad del sitio.

Aunque es posible que sólo un encaminador selectivo sea instalado entre una red interna e Internet, -como se mostró en la Figura 1 - esto impone una enorme responsabilidad sobre él. Este dispositivo no sólo es necesario para desempeñar todas las rutas y toma de decisiones en el encaminamiento, el inconveniente es que es el único sistema de protección. Si su seguridad falla (o se desmorona bajo ataque), la red interna está expuesta. Además, dicho dispositivo no puede modificar los servicios y puede permitir o negar un servicio, pero no puede proteger operaciones individuales dentro de uno. Si un servicio deseado tiene operaciones inseguras, o si el servicio se presta normalmente por un servidor inseguro, el filtrado de paquetes por sí solo no puede protegerlo.

Como se observó, el filtrado de paquetes en modo encaminador presenta grandes deficiencias para proteger una red, es debido a esto que evolucionó hacia otras arquitecturas que proporcionan seguridad adicional en la implementación de cortafuegos.

2. Arquitecturas de cortafuegos.

En esta sección se describen las diferentes maneras de funcionamiento de los cortafuegos tipo encaminador.

2.1 Arquitectura Anfitrión Dual-Homed.

De acuerdo a Zwicky (2000), una arquitectura *anfitrión dual-homed* se construye en base a una computadora que tiene al menos dos interfaces de red. Esta arquitectura podría actuar como un encaminador entre las redes de estas interfaces de red que posee, encaminando paquetes IP de una red a otra. Sin embargo, para aplicar esta arquitectura de cortafuegos es necesario desactivar esta función de encaminamiento. Así, paquetes IP de una red (por ejemplo, Internet) no son encaminadas directamente a la otra red (por ejemplo, la red interna). Los sistemas dentro del cortafuegos pueden comunicarse con el *anfitrión* dual-homed, y sistemas fuera del cortafuegos (en la Internet) pueden comunicarse con el *anfitrión* dual-homed, pero estos sistemas no pueden comunicarse directamente entre sí. El tráfico IP entre ellos es completamente Bloqueado.

La arquitectura de red para un cortafuegos *anfitrión* dual-homed es bastante simple: El cortafuegos se coloca entre Internet y la red interna como lo muestra la Figura 2.

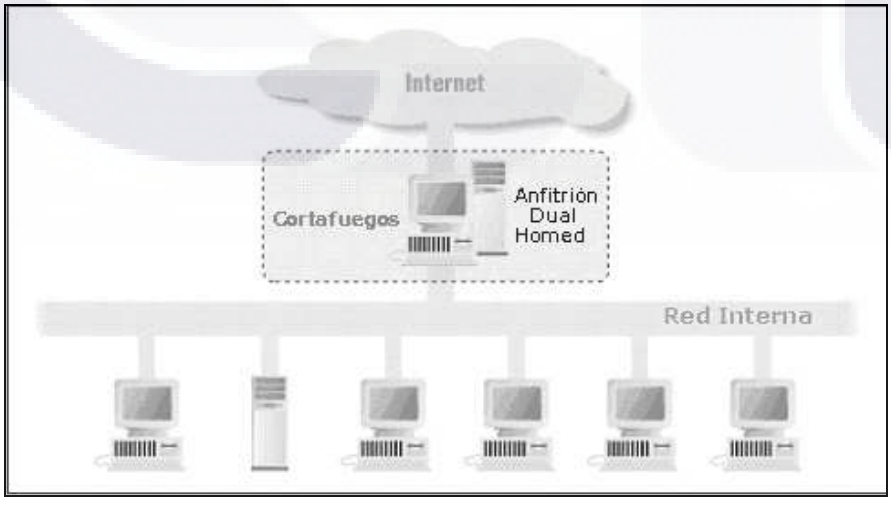


Figura 2: Arquitectura Anfitrión Dual-Homed (Adaptado de Zwicky, Cooper et al. 2000).

Esta arquitectura puede proporcionar un gran nivel de control. Si no está permitiendo que los paquetes IP puedan ir entre las redes externas e internas, puede estar seguro de que cualquier paquete de la red interna que tenga un origen externo es evidencia de algún tipo de problema de seguridad.

Un *anfitrión* dual-homed sólo puede proporcionar servicios de tipo Proxy, es decir, obliga a que los usuarios tengan que autenticarse directamente dentro del *anfitrión* dual-homed. Estas cuentas de usuarios almacenadas en el anfitrión bastión representan por sí mismas un problema de seguridad. Además, la mayoría de los usuarios les resulta inconveniente utilizar un *anfitrión* dual-homed de autenticación para acceder a los recursos.

2.2 Arquitectura Anfitrión Selectivo.

Considerando que una arquitectura *anfitrión* dual-homed proporciona una serie de servicios hacia múltiples redes (pero sin encaminamiento de paquetes), una arquitectura de anfitrión selectivo (ver Figura 3) proporciona una serie de servicios solo atribuidos a la red interna, usando un encaminador. En esta arquitectura, lo principal es la seguridad proporcionada por el filtrado de paquetes. Por ejemplo, el filtrado de paquetes es lo que impide que las personas vean alrededor de los servidores Proxy para hacer conexiones directas.

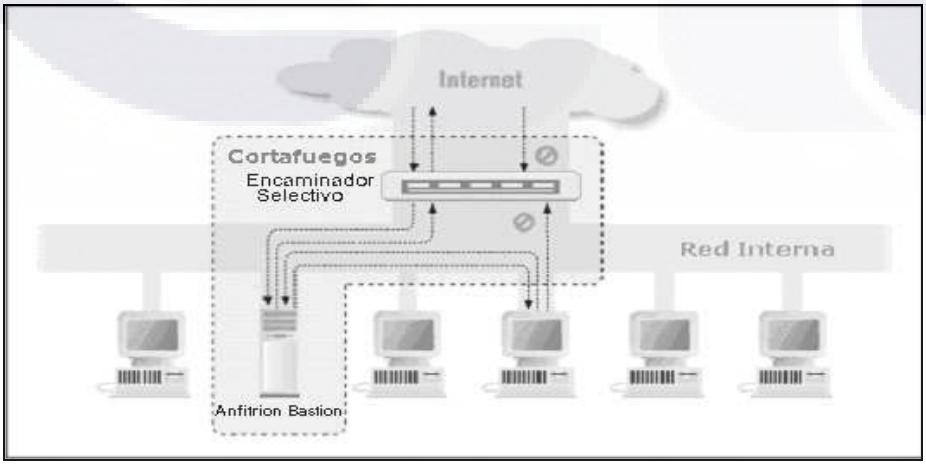


Figura 3: Arquitectura anfitrión selectivo *(Adaptado de Zwicky, Cooper et al. 2000)*.

El anfitrión bastión se encuentra en la red interna. El filtrado de paquetes en el encaminador selectivo está configurado de tal manera que el anfitrión bastión es el único sistema en la red interna que puede abrir conexiones directamente a Internet (por ejemplo, para entregar mensajes). Entonces, sólo ciertos tipos de conexiones son permitidas. Cualquier sistema externo que intente acceder a los sistemas internos o de los servicios tendrá que conectarse a este anfitrión. El anfitrión bastión, por lo tanto, necesita mantener un alto nivel de seguridad.

El filtrado de paquetes le permite también al anfitrión bastión abrir conexiones (lo que es permitido será determinado por su política de seguridad) para el mundo exterior.

El filtrado de paquetes en el encaminador selectivo puede hacer al menos alguno de lo siguiente:

- Permitir que otros anfitriones internos puedan abrir conexiones a servidores en Internet para ciertos servicios.
- Inhabilitar todas las conexiones internas de anfitriones (obligándolos a utilizar servicios a través de Proxy en el anfitrión bastión).

2.3 Arquitectura Subred Selectiva.

La arquitectura de subred selectiva (ver Figura 4) añade un nivel adicional de seguridad a la arquitectura de anfitrión selectivo añadiendo un perímetro de la red que aísla el interior de la red de Internet.

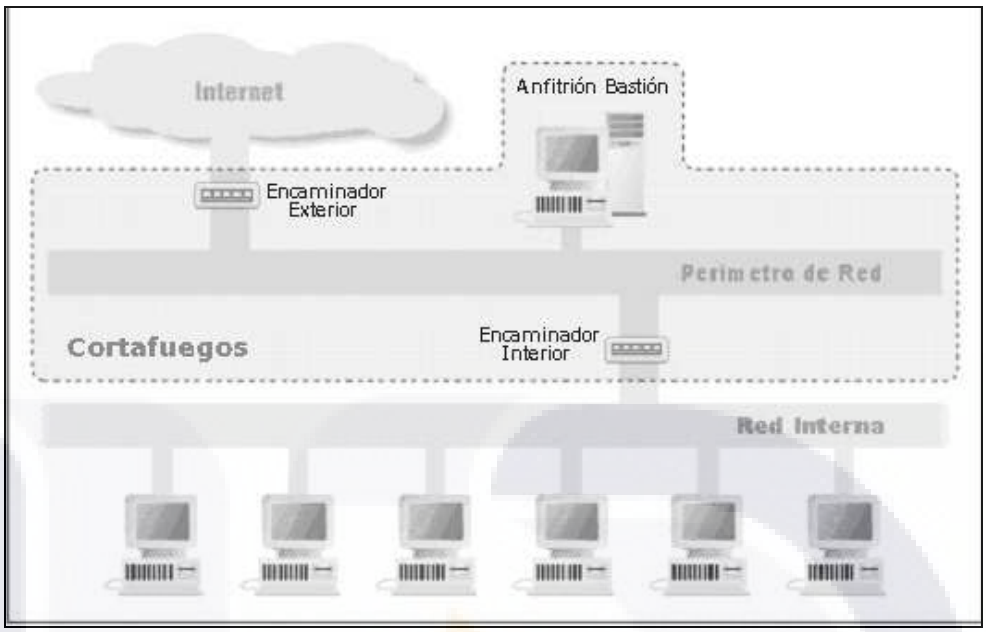


Figura 4: Arquitectura subred selectiva (Adaptado de Zwicky, Cooper et al. 2000).

En la arquitectura más simple de subred selectiva hay dos encaminadores selectivos, cada uno de ellos conectado a la red de perímetro. Uno se encuentra entre la red de perímetro y la red interna, y el otro se sitúa entre el perímetro de la red y la red exterior normalmente Internet. Para penetrar en la red interna con este tipo de arquitectura, un atacante tendría que superar los dos encaminadores. Incluso si el atacante de alguna manera rompió el anfitrión bastión, todavía tiene que superar el encaminador de interior. No existe un único punto vulnerable que comprometa la red interna.

3. La necesidad de cortafuegos.

En años recientes, Internet era soportada por una relativamente pequeña comunidad de usuarios abiertos a la colaboración y comunicación. Esta visión fue desafiada por el gusano Morris (Spafford 1988; Rochlis 1989; Denning 1989.; Spafford 1991). Sin embargo, aun sin este gusano los ataques e intrusiones en esa misma época fueron aumentando. Stoll (1988) descubre a un espía alemán dentro de su sistema. Dichos incidentes marcaron el fin de la época abierta de Internet.

Tomando como referencia los anteriores eventos queda muy claro el hecho de que no se puede confiar en todos los usuarios. Cuando se interconectan redes, existen diferentes niveles de confianza para los diferentes niveles de interconexión, en ese caso significa que una organización cree que tanto el software como sus usuarios no son maliciosos.

Los cortafuegos pueden ser usados para reforzar el ámbito de confianza, los cuales son implantados por una variedad de razones. Algunas de ellas se detallan en las siguientes secciones.

3.1 Problemas de Seguridad en Sistemas Operativos.

Los sistemas operativos tienen una historia de configuraciones inseguras. Por ejemplo, Windows 95 y Windows 98 fueron ampliamente distribuidos con la compartición de archivos activada por defecto. Una serie de virus explotaron esta vulnerabilidad (CERT 2000b; CERT 2000c).

Un segundo ejemplo son las versiones 6.2 y 7.0 de Red Hat Linux. Estas versiones permitían 3 exploits remotos cuando se usaban con la configuración por default *(CERT 2001a)*. Si una computadora personal (PC) en el interior es comprometida, las computadoras restantes son igualmente vulnerables (Muffett 1995).

Generalmente las primeras versiones de cualquier sistema operativo posee vulnerabilidades en sus configuraciones o servicios instalados, estos problemas de seguridad son corregidos en parches o "*fixes"* que son ofrecidos posteriormente a los usuarios de dichos sistemas.

3.2 Prevención de Acceso a Información.

Otro ejemplo de protección a una red es el uso de cortafuegos nacionales. Por ejemplo China, este cortafuegos existe no solo para protegerlos de ataques

externos, sino también para (intentar) limitar las actividades de sus usuarios en Internet (Ingham and Forrest 2002).

Una idea similar es el uso de filtros obligatorios en los Estados Unidos por la "*Children's Internet Protection Act* (CHIPA)". Esta ley obliga a escuelas y librerías que reciban apoyo federal pongan ciertos filtros para todo el contenido web (Ingham and Forrest 2002). Este tipo de filtros esta generalmente enfocado a principalmente categorías como: pornografía, violencia, racismo, etc.

3.3 Prevención de fuga de información.

Ya que todo el tráfico generado en la red debe pasar a través del cortafuegos, este puede usarse para reducir la fuga de información, lo que a su vez puede provocar espionaje industrial o robo de información (Ranum 1992).

El espionaje industrial es la obtención ilícita de información relativa a la investigación, desarrollo y fabricación de prototipos, mediante las cuales las empresas pretenden adelantarse a sus competidores en la puesta en el mercado de un producto novedoso. La creciente reducción de los plazos transcurridos entre la idea novedosa y la puesta en el mercado del producto, así como la cada día mayor obsolescencia de los productos de las nuevas tecnologías, hacen que estos sectores industriales sean el caldo de cultivo ideal para este tipo de actividades ilícitas (Wikipedia.Espionaje 2008).

El robo de identidad es uno de los delitos de más rápido crecimiento en el mundo. Cada cuatro segundos una identidad es robada y alrededor de 10 millones de estadunidenses han resultado afectados, los perjudicados tardan unas 600 horas en librarse de esta pesadilla y varios años en recuperar su buen nombre e historial crediticio (UNAM 2007).

El robo de identidad permite al delincuente abrir cuentas de banco, obtener tarjetas de crédito y teléfonos celulares, arrendar autos e inclusive departamentos a nombre de la víctima sin que ésta se entere (UNAM 2007).

Un informe reciente de Symantec, una compañía de seguridad en la Internet, dice que en Estados Unidos se encuentra más de la mitad de los "servidores de la economía ilegal" utilizados para vender información confidencial y datos personales interceptados en la red. Este tráfico ilícito es un indicio de que, en cierta medida, los delincuentes han empezado a cejar en sus intentos de invadir las redes de los bancos y robar las bases de datos de sus clientes. A medida que las empresas de servicios financieros han reforzado sus sistemas de seguridad, los ladrones han empezado a acechar las cuentas bancarias y las tarjetas de crédito de usuarios individuales (UNAM 2007).

Symantec calcula que en el segundo semestre de 2006 alrededor de 6 millones de computadoras en todo el mundo fueron infectadas por *bots* (programas automatizados que son diseñados y propagados con fines ilícitos), un incremento de 29% en comparación con los seis meses previos. Cuatro de cada cinco de esos procesadores habían sido atacados por programas espías conocidos como troyanos, los cuales extraen información confidencial al registrar teclas oprimidas y sitios de Internet visitados. Otros usuarios confiados fueron enviados a sitios falsos donde los engañaron para sacarles información confidencial (UNAM 2007).

3.4 Reforzar Políticas.

Los cortafuegos son una parte de todo el conjunto de políticas de seguridad, ya que refuerzan las políticas del tráfico de red entrante y saliente. Estas políticas pueden limitar las aplicaciones usadas, así como la conexión de equipos remotos y/o los anchos de banda permitidos, (Hambridge 1993; Treese 1993), además de protegernos contra ataques a bajo nivel tales como: escaneo de Puertos, IP Spoofing e Intercepción de Paquetes (Zwicky, Cooper et al. 2000).

3.5 Auditar.

Si existiera una rotura en la red, la cual no incluye al cortafuegos, un rastreo por las bitácoras del este puede ayudar a determinar lo ocurrido. (Ingham and Forrest 2002). La técnica para hacer lo anterior se le denomina Sistema de Detección de Intrusos (IDS por sus siglas en inglés).

Un cortafuegos resulta bastante útil como primer esquema de detección de intrusos y respuesta automática ante ataques. Esta respuesta será habitualmente el bloqueo de la dirección atacante en el propio *cortafuegos* (Iris 2008).

Para decidir qué tipos de ataques detectar y bloquear en el cortafuegos, debemos pensar con qué información trabaja habitualmente el cortafuegos. Este lo hará al menos con los cinco elementos que definen una conexión bajo la pila Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP): 1) dirección origen, 2) dirección destino, 3) puerto origen, 4) puerto destino y 5) protocolo. De estos cinco, quizás los dos menos importantes (de cara a detectar ataques) son quizás el protocolo utilizado y el puerto origen de la conexión; por tanto, son los otros tres elementos los que nos ayudarán en la constitución de nuestro IDS y los que nos facilitarán el poder lanzar una respuesta automática contra el atacante (Iris 2008).

Conociendo las direcciones origen y destino así como el puerto destino de una conexión se pueden detectar cierto tipo de ataques. Por ejemplo, el escaneo de puertos, tanto horizontales como verticales, que se lanzan contra nuestros sistemas. Northcutt (1999) explica la técnica de detección de estos ataques , la cual está basada por el momento en comprobar X eventos de interés dentro de una ventana de tiempo Y. Así, podemos analizar en nuestro cortafuegos cuándo una misma dirección origen accede a un determinado puerto de varios destinos en menos de un cierto tiempo umbral (escaneo horizontal) o cuando accede a diferentes puertos bien conocidos de un mismo sistema también en menos de ese tiempo umbral (escaneo vertical) (Iris 2008) .

Una técnica alternativa que con frecuencia suele ser utilizada con bastante efectividad para detectar escaneos verticales consiste en vigilar del acceso a determinados puertos de los sistemas protegidos por el *cortafuegos*, acceso que con toda probabilidad representará un intento de violación de nuestras políticas de seguridad (Iris 2008) .

4. Modelado de 7 Capas de OSI de la ISO.

De acuerdo a Tanenbaum (2003), el modelo de referencia para la interconexión de sistemas abiertos *Open Systems Interconnection (*OSI*),* fue aprobado por el organismo internacional *International Standards Organization (*ISO*)* en 1984, bajo la norma ISO 7498, después de varios años de arduo trabajo.

El modelo de referencia OSI proporciona una arquitectura de 7 niveles, alrededor de los cuales se pueden diseñar protocolos específicos que permitan a diferentes usuarios comunicarse abiertamente. También se tomó en cuenta para el desarrollo del modelo OSI, que cada nivel debe contar con ciertas premisas, las cuales son las siguientes:

- Cada nivel realiza tareas únicas y específicas y debe ser creado cuando se necesite un grado diferente de abstracción.
- Todo nivel debe tener conocimiento de los niveles inmediatamente adyacentes y sólo de éstos.
- Todo nivel debe servirse de los servicios del nivel anterior, a la vez que los debe de prestar al superior.
- Los servicios de un nivel determinado son independientes de su implantación práctica.
- Los límites de cada nivel se deben seleccionar, teniendo en cuenta que minimicen el flujo de información a través de las interfaces establecidas.

La siguiente figura muestra el ordenamiento y funciones de las capas de acuerdo a lo mencionado.

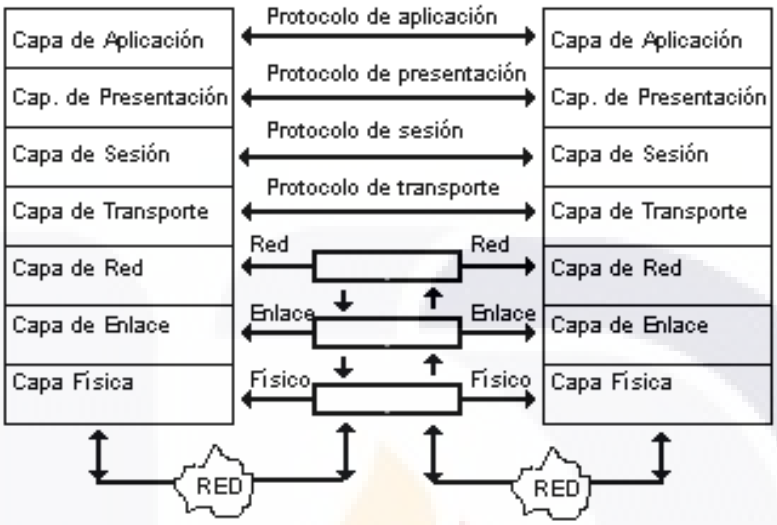


Figura 5: Ordenamiento y funciones del modelo 0SI *(Adaptado de Tanenbaum 2003)*.

En las siguientes secciones se detalla cada una de las capas del modelo OSI.

4.1 Capa Física.

De acuerdo a Tanenbaum (2003), el nivel físico es el encargado, primordialmente, de la transmisión de los bits de datos (0s ó 1s) a través de los circuitos de comunicaciones. El propósito principal de este nivel es definir las reglas para garantizar que cuando la computadora emisora transmita el bit “1”, la computadora receptora verifique que un “1” fue recibido y no un “0”. Este es el nivel de comunicación física de circuitos.

Adicionalmente, esta capa provee los medios mecánicos, eléctricos, funcionales y de procedimiento para establecer, mantener y liberar conexiones físicas entre el dispositivo Terminal (DTE) y el punto de conexión a la red (DCE), o entre dos DTE.

4.2 Capa de Enlace.

De acuerdo a Tanenbaum (2003), es el nivel de datos en donde los bits tienen algún significado en la red. Este nivel puede verse como el departamento de recepción y envío de una compañía de manufactura, el cual debe tomar los paquetes que recibe de la Capa de Red y prepararlos de la forma correcta (tramas) para ser transmitidos por el nivel físico. De igual forma sucede cuando recibe paquetes (bits) del nivel físico y tiene que ponerlos en la forma correcta (tramas) para verificar si la información que está recibiendo no contiene errores, si los paquetes vienen en orden, si no faltan paquetes, etc., para entregarlos a nivel de red sin ningún tipo de error.

Dentro de sus funciones se incluyen la de notificar al emisor (la computadora remota) si algún paquete (trama) se recibe en mal estado (basura); si alguna de las tramas no se recibieron y se requieren que sean enviadas nuevamente (retransmisión), o si una trama esta duplicada, también cuando la trama llegó sin problemas. En resumen, es responsable de la integridad de la recepción y envío de la información, así como de saber dónde comienza la transmisión de la trama y dónde termina, y garantizar que tanto la computadora transmisora como la receptora estén sincronizadas en su reloj y que emplean el mismo sistema de codificación y decodificación.

En esta capa se determina el uso de una disciplina de comunicaciones conocida corno *Hígh Level Data Link Control* (HDLC*).* El HDLC es el protocolo de línea considerado como un estándar universal, que muchos toman como modelo. Los datos en HDLC se organizan en tramas. La trama es un encuadre que incluye bits de redundancia y control para corregir los errores de transmisión; además, regula el flujo de las tramas para sincronizar su transmisión y recepción, también enmascara a las capas superiores de las imperfecciones de los medios de transmisión utilizados.

4.3 Capa de Red.

De acuerdo a Tanenbaum (2003), el nivel de red es el responsable del direccionamiento de mensajes y de la conversión de las direcciones y nombres lógicos a físicos. También determina la ruta del mensaje desde la computadora emisora hasta la computadora receptora, dependiendo de las condiciones de la red.

Dentro de las funciones de encaminamiento de mensajes evalúa la mejor ruta que debe seguir el paquete, dependiendo del tráfico en la red, el nivel de servicios, etc. Los problemas de tráfico que controla tienen que ver con el encaminamiento, intercambio *(switching)* y congestionamiento de paquetes en la red.

Asimismo, maneja pequeños paquetes de datos juntos para la transmisión a través de la red, así como la reestructuración de tramas de datos grandes (números de bits) en paquetes pequeños. En la computadora receptora se reensamblan los paquetes en su estructura de datos original (trama).

A la información proveniente de la capa de transporte se le agregan componentes apropiados para su encaminamiento en la red y para mantener un cierto nivel en el control de errores. La información es presentada según el método de comunicaciones para acceder a la red de área local, la red de área extendida y la conmutación de paquetes.

El diseño de este nivel debe considerar que:

- Los servicios deben ser independientes de la tecnología empleada en la red de datos.
- El nivel de transporte debe ser indiferente al número, tipo y topologías de las redes utilizadas.
- La numeración de la red debe ser uniforme a través de LANs y WANs.

4.4 Capa de Transporte.

De acuerdo a Tanenbaum (2003), El nivel de transporte es llamado ocasionalmente el nivel de *anfitrión a anfitrión* o el nivel de *end to end,* debido a que en él se establecen, mantienen y terminan las conexiones lógicas para la transferencia de información entre usuarios. En particular de la capa 4 hasta la 7 son conocidas como niveles *end to end* y los niveles 1 a 3 son conocidos como niveles de protocolo.

Este nivel puede incluir las especificaciones de los mensajes de difusión, los tipos de datagramas, los servicios de correo electrónico, las prioridades de los mensajes, la recolección de la información y su administración, seguridad, tiempos de respuesta, estrategias de recuperación en casos de falla y segmentación de la información cuando el tamaño es mayor al máximo del paquete según el protocolo.

4.5 Capa de Sesión.

De acuerdo a Tanenbaum (2003), este nivel es el que permite que 2 aplicaciones en diferentes computadoras establezcan, usen y terminen la conexión llamada sesión. El nivel de sesión maneja el diálogo que se requiere en la comunicación de 2 dispositivos. Establece reglas para iniciar y terminar la comunicación entre dispositivos y brinda el servicio de recuperación de errores; es decir, si la comunicación falla y ésta es detectada, el nivel de sesión puede retransmitir la información para completar el proceso en la comunicación. El nivel de sesión es el responsable de iniciar, mantener y terminar cada sesión lógica entre usuarios finales.

Para entender mejor este nivel, se puede pensar en el sistema telefónico. Cuando se levanta el teléfono, espera el tono y marca un número, en ese momento se está creando una conexión física que va desde el nivel uno (físico) como un protocolo de persona a red. Al momento de hablar con la persona en el otro extremo de la línea, se encuentra en una sesión persona a persona. En otras

palabras, la sesión es el diálogo de las dos personas que se transporta por el circuito de la conexión telefónica.

También en este nivel se ejecutan funciones de reconocimiento de nombres para el caso de seguridad relacionado a aplicaciones que requieren comunicarse a través de la red.

4.6 Capa de Presentación.

De acuerdo a Tanenbaum (2003), el nivel de presentación define el formato en que la información será intercambiada entre aplicaciones, así como la sintaxis usada entre las mismas. Se traduce la información recibida en el formato del nivel de aplicación a otro intermedio reconocido. En la computadora receptora, la información es traducida del formato intermedio al usado en el nivel de aplicación de dicha computadora y es, a su vez, responsable de la obtención y liberación de la conexión de sesión cuando existan varias alternativas disponibles.

El nivel de Presentación, maneja servicios como la administración de la seguridad de la red, como la encriptación y desencriptación, también brinda las reglas para la transferencia de información y comprime datos para reducir el número de bits que necesitan ser transmitidos.

4.7 Capa de Aplicación.

Tanenbaum (2003) explica que la capa de aplicación es el nivel más alto del modelo de referencia, el nivel de aplicación es el medio por el cual los procesos de aplicación acceden al entorno. Por ello, este nivel no interactúa con uno más alto.

Además, proporciona los procedimientos precisos que permiten a los usuarios ejecutar los comandos relativos a sus propias aplicaciones. Estos procesos de aplicación son la fuente y el destino de los datos intercambiados. Se distinguen primordialmente 3 tipos de procesos de aplicación:

1. Procesos propios del sistema.
2. Procesos de gestión.
3. Procesos de aplicación del usuario.

5. GNU/Linux.

GNU, que significa *ñu* en inglés, es un acrónimo recursivo de *GNU No es Unix* y en español se pronuncia fonéticamente (Free Software Foundation 2009).

El *Software Libre* es un asunto de libertad, no de precio. Para entender el concepto, debe pensarse en *libre* como en libertad de expresión, no como en cerveza gratis. Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software (Free Software Foundation 2009):

1. La libertad de usar el programa, con cualquier propósito (libertad 0).
2. La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
3. La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
4. La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). De igual forma que la libertad 1 el acceso al código fuente es un requisito previo.

El proyecto GNU se inició en 1984 con el objetivo de crear un sistema operativo completo tipo Unix de software libre: el sistema GNU. El núcleo de GNU no está finalizado, así que se usa GNU con el núcleo Linux. La combinación de GNU y Linux es el sistema operativo GNU/Linux; actualmente se usa en millones de ordenadores (Free Software Foundation 2009).

Desde la introducción del núcleo Linux por Linus Torvald en 1991 y la integración con el proyecto GNU para crear GNU/Linux, fue bien recibido por la industria y academia con varios fines, tales como uso personal y negocios, así como aplicaciones de investigación. En 2002 Reemplazó al Sistema Operativo MAC como el número 2 en S. O. de escritorio y para ese año estaba instalado en el 25% de los servidores corporativos (BusinessWeek 2003.).

Dado que el sistema operativo Ubuntu fue donde se instaló y configuró el cortafuegos transparente a continuación se presenta la historia de esta distribución.

5.1 Historia del Sistema Operativo GNU/Linux: Ubuntu.

Ubuntu (Ubuntu 2009) es una distribución GNU/Linux, la cual ofrece un sistema operativo predominantemente enfocado a computadoras personales, aunque también proporciona soporte para servidores. Esta es una de las más importantes distribuciones de GNU/Linux a nivel mundial, la cual se basa en Debian GNU/Linux y concentra su objetivo en la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares (cada 6 meses: las versiones .04 en abril y las .10 en octubre). El principal patrocinador es una empresa privada llamada Canonical Ltd.

Versiones

Ubuntu 4.10 - Warty Warthog Fue la primera publicación de Ubuntu realizada en octubre de 2004, y recibió ese nombre (Jabalí Verrugoso) porque fue publicado con una gran cantidad de bugs y errores *"warts and all"* (con verrugas y todo). Las principales características son las siguientes (Ubuntu 2009).

- Contiene solo software libre.
- 100% libre de costos o cargos.
- Actualizaciones sin costo de seguridad durante 18 meses.
- Soporte para procesadores x86, am64 y ppc
- Compromiso para crear una nueva versión cada 6 meses.

Ubuntu 5.04 - Hoary Hedgehog Las nuevas características son las siguientes (Wiki.Ubuntu 2008):

- El Live CD fue mejorado y extendido a las versiones de 64bits y Power PC
- Se crea una nueva distribución llamada Kubuntu. Fue construida por la Fundación Ubuntu pero en lugar de usar GNOME como entorno de escritorio, Kubuntu usaba KDE
- Incluye software de primera calidad para productividad como: Evolution 2.2.1.1 y OpenOffice.org 1.1.3
- Mejora el soporte para computadoras portátiles.
- Se tiene un modo de instalación minimalista creado para servidores.

Ubuntu 5.10 - Breezy Badger tercera distribución, orientada al ámbito de la educación y sus principales caracteristicas son las siguientes (Wiki.Ubuntu 2008):

- Nacimiento de Edubuntu,
- Nace la version para servidor
- La distribución Kubuntu reemplazó Kynaptic (gestor de paquetes) por Adept y se convirtió en la primera en usar devtags para una búsqueda más rápida de aplicaciones para Adept.

Ubuntu 6.06 - Dapper Drake Dapper Se convirtió en la primera versión que no cumple los 6 meses de rigor debido a la intención de hacer una versión LTS (Long Time Support) con garantías y las principales características son las siguientes (Wiki.Ubuntu 2008):

- Los usuarios del Live CD podían instalar Ubuntu en sus discos duros, ya no se distribuye el disco de instalación y el Live CD solo el Live CD.

- Disminución en el tiempo de carga del sistema, un apagado gráfico, una nueva herramienta de actualización y una mejor reproducción de vídeo.
- Nace Xubuntu, una nueva hermana de Ubuntu que usaba el escritorio Xfce y estaba dirigida al uso en equipos antiguos.

Ubuntu 6.10 - Edgy Eft Sus principales características son (Wiki.Ubuntu 2008):

- El encendido y apagado de sistema ahora es más rápido.
- Kubuntu añadió un software de gestión de fotos, digiKam.
- Se rediseñó el panel de configuración de sistema; y se mejoraron los botones y el soporte de control de batería de los portátiles.

Ubuntu 7.04 - Feisty Fawn Sus principales características son (Wiki.Ubuntu 2008):

- Soporte para arquitectura SPARC en la versión.
- Incluye un asistente de migración de Windows y la incorporación del controlador ntfs-3g por defecto.
- Soporte para virtualización a nivel de núcleo.

Ubuntu 7.10 - Gutsy Gibbon Sus principales características son (Wiki.Ubuntu 2008):

- Incluye mayor seguridad gracias al AppArmor security framework.
- Mayor velocidad en la búsqueda de archivos.
- Soporte mejorado del sistema de archivos NTFS.
- Interfaz más rápida para el cambio de usuario

Ubuntu 8.04 - Hardy Heron Sus principales características son (Wiki.Ubuntu 2008):

- Fue el segundo lanzamiento LTS.

- Incluye por defecto de nuevas aplicaciones como: Tracker, Brasero, Transmission, Vinagre VNC, PulseAudio.
- Se hizo posible el tener acceso al *Active Directory* usando *Likewise Open*.
- Es la primera versión que incluye el instalador Wubi en el Live CD que permite instalar Ubuntu como un programa de Windows sin necesidad de hacer ninguna partición en el disco duro.

Se aprecia que el desarrollo de Ubuntu es constante, debido a esto Ubuntu se ha convertido en una de las distribuciones más importantes de GNU/Linux, gracias a esto compañías importantes como IBM y HP están ofreciendo soporte y equipos con Ubuntu pre instalado.

5.2 Cortafuegos en GNU/Linux.

De acuerdo a Zwicky (2000), el núcleo de GNU/Linux ofrece toda una serie características inmersas que le permiten funcionar bastante bien como un cortafuegos. La implementación de red incluye la ejecución de código para hacer de filtro IP en una serie de diferentes formas, y proporciona un mecanismo para configurar con bastante precisión qué tipo de reglas que desea poner en su lugar. El cortafuegos de GNU/Linux es lo suficientemente flexible como para que sea muy útil en cualquier configuración o arquitectura de cortafuegos.

Para construir un cortafuegos IP en GNU/Linux, es necesario disponer de un núcleo construido con soporte para cortafuegos IP con el apoyo y la utilidad de configuración adecuada. En todos los núcleos anteriores a la serie 2.2 se usaba la utilidad **ipfwadm.** Los núcleos 2.2.x marcaron el lanzamiento de la tercera generación de cortafuegos IP para GNU/Linux llamado **IP chains**. IP chains utiliza un programa similar a **ipfwadm** llamado **ipchains.** Núcleos de Linux 2.3.15 y posteriores soportan la cuarta generación de cortafuegos IP llamado *netfilter.* (Zwicky, Cooper et al. 2000)

El código de *netfilter* es el resultado de un gran rediseño en el manejo del flujo de paquetes en GNU/Linux, incluye compatibilidad hacia atrás para **ipfwadm** e **ipchains** así como un nuevo comando llamado **iptables** (Zwicky, Cooper et al. 2000). Las siguientes secciones explican brevemente cada una de las versiones de este tipo de cortafuegos.

5.2.1 Netfilter e Iptables.

Según Zwicky (2000), al momento de desarrollar el cortafuegos iptables, este se creó con la idea de que los cortafuegos IP deberían de ser menos complejos, entonces se simplificaron los aspectos de procesamiento de datagramas en el en el código del cortafuegos en el núcleo y fue creando así un marco de referencia de filtrado que es a la vez mucho más limpio y mucho más flexible. Este marco de referencia fue llamado **netfilter**.

Antes de iptables, fue inevitable la complejidad en el diseño del conjunto de reglas del cortafuegos que el administrador debía crear, dado que todas las ampliaciones para el filtrado directo requerían modificaciones al núcleo de Linux, porque todas las políticas de filtrado se aplicaron allí y no había forma de proporcionar una interfaz transparente en el mismo. *Netfilter* aborda tanto la complejidad y la rigidez de las soluciones actuales mediante la aplicación de un marco de referencia genérico en el núcleo que simplifica la forma en que los datagramas son procesados y proporciona una capacidad de extender la política de filtrado sin tener que modificar el núcleo (Zwicky, Cooper et al. 2000).

Las diferencias fundamentales son la supresión de la función de enmascaramiento del código central y un cambio en la ubicación de la entrada y salida de las cadenas. Para acompañar estos cambios, una nueva herramienta de configuración y extensible llamado **iptables** fue creado.

En ipchains, la cadena de entrada se aplica a todos los datagramas recibidos por el anfitrión, con independencia de que estén destinados hacia la computadora local o encaminados a otros receptores. En *netfilter,* la cadena de

entrada se aplica *sólo* a los datagramas destinados a la máquina local, y la cadena adelante sólo se aplica a los datagramas destinados a *otro* anfitrión. Del mismo modo, en ipchains, la cadena de salida se aplica a todos los datagramas dejando a la máquina local, independientemente de si el datagrama se genera en la máquina local o de alguna otra ruta de acogida. En *netfilter,* la cadena de salida se aplica *sólo* a los datagramas generados en este anfitrión y no se aplica a los datagramas están dirigiendo desde otro anfitrión. Este cambio por sí solo ofrece una enorme simplificación de muchas configuraciones de cortafuegos (Zwicky, Cooper et al. 2000).

En la Figura 6 los componentes etiquetados "demasq" y "masq" son componentes separados del núcleo responsable de la entrada y salida de procesamiento de datagramas masqueraded. Estos han sido re-implementados como módulos de *netfilter.* En la aplicación *netfilter* con **iptables,** esta complejidad desaparece completamente (Zwicky, Cooper et al. 2000).

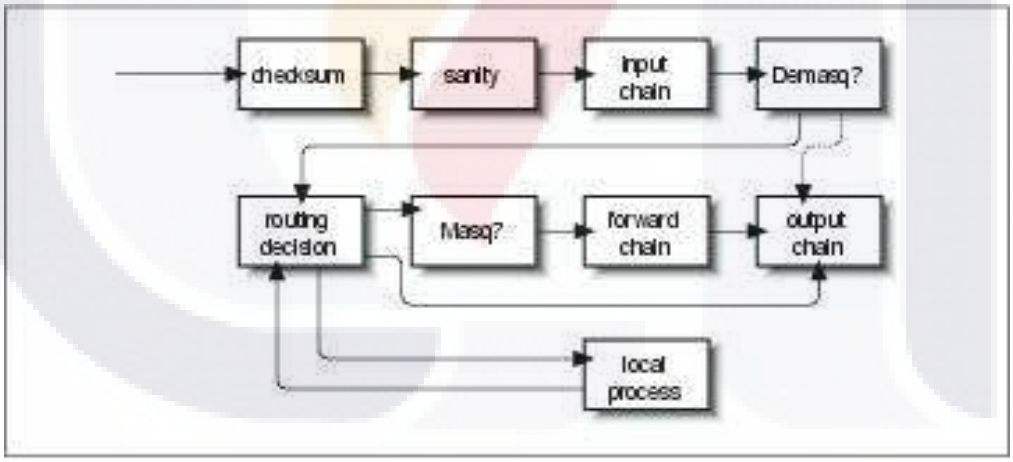


Figura 6: Procesamiento de Datagramas en IPChains *(Adaptado de Zwicky, Cooper et al. 2000)*.

Esta es la forma obvia de diseñar reglas de cortafuegos y ayudó a simplificar el diseño de configuraciones.

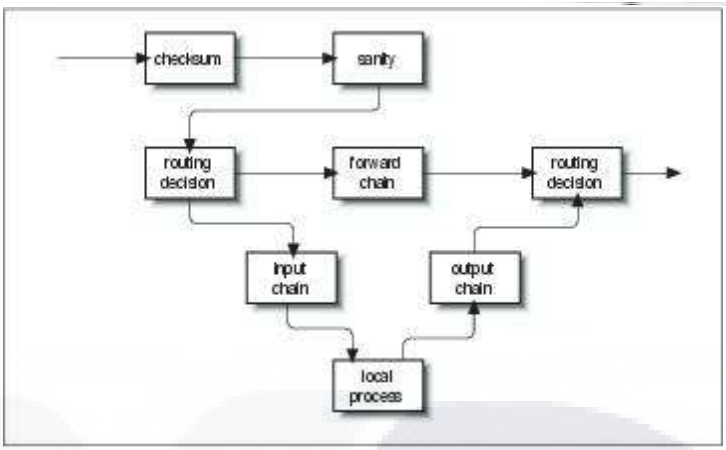


Figura 7**:** Procesamiento de Datagramas en netfilter *(Adaptado de Zwicky, Cooper et al. 2000)*.

5.2.2 L7-filter.

L7-filter es un clasificador para GNU/Linux de Netfilter que identifica paquetes basados en la capa de aplicaciones de datos del modelo OSI. Puede clasificar los paquetes como Kazaa, protocolo de transferencia de hipertexto (HTTP), Jabber, Citrix, Bittorrent, protocolo de transferencia de archivos (FTP), Gnucleus, eDonkey2000, etc., independientemente del puerto.

L7-filter no es un completo paquete de configuración y / o solución de cortafuegos. Es solo un componente identificador de paquetes (con sólo una forma para identificación).Los métodos son: Simple identificación numérica de paquetes, como identificación del número de puerto, el número de IP, bytes transferidos, y así sucesivamente, por ejemplo módulos estándar de iptables.

Existen principalmente dos métodos para la identificación de paquetes en la capa de aplicación, los métodos son los siguientes:

1. Basado en la identificación de expresiones regulares en la capa de aplicación de paquetes, por ejemplo L7-filter.
2. Basado en la identificación en base a la función del paquete en la capa de aplicación, por ejemplo Proyecto para identificar Peer-to-Peer (IPP2P).

6. Cortafuegos GNU/Linux transparente en modo puente.

En general, cualquier computadora con 2 interfaces de red (NIC) puede configurarse como encaminador. Estas dos NICs requieren estar configuradas con diferentes direcciones IP en subredes diferentes tal como se ilustra en la siguiente figura:

Figura 8: Computadora configurada para encaminador IP. (Adaptado de James and Yu 2004).

En esta configuración podemos simplemente activar la opción *IP forwarding* y nuestra computadora se convertirá en un encaminador IP. El procedimiento en GNU/Linux para esta configuración es ejecutar simplemente el siguiente comando en consola (James and Yu 2004):

```
#echo "1" > /proc/sys/net/ipv4/ip_forward
```

Figura 9: Comando para activar un encaminador IP en LINUX (Adaptado de James and Yu 2004).

La pila de protocolos de la configuración en un encaminador IP se observa en la siguiente figura:

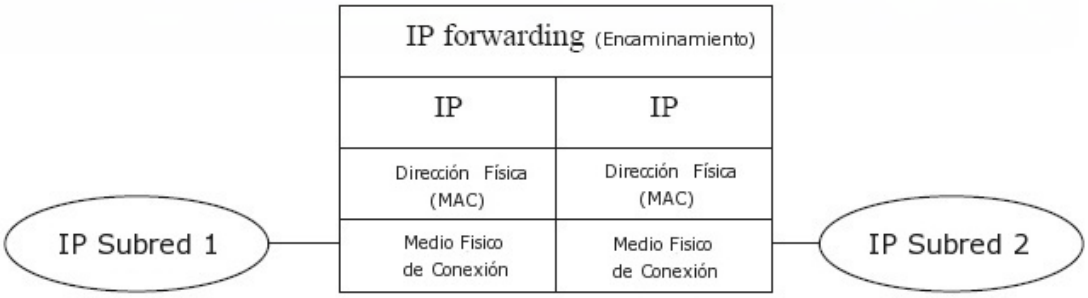


Figura 10: Pila de protocolos para encaminador IP (Adaptado de James and Yu 2004).

Por el contrario, resulta compleja la configuración de un puente debido a que todas las NICs comparten una sola dirección IP. GNU/Linux contiene el software para crear un puente dentro del mismo Núcleo, la configuración de puente puede ser activada con la utilidad **bridge**. En la siguiente configuración (ver Figura 11) se tiene una computadora con GNU/Linux la cual tiene múltiples NICs pero solo una dirección IP (James and Yu 2004):

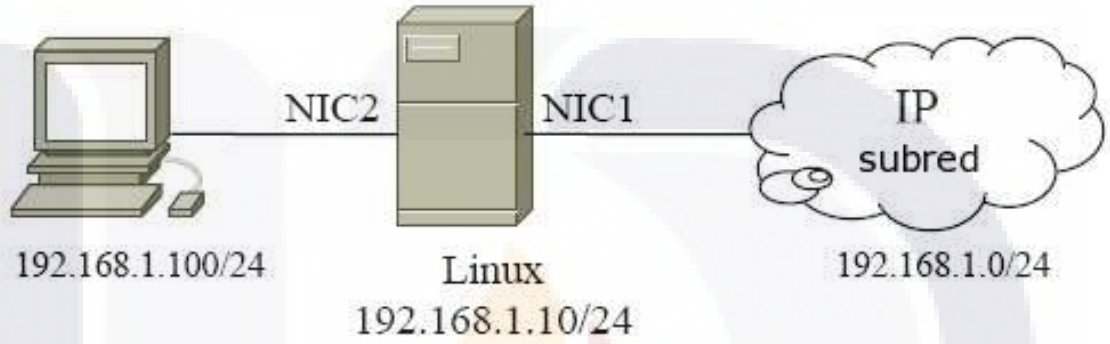


Figura 11: Configuración de GNU/Linux para un puente Ethernet (Adaptado de James and Yu 2004).

Para crear una interface puente Ethernet, es necesario agregar una capa de puente entre la capa de IP y la capa de dirección física (MAC) (James and Yu 2004) tal como se ilustra en la siguiente figura:

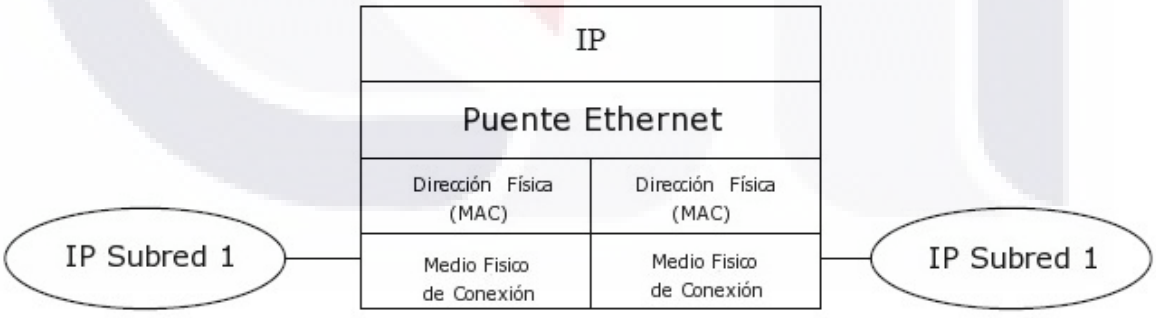


Figura 12: Pila de protocolos para un puente ethernet en GNU/Linux (Adaptado de James and Yu 2004).

Por defecto, una dirección IP esta ligada a una NIC, para crear una interfaz puente es necesario remover esta dirección IP de la NIC y después crear una nueva dirección IP a la interfaz puente.

Un ejemplo del procedimiento de configuración de un puente se muestra en la siguiente figura:

```
# ****************** Creación de la Interfaz puente  con el nombre brl
brctl addbr br1
# ****************** Agregamos las interfaces físicas a la interfaz puente
brctl addif br1 eth0
brctl addif br1 eth1
# ****************** Se Resetea la insterface IP
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
#****************** Activamos el puente
ifconfig br1 up
# ****************** Asignamos una dirección IP al puente
ifconfig br1 192.168.1.10 netmask 255.255.255.0 up
# ****************** Asignamos la Puerta de enlace default
route add default gw 192.168.10.1
```

Figura 13: Script para activar un puente en GNU/LINUX (Adaptado de James and Yu 2004).

Esta configuración de puente soporta el algoritmo Spanning Tree y el Protocolo STP definido en IEEE 802.1D

6.1 Características de un cortafuegos transparente.

Debido a que el dispositivo puente trabaja en la capa de enlace de datos del modelo OSI este no necesita acceder a la tabla de encaminamiento. Un cortafuegos en puente usa la información listada en la sección de Filtrado para decidir si se bloquea o no un paquete. Un cortafuegos en puente puede reconocer datos en las diferentes capas del modelo OSI como lo son la capa de red y la capa de transporte, dado que el cortafuegos en puente sigue siendo un filtro las desventajas del filtrado de paquetes siguen aplicándole (Ingham and Forrest 2002).

Lo que hace diferente a un cortafuegos en puente con respecto al filtrado de paquetes en modo encaminador es que este puede ponerse en cualquier lado dado que este es transparente al nivel de red, puede ser usado para proteger una

sola computadora o un pequeño grupo de computadoras que necesiten estar en la misma subred. Un cortafuegos en puente no necesita tener dirección propia de IP, el puente por si mismo puede ser inmune a cualquier ataque que haga uso de IP. Y no son necesarios cambios en la configuración en los anfitriones protegidos cuando el cortafuegos sea instalado. Los tiempos de instalación pueden ser mínimos llegando incluso a solo 3 segundos en el tiempo de instalación (Limoncelli 1999), por lo tanto los usuarios son mínimamente interrumpidos cuando el puente es instalado.

El primer cortafuegos en puente fue descrito en 1997 y desarrollado para PC corriendo el sistema operativo Disk Operating System (DOS) (Kahn 1997). Una investigación posterior sobre el cortafuegos en puente fue publicado en 1999 (Liu 1999). Keromytis y Wright (Keromytis 2000) debaten en el uso de IPsec en un cortafuegos en puente para establecer de manera segura Redes Virtuales.

7. Pregunta De Investigación.

Dada la problemática anterior, se pude notar la necesidad de crear tecnologías que mejoren sustancialmente la seguridad en las redes. Además, dichas tecnologías deben ser más eficientes, económicas y asequibles a los usuarios. Es por eso que se plante la siguiente pregunta de investigación:

¿Tiene mejor desempeño un desarrollo de cortafuegos de aplicación en modo transparente en GNU/Linux, que soluciones de cortafuegos de aplicación en modo transparente en el Sistema Operativo Windows y en un Sistema Integrado?

8. Hipótesis.

Tomando en cuenta la pregunta de investigación, se plantean un conjunto de hipótesis asociadas a la misma. Estas son las siguientes:

H1: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene un menor consumo de procesador.

H1.a: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene un menor consumo de procesador que una solución de cortafuegos transparente en software con Sistema Operativo Windows.

H1.b: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene un menor consumo de procesador que una solución de cortafuegos transparente en un Sistema Integrado.

H2: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene un menor consumo de memoria RAM.

H2.a: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene un menor consumo de memoria RAM que una solución de cortafuegos transparente en software con Sistema Operativo Windows.

H2.b: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene un menor consumo de memoria RAM que una solución de cortafuegos transparente en un Sistema Integrado.

H3: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene menor jitter en un flujo multimedia.

H3.a: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene menor jitter en un flujo multimedia que una solución de cortafuegos transparente en software con Sistema Operativo Windows.

H3.b: Un desarrollo de cortafuegos de aplicación en modo transparente usando GNU/Linux tiene menor jitter en un flujo multimedia que una solución de cortafuegos transparente en un Sistema Integrado.

III METODOLOGIA.

La presente investigación fue cuantitativa con diseño experimental, la cual tuvo dos objetivos:

1. Desarrollar un cortafuegos de aplicación transparente con GNU/Linux.
2. Comparar el desempeño de cortafuegos creado contra dos soluciones similares, una solución en Windows y una solución en sistema integrado.

Para el primer objetivo se eligió GNU/Linux dado que es de código libre lo cual tiene la ventaja de que se tiene acceso a la documentación de todo su código de igual manera se tiene permiso para poder modificar dicho código.

Para el segundo objetivo se definió el desempeño en base a 3 parámetros los cuales fueron: menor consumo de CPU, menor consumo de memoria RAM y menor jitter en el flujo multimedia. Los dos primeros parámetros están relacionados con la cantidad de tráfico concurrente que podrían llegar a soportar cada cortafuegos y esto a su vez repercutiría en el propio desempeño del cortafuegos. El tercer parámetro es importante medirlo ya que es crítico para tráfico multimedia como por ejemplo de voz sobre IP (VoIP), entre menor sea el jitter mejor funcionamiento tendrá el tráfico multimedia.

Los cortafuegos de aplicación en modo transparente evaluados fueron:

- Websense; Solución de cortafuegos transparente y filtrado de páginas web por contenido en software con Sistema Operativo Windows.
- PacketShaper; Solución de cortafuegos transparente y administración de ancho de banda en Sistema Integrado.
- Netfilter con L7; Desarrollo de cortafuegos de aplicación en modo transparente en Sistema Operativo GNU/Linux

A continuación se muestran las variables de la presente investigación:

- Independientes:

  Tecnología del cortafuegos; Definida por la solución de cortafuegos de aplicación en modo transparente.

  1. En Windows
  2. En GNU/Linux
  3. En Sistema Integrado

- Dependientes: Desempeño definido por:

  1. Porcentaje de uso de CPU
  2. Cantidad de uso de memoria RAM
  3. Jitter en el flujo multimedia

En las siguientes secciones se verá más a detalle el procedimiento que se siguió para la creación del cortafuegos transparente GNU/Linux y las herramientas utilizadas, así como la metodología que se siguió en la estructura de los tres escenarios, la adquisición y procesamientos de los datos y finalmente las pruebas de hipótesis.

1. Modelo de Investigación.

A continuación se presenta el modelo de investigación que esta dado por las variables independientes, dependientes e hipótesis.

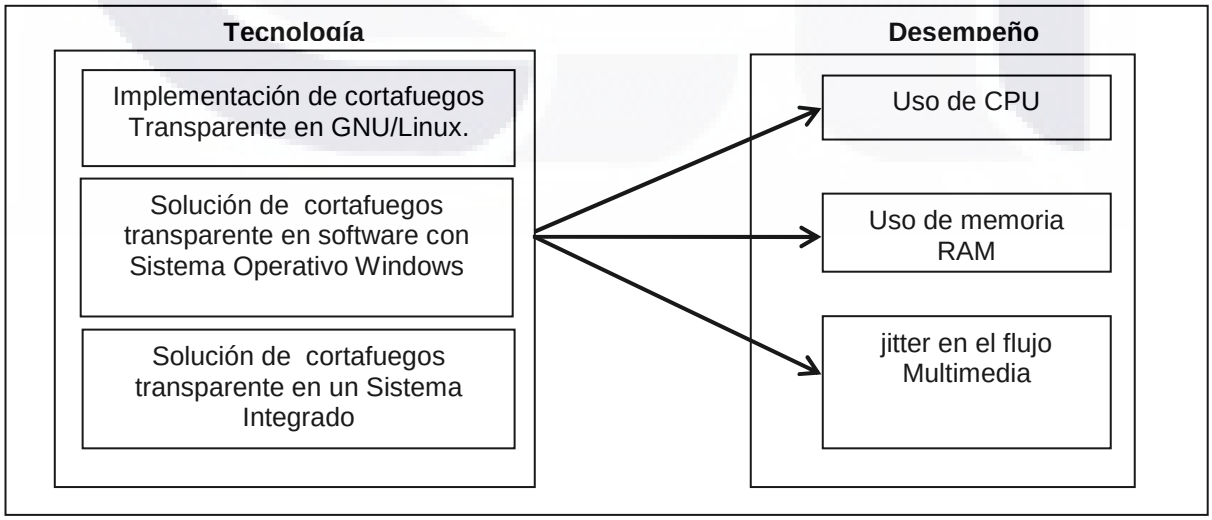


Figura 14: Modelo de Investigación.

Como puede observarse, la variable tecnología fue evaluada en base al uso de CPU, uso de memoria RAM y jitter en el flujo multimedia. Esto nos creó un análisis factorial.

2. Desarrollo del cortafuegos de aplicación en modo transparente.

2.1 Instalación de sistema operativo.

El sistema operativo utilizado fue ***Ubuntu Server 8.04.2 LTS***. La instalación de los paquetes se realizó desde los repositorios oficiales de la distribución Hardy. En la instalación de sistema operativo el único servicio que se le instalo al equipo fue el servicio de Secure Shell para poder conectarse de manera remota y proseguir con la instalación y configuración de las herramientas necesarias para crear el cortafuegos, crear y configurar el puente a si como las herramientas necesarias para leer el uso de CPU y uso de memoria RAM.

2.2 Instalación y configuración del cortafuegos transparente.

Después de instalado el sistema operativo se configuró el cortafuegos para ello se instaló y configuró la herramienta ***l7-filter***, esta herramienta convierte al cortafuegos de GNU/Linux iptables en un cortafuegos de aplicación ya que puede identificar protocolos, una ves identificado el protocolo con l7-filter se utiliza iptables para bloquearlo, en la instalación de l7-filter no fue necesario compilar el núcleo de GNU/Linux ya que l7-filter tiene una versión que se instala y funciona en el espacio de usuario es decir trabaja en la capa superior a el núcleo como un servicio común.

Una vez probado el correcto funcionamiento de l7filter se tenia un cortafuegos de aplicación en GNU/Linux el siguiente paso fue convertir este cortafuegos de aplicación en un cortafuegos de aplicación en modo transparente pare ello se necesitó la herramienta ***brige-utils*** esta herramienta utilizó dos interfaces de red, una vez configurado el puente se dio por terminado el cortafuegos en GNU/Linux.

3. Herramientas de apoyo.

A continuación se describen las herramientas que fueron utilizadas en el laboratorio para leer el uso de CPU, memoria RAM y jitter de paquetes multimedia.

3.1 Herramienta para generar tráfico y medir jitter del flujo multimedia.

Se utilizó la herramienta *Generador de Tráfico de Internet Distribuido* (D-ITG) (Botta, Dainotti et al. 2007). Es una plataforma capaz de producir con exactitud el tráfico usando procesos estocásticos con base a patrones definidos como lo son: el tiempo de salida entre los paquetes (IDT) y el tamaño de paquete (PS), estos procesos se implementan utilizando secuencias variables de números aleatorios. Una gran variedad de distribuciones de probabilidad está disponible: Constante, Uniforme, Exponencial, Pareto, Cauchy, Normal, Poisson y Gamma.

D-ITG soporta la emulación de diversos protocolos como: TCP, UDP, ICMP, DNS, Telnet y VoIP (G.711, G.723, G.729, Detección de actividad de voz, comprimidos RTP). Esto significa que al seleccionar alguno de los protocolos y la distribución el IDT y PS serán automáticamente establecidos.

D-ITG puede realizar la medición del tiempo de la demora del paquete tanto en un solo sentido (OWD), así como la medición del tiempo de la demora del paquete en su viaje de ida y vuelta (RTT), permite evaluar la pérdida de paquetes, el temblor de la voz y el ancho de banda real de transferencia.

Para la generación de cada experimento fue posible fijar una semilla para la generación de las variables implicadas. Esta opción nos dio la posibilidad de repetir varias veces el experimento con exactamente el mismo patrón de tráfico mediante el uso de la misma semilla. Además, ITG-D permite la fijación de TOS (DS) TTL y paquetes de campos. D-ITG actualmente esta disponible para GNU/Linux y Windows.

3.2 Herramientas para medir el uso de CPU y memoria RAM en los cortafuegos transparentes.

Como se comento anteriormente se tendrán tres escenarios en el laboratorio: Windows, GNU/Linux Y Sistema Integrado.

3.2.1 Escenario Windows.

Se Utilizó la herramienta; Instrumental de administración de Windows (WMI, *Windows Management Instrumentation*) es la implementación de Microsoft de Web-Based Enterprise Management (WBEM) una iniciativa que pretende establecer normas estándar para tener acceso y compartir la información de administración a través de la red de computadoras. WMI cumple con WBEM y proporciona compatibilidad integrada para el Modelo de información común (CIM, Common Information Model por sus siglas en Inglés), que describe los objetos existentes en un entorno de administración (Microsoft 2009).

WMI posee varias herramientas administrativas; Propiedades del sistema, Información del sistema y el componente Dependencias de Servicios. (Microsoft 2009). En la presente investigación fue utilizada la herramienta Información del sistema. A continuación se incluye una breve descripción de este componente:

- La herramienta Información del sistema recopila y muestra la información de configuración del sistema. Esto resulta especialmente útil a los técnicos de soporte para solucionar los problemas de los sistemas.

3.2.2 Escenario GNU/Linux.

Para medir el uso de memoria RAM se utilizó la herramienta ***free,*** esta herramienta nos despliega información de la memoria RAM utilizada.

Para medir el uso de CPU se utilizó la herramienta ***mpstat,*** esta herramienta nos despliega información del uso de CPU.

3.2.3 Escenario Sistema Integrado.

Se utilizaron dos herramientas propias del sistema integrado:

sys healt: que nos despliega información de uso del CPU.

sys kmemory pkt buckets*: que* muestra información de la memoria RAM utilizada.

4. Estructura y descripción del laboratorio.

Para entender mejor la interacción que hubo con los equipos se muestra la estructura lógica del laboratorio en la siguiente imagen, en ella se logra apreciar la interacción de los equipos con las diversas herramientas utilizadas:

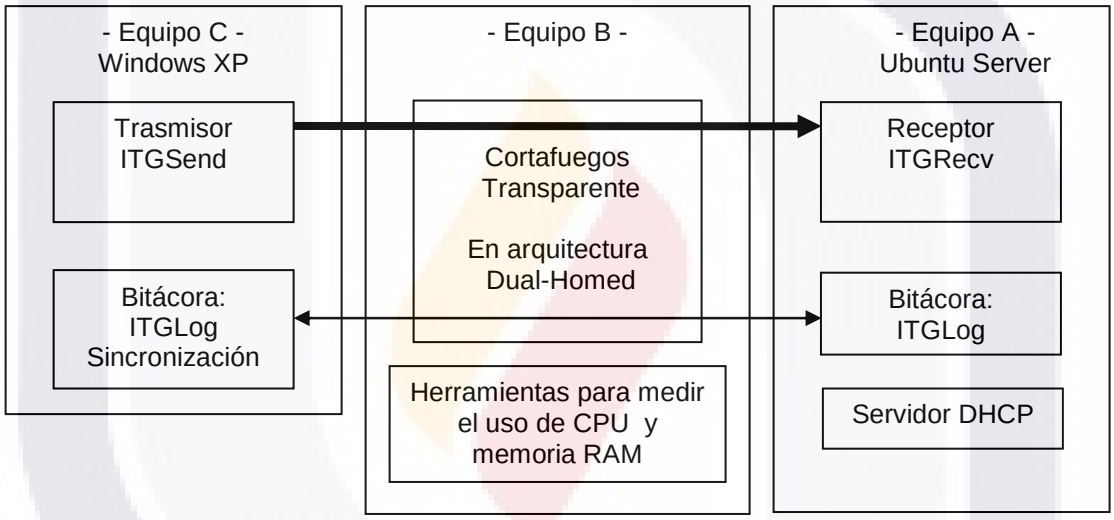


Figura 15: Estructura lógica del laboratorio.

En el ***equipo C*** se utilizó la herramienta ITGSend para generar 3 flujos de datos, los flujos de datos fueron los siguientes:

1. Voz sobre IP (VoIP)
2. aMule
3. ICMP – Ping

Los tres flujos tuvieron una duración de 5 minutos, el primer flujo fue de multimedia, el flujo numero 2 y 3 se utilizaron para incrementar el estrés en los

cortafuegos. Se guardo la configuración de los 3 flujos para poder repetir exactamente la misma cantidad de paquetes y con las mismas características en cada uno de los 3 escenarios como ya se mencionó los escenarios fueron los siguientes:

1. Escenario de Windows.
2. Escenario de GNU/Linux.
3. Escenario de Sistema Integrado.

Cada escenario tuvo una duración de 5 minutos y se tomó una medición cada dos segundos de CPU, memoria RAM y jitter de paquetes multimedia en cada escenario se tomaron 150 muestras de cada uno de los 3 datos mencionados.

El ***equipo B*** fue el cortafuegos, los cortafuegos fueron configurados solo para identificar el tráfico que pasara a través de ellos. No fue necesario aplicar bloqueo ya que no se midió ese parámetro, En los cortafuegos solo se midió el estrés en el CPU y memoria RAM causado por la identificación de paquetes.

El ***equipo A*** fue el que recibió los 3 flujos de datos enviados por el equipo C, la recepción de los datos fue posible gracias a la herramienta ITG-Recv, una vez terminada la transmisión de datos se utilizó la herramienta ITG-Dec para poder obtener el jitter del primer flujo.

4.1 Características en hardware de los equipos utilizados.

Las características de hardware del ***equipo A*** fueron las siguientes:

- CPU : AuthenticAMD CPU 1100 MHz, tamaño de cache: 64 KB
- memoria DDR-1 480 MB
- Tarjetas de Red
    - eth1 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+ (rev 10)
    - eth2 Ethernet controller: Silicon Integrated Systems [SiS] SiS900 PCI Fast Ethernet (rev 90)

Las características de hardware del ***equipo B*** Cortafuegos en Windows y Cortafuegos en GNU/Linux fueron las siguientes:

- CPU : Intel(R) Pentium(R) 4 CPU 1.80GHz, tamaño de cache : 256 KB
- memoria DDR-1 480 MB
- Tarjetas de Red
  - eth2 Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
  - eth3 Intel Corporation 82557/8/9/0/1 Ethernet Pro 100 (rev 02)
  - eth4 VIA Technologies, Inc. VT6102 [Rhine-II] (rev 74)

Las características de hardware del ***equipo B*** Cortafuegos en Sistema Integrado fueron las siguientes:

- Sistema Integrado PacketShaper
  - Version: PacketShaper v8.3.3g1 2008-11-07
  - Product: PacketShaper 7500
  - Part Number: 123-0001-01 REV C
  - Serial Number: 175-10010384
- CPU: Compatible Intel(R) CPU 2.00GHz.
- memoria: 2016 MB
- Tarjetas de Red
  - Inside MAC Address: 00:60:fb:68:26:9e
  - Outside MAC Address: 00:60:fb:68:26:9f
  - Management MAC Address: 00:60:fb:68:26:a0

Las características de hardware del ***equipo C*** fueron las siguientes:

- CPU : Intel(R) Pentium(R) III CPU 1133MHz, tamaño de cache : 512 KB
- memoria DDR-1 384 MB
- Tarjeta de Red
  - Intel Corporation 82801CAM PRO/100 VE (LOM) Ethernet Controller.

4.2 Descripción de los 3 escenarios.

En el primer escenario fue necesario un concentrador para poder conectar el cortafuegos de Windows ya que Websense Express necesita “ver” todo el tráfico que circula por el concentrador, la forma de trabajar de Websense Express es interceptando los paquetes. A continuación se muestra el diagrama de conexión física del escenario Windows.

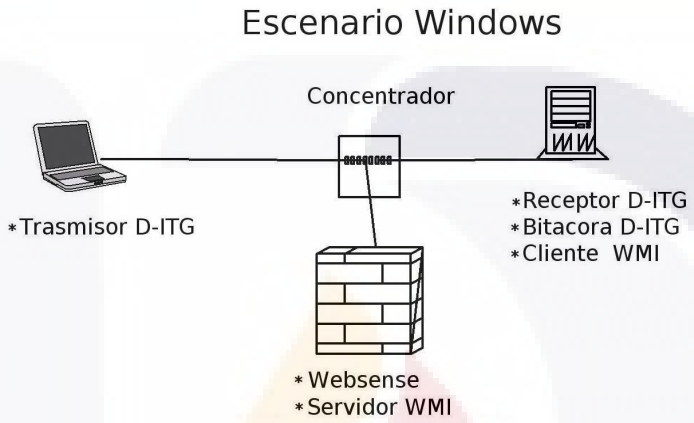


Figura 16: Diagrama físico de conexión del primer escenario.

ITGSend generó 3 flujos que atravesaron el cortafuegos el primero de ellos fue el flujo multimedia, del cual se obtuvo el jitter con la herramienta ITGDec en el Equipo A. A continuación se muestra el diagrama lógico de este escenario:

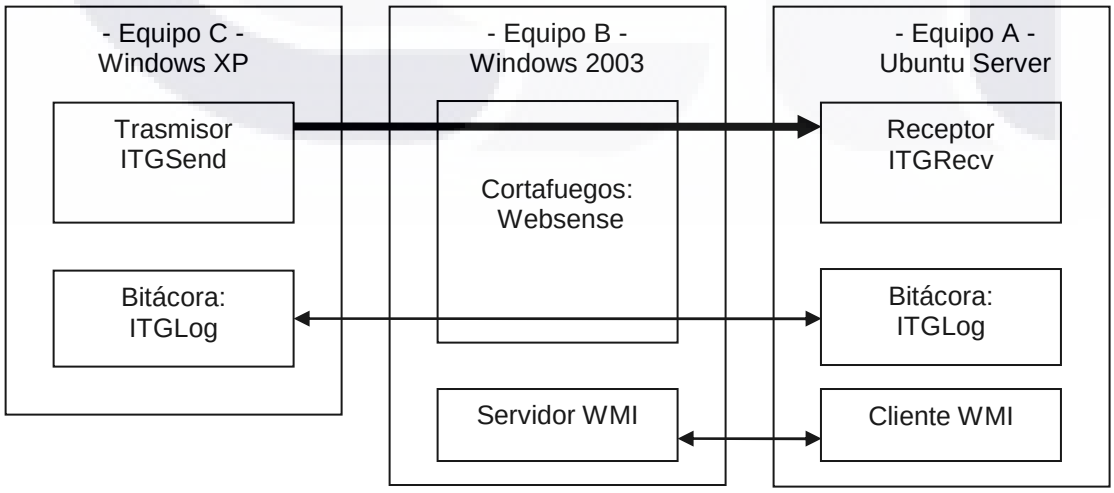


Figura 17: Diagrama lógico de conexión del primer escenario.

En el escenario Windows fue utilizado el servicio de WMI de Windows2003 para leer el consumo de CPU y el consumo de memoria RAM.

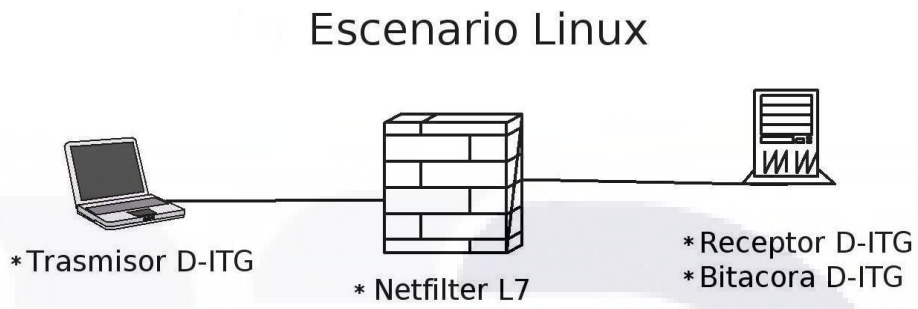


Figura 18: Diagrama físico de conexión del segundo escenario.

En el segundo escenario se observa que el cortafuegos GNU/Linux esta conectado en modo puente.

ITGSend generó 3 flujos que atravesaron el cortafuegos el primero de ellos fue el flujo multimedia, del cual se obtuvo el jitter con la herramienta ITGDec en el Equipo A. A continuación se muestra el diagrama de conexión física del escenario Linux.

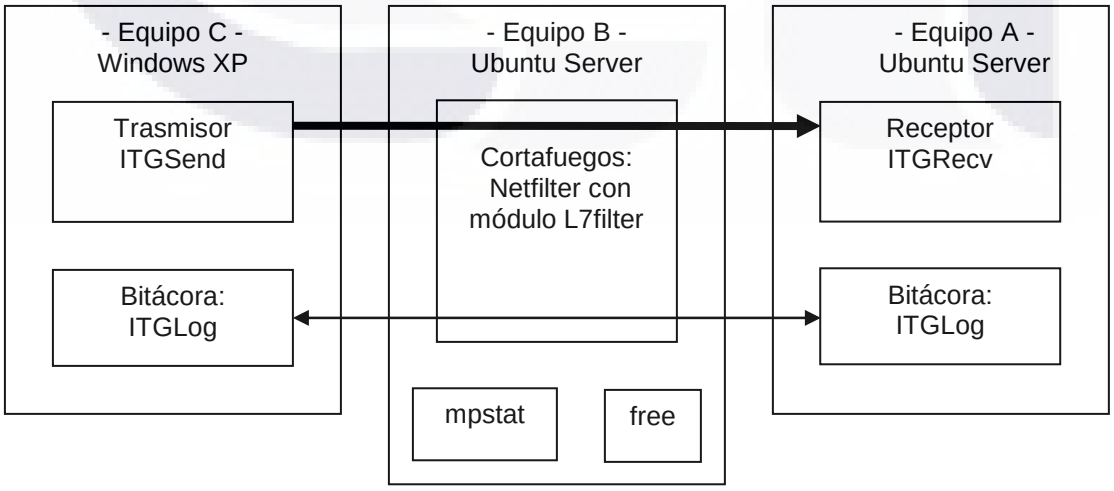


Figura 19: Diagrama lógico de conexión del segundo escenario.

En el escenario Linux fue utilizada la herramienta *mpstat* para obtener el consumo de CPU y se utilizó la herramienta *free* para obtener el consumo de memoria RAM.

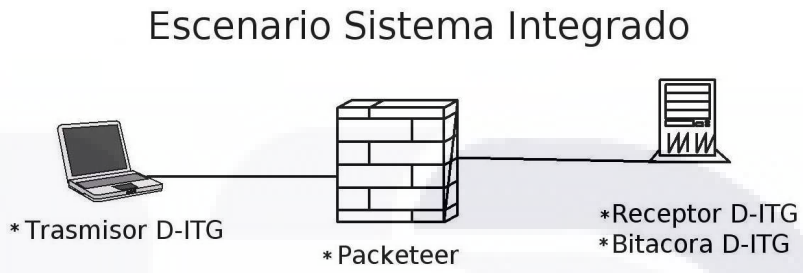


Figura 20: Diagrama físico de conexión del tercer escenario.

El escenario fue conectado de manera similar al segundo escenario ya que PacketShaper también fue conectado en modo puente.

ITGSend generó 3 flujos que atravesaron el cortafuegos el primero de ellos fue el flujo multimedia, del cual se obtuvo el jitter con la herramienta ITGDec en el Equipo A.

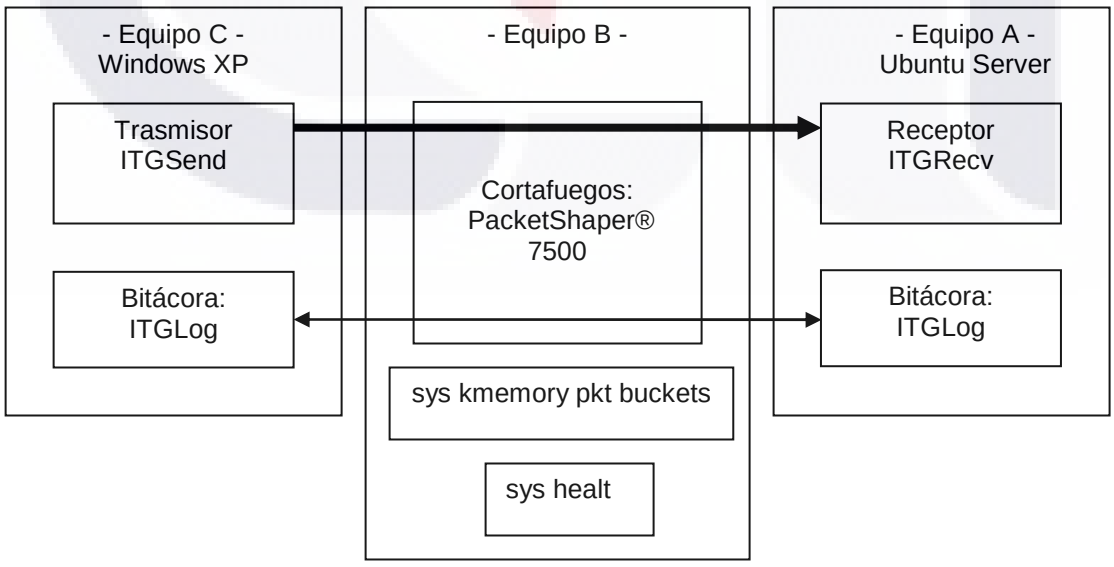


Figura 21: Diagrama lógico de conexión del tercer escenario.

En el escenario sistema entregado fue utilizada la herramienta *sys healt* para obtener el consumo de CPU y se utilizó la herramienta *sys kmemory pkt buckets* para obtener el consumo de memoria RAM.

IV RESULTADOS

Como ya se mencionó anteriormente, se obtuvieron 150 muestras en cada uno de los escenarios del experimento. Estos datos fueron analizados para obtener estadísticas descriptivas y pruebas de hipótesis con dichos datos. A continuación se presentan los resultados obtenidos:

1. Estadística descriptiva.

La Tabla 1 describe el comportamiento estadístico de la muestra de datos. Los datos de CPU están medidos en porcentaje de utilización, los datos de memoria están dados en bits utilizados, y los datos de jitter están en unidades de tiempo.

- Se puede observar que en el caso de uso de CPU, la implementación más eficiente en promedio es del sistema integrado ya que utiliza solamente el 1.34% comparado con el 5.7% y 7.67% de Linux y Windows respectivamente. Por lo tanto, la mejor solución es el sistema integrado. Es decir, la implementación bajo Linux requiere aproximadamente 4.5 veces más tiempo y bajo Windows aproximadamente 5.5 veces más tiempo. En consecuencia, el procesamiento de las peticiones es más lento en estos últimos.

- Por otro lado, al comparar los promedios de la cantidad de memoria principal utilizada tenemos que la mejor solución es la implementada bajo el sistema operativo Linux ya que solo utiliza 73 megabits de memoria comparado con 271 y 442 megabits en el sistema integrado y Windows respectivamente. Podemos observar que la implementación bajo Linux representa aproximadamente el 27% de memoria comparado con el sistema integrado y el 16.5% comparado con la implementación bajo Windows. En consecuencia, se puede decir que la solución propuesta bajo Linux es la mejor.

- Por último, al comparar los promedios del tiempo utilizado para el procesamiento tenemos que la mejor solución es la implementada bajo el sistema operativo Linux ya que solo utiliza 0.278 milisegundo comparado con 0.282 y 1.01 milisegundos en el sistema integrado y Windows respectivamente. Podemos observar que la implementación bajo Linux es prácticamente equivalente con la solución de sistema integrado puesto que la diferencia entre ambos es de 0.01 milésimas, pero comparándolos con la implementación bajo Windows, esta última es aproximadamente 4 veces más grande que las otras dos. Por lo tanto, se puede decir que la solución propuesta bajo Linux y bajo el sistema integrado son las mejores.

2. Frecuencias

Estadísticas

	CPU GNU/ Linux	memoria Utilizada GNU/Linux	Multimedia jitter GNU/Linux	CPU Windows	memoria Utilizada Windows	Multimedia jitter Windows	CPU S. I.	memoria Utilizada Sistema Integrado	Multimedia jitter Sistema Integrado
Validos	150	150	150	150	150	150	150	150	150
Perdidos	0	0	0	0	0	0	0	0	0
Media	5.7467	0.73E8	.000278	7.67	4.42E8	.001017	1.34	2.71E8	.000282
Error estándar de la Media	.04652	23959.377	.000020	.938	89860.856	.000012	.001	5383.111	.000023
Mediana	6.0000	0.73E8	.000157	5.00	4.42E8	.001003	1.34	2.71E8	.000131
Modo	6.00	72749056[a]	.000142	7	440913920[a]	.000998[a]	1	270976000	.000115
Desviación Estándar	.56978	293441.2	.000250	11.490	1100566.2	.000149	.008	65929.3	.000293
Variante	.325	8.611E10	.000	132.020	1.211E12	.000	.000	4.347E9	.000
Skewness	.044	.299	2.428	3.602	.481	1.784	.162	.018	2.028
Error estándar del Skewness	.198	.198	.198	.198	.198	.198	.198	.198	.198
Kurtosis	-.411	-.956	6.020	12.831	-.525	11.689	.650	-1.176	3.380
Error estándar de Kurtosis	.394	.394	.394	.394	.394	.394	.394	.394	.394
Rango	2.00	1081344	.001323	58	4391336	.001193	0	227328	.001295
Mínimo	5.00	72495104	.000131	0	440057856	.000666	1	27089612	.000110
Máximo	7.00	73576448	.001454	58	444449192	.001859	1	27112345	.001405

a. Existen Modos múltiples, se muestra el valor más pequeño.

Tabla 1: Distribuciones de frecuencias de los resultados.

La Figura 22 muestra los histogramas de frecuencia obtenidos al medir la variable uso de CPU, en los cuales se puede observar gráficamente el comportamiento y los resultados descritos anteriormente.

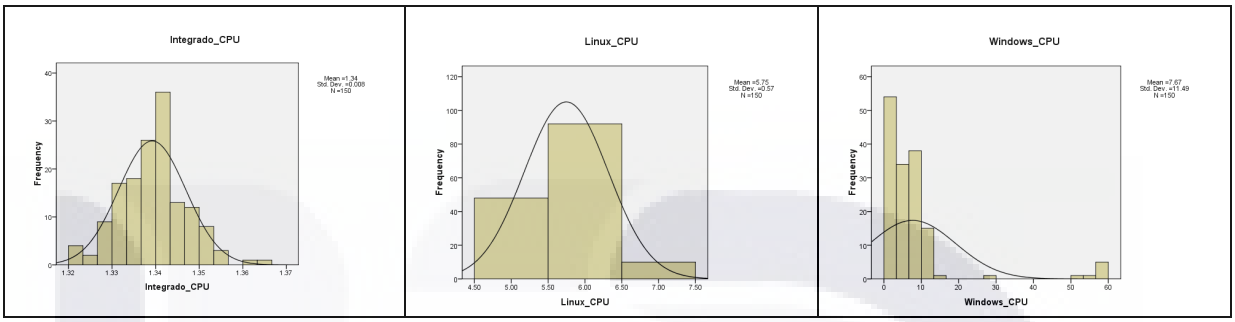


Figura 22: Histogramas de frecuencias con respecto a la utilización de CPU.

Adicionalmente, la Figura 23 muestra los histogramas de frecuencia obtenidos al medir la variable uso de memoria RAM, en los cuales se puede observar gráficamente el comportamiento y los resultados descritos anteriormente.

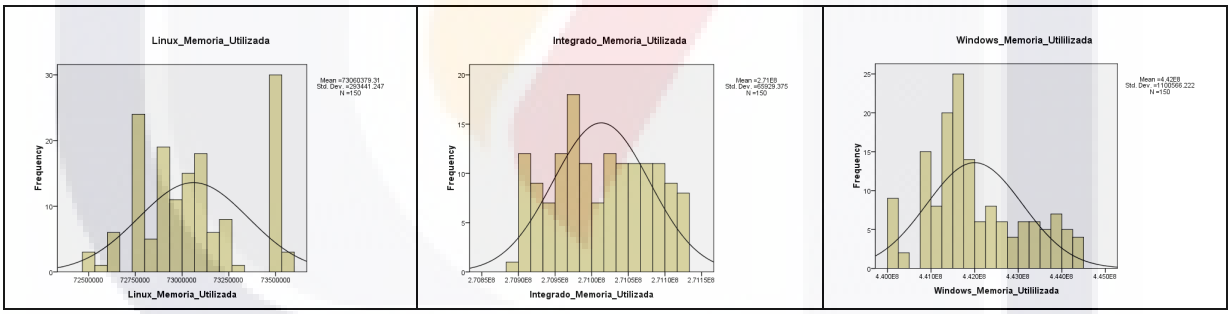


Figura 23: Histogramas de frecuencias con respecto a la memoria RAM utilizada.

Por último, la Figura 24 muestra los histogramas de frecuencia obtenidos al medir la variable jitter del flujo multimedia, en los cuales se puede observar gráficamente el comportamiento y los resultados descritos previamente.

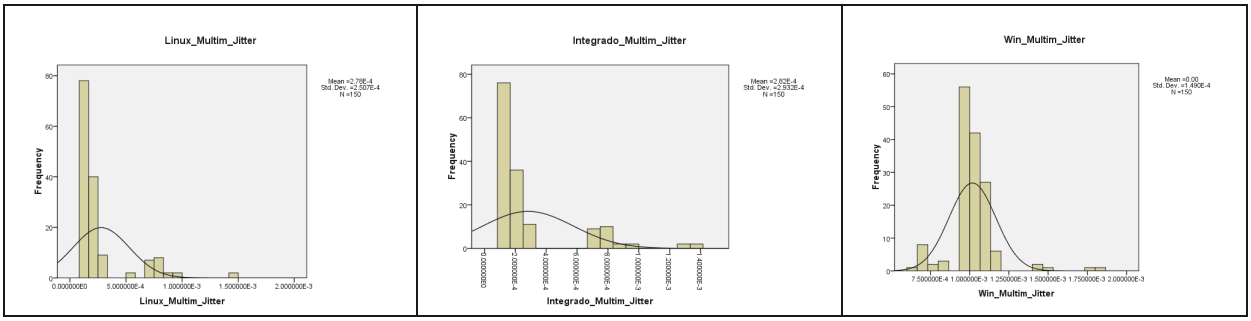


Figura 24: Histogramas de frecuencias con respecto al jitter del flujo multimedia.

3. Pruebas de hipótesis.

Con el fin de verificar los resultados obtenidos en la sección anterior se realizó una prueba de ANOVA. Los resultados de esta se muestran en la Tabla 2. Se puede observar que con respecto a las tres variables existen diferencias significativas. Es decir, existe un comportamiento diferente en los cortafuegos implementados bajo los tres esquemas propuestos (p<=.001). Por lo tanto, uno de dichas implementaciones es mejor comparada con las otras dos para cada variable de estudio utilizada.

ANOVA

		Suma de cuadrados	df	Media Cuadrada	F	Significancia
CPU Utilizado	Entre Grupos	3162.938	2	1581.469	35.849	.000
	Dentro Grupos	19719.376	447	44.115		
	Total	22882.314	449			
Bits memoria Utilizada	Entre Grupos	1.023E19	2	5.113E18	1.178E7	.000
	Dentro Grupos	1.940E14	447	4.339E11		
	Total	1.023E19	449			
Multimedia jitter	Entre Grupos	.000	2	.000	477.615	.000
	Dentro Grupos	.000	447	.000		
	Total	.000	449			

Tabla 2: Análisis de varianza del modelo.

Una vez comprobado que existieron diferencias entre las muestras obtenidas, es necesario identificar cual de las implementaciones es mejor bajo cada variable de medición propuesta. Para dicho efecto, se realizó análisis post

hoc a través de LSD. La Tabla 3 muestra los resultados de dicho análisis. Los resultados son como sigue:

- Se puede observar que en el caso de uso de CPU, la implementación más eficiente en promedio es del sistema integrado comparado con Linux y Windows respectivamente. Por lo tanto, la mejor solución es el sistema integrado, lo cual corrobora lo anteriormente descrito. Además, esto nos dice que se acepta la hipótesis H1a ya que la implementación GNU/Linux es la mejor comparada con la implementación bajo Windows ($p<=.012$). Adicionalmente, la implementación bajo sistema integrado tiene un desempeño significativo mejor comparada con la implementación en GNU/Linux, por lo que la hipótesis H1b se rechaza ($p<=.001$).

- Por otro lado, al comparar los resultados de la cantidad de RAM utilizada tenemos que la mejor solución es la implementada bajo el sistema operativo Linux comparado con en el sistema integrado y Windows, lo cual corrobora lo anteriormente descrito. Además, esto nos dice que se *comprueba la hipótesis (H2)* ya que la implementación GNU/Linux fue significativamente mejor comparada con la solución integrada y con la implementación bajo Windows. Esto último, nos siguiere aceptar tanto la hipótesis 2a ($p<=.001$) como la 2b ($p<=.001$).

- Por último, al comparar los promedios del tiempo utilizado para el procesamiento tenemos que la mejor solución es la implementada bajo el sistema operativo Linux comparado con la solución integrada pero sin una diferencia significativa y que ambas son mejores que la solución bajo Windows, lo cual corrobora lo anteriormente descrito. Además, esto nos dice que se acepta la hipótesis H3a ya que la implementación GNU/Linux es mejor comparada con la implementación bajo Windows ($p<=.001$). Adicionalmente, la implementación bajo Linux tiene un desempeño

significativo similar comparada con la implementación en el sistema integrado, por lo que la hipótesis H3b se rechaza (p<=.89).

Post Hoc

Comparaciones Múltiples								
Variable Dependiente		(I) Tipo de S.O.	(J) Tipo de S.O.	Diferencia de medias (I-J)	Error estándar	Sig.	95% Intervalo de confianza	
							Límite inferior	Límite superior
CPU Utilizado	LSD	Windows	GNU/Linux	1.92667*	.76694	.012	.4194	3.4339
			Integrado	6.33412*	.76694	.000	4.8269	7.8414
		GNU/Linux	Windows	-1.92667*	.76694	.012	-3.4339	-.4194
			Integrado	4.40745*	.76694	.000	2.9002	5.9147
		Integrado	Windows	-6.33412*	.76694	.000	-7.8414	-4.8269
			GNU/Linux	-4.40745*	.76694	.000	-5.9147	-2.9002
Bits de Memoria Utilizada	LSD	Windows	GNU/Linux	3.689E8	76061.392	.000	3.69E8	3.69E8
			Integrado	1.710E8	76061.392	.000	1.71E8	1.71E8
		GNU/Linux	Windows	-3.689E8	76061.392	.000	-3.69E8	-3.69E8
			Integrado	-1.980E8	76061.392	.000	-1.98E8	-1.98E8
		Integrado	Windows	-1.710E8	76061.392	.000	-1.71E8	-1.71E8
			GNU/Linux	1.980E8	76061.392	.000	1.98E8	1.98E8
Multimedia jitter en segundos	LSD	Windows	GNU/Linux	.000739767*	.000027568	.000	.00068559	.00079394
			Integrado	.000735967*	.000027568	.000	.00068179	.00079014
		GNU/Linux	Windows	-.00073976*	.000027568	.000	-.0007939	-.00068559
			Integrado	-.000003800	.000027568	.890	-.0000579	.00005038
		Integrado	Windows	-.00073596*	.000027568	.000	-.0007901	-.00068179
			GNU/Linux	.000003800	.000027568	.890	-.0000503	.00005798
*. La diferencia entre medias es significante al nivel de 0.5								

Tabla 3: Resultados del análisis Post Hoc.

V CONCLUSIONES.

La intención de la presente investigación fue, primeramente, ver la factibilidad de crear un cortafuegos de aplicación en modo transparente con GNU/Linux dado la gran importancia que adquirieron este tipo de dispositivos para cualquier red de computadoras que esté conectada hacia internet. Este tipo de cortafuegos generalmente, son colocados en el punto de conexión hacia internet. Se descubrió que con las herramientas utilizadas: Netfilter y L7-filter se pudo crear de manera relativamente sencilla el cortafuegos de aplicación: Después, gracias al uso de la herramienta brigde-utils fue posible configurar el cortafuegos transparente en modo puente.

Una vez creado el cortafuegos, se continuó con el segundo objetivo de la investigación que fue el medir el desempeño del cortafuegos creado bajo GNU/Linux contra una solución similar en Windows y una solución similar en sistema integrado. Los parámetros para medir el desempeño fueron el menor porcentaje de uso de CPU, la menor cantidad de memoria RAM utilizada y un menor jitter en un flujo multimedia. Es importante medir el consumo de CPU ya que este impacta en la velocidad a la que son procesados lo paquetes que atraviesan el cortafuegos. La memoria RAM en este tipo de equipos es importante porque influye en el número de conexiones simultaneas que puede manejar el cortafuegos, finalmente el jitter es también importante ya que las aplicaciones multimedia actuales requieren un jitter pequeño para poder funcionar adecuadamente.

Al compararse las soluciones se obtuvieron tres hipótesis donde se propuso como mejor al desarrollo en GNU/Linux. De los resultados obtenidos se puede concluir que la solución propuesta podría trabajar en ambientes reales, en redes donde se quiera bloquear aplicaciones como programas de mensajería instantánea, programas Peer to Peer como Ares, Edonkey etc. Las aplicaciones mencionas utilizan el puerto 80 utilizado por el protocolo http por lo que no es posible cerrar dicho puerto.

La primera hipótesis parte del argumento que la solución propuesta bajo GNU Linux tiene un mejor desempeño con respecto al porcentaje de uso de CPU comparado con una solución bajo Windows y otra en sistema integrado. Para este caso, no se encontró evidencia que soporte la hipótesis planteada ya que, efectivamente, la solución GNU Linux si fue mejor que la solución de Windows (H1a) pero no es mejor comparada con la solución integrada (H1b). En consecuencia, debido a la evidencia encontrada se recomienda utilizar una solución integrada cuando el sistema a implementar demande como principal variable el porcentaje usado de CPU. Por ejemplo, cuando se tengan sistemas SCADA y/o enormes bases de datos centralizadas utilizadas en aplicaciones críticas, ya que este tipo de aplicaciones generan una gran cantidad de tráfico y necesitan trabajar en tiempo real.

La segunda hipótesis también parte del argumento que la solución propuesta bajo GNU/Linux tiene un mejor desempeño con respecto al uso de memoria RAM comparado con una solución bajo Windows y otra en sistema integrado. Para este caso, se encontró evidencia que soporta la hipótesis planteada ya que, efectivamente, la solución GNU/Linux fue mejor que la solución de Windows (H2a) así como también comparada con la solución integrada (H2b). En consecuencia, debido a la evidencia encontrada se recomienda utilizar una solución GNU/Linux cuando el sistema a implementar demande como principal variable el uso de memoria RAM. Por ejemplo, cuando se tengan redes WAN y MAN con miles de usuarios, generalmente encontradas en corporativos y universidades.

Finalmente, la tercera hipótesis también parte del argumento que la solución propuesta bajo GNU/Linux tiene un mejor desempeño con respecto al jitter en un flujo multimedia comparado con una solución bajo Windows y otra en sistema integrado. Para este caso, se encontró evidencia que soporta la hipótesis planteada ya que, efectivamente, la solución GNU/Linux fue mejor que la solución de Windows (H3a) así como también comparada con la solución integrada pero

sin una diferencia significativa (H3b). En consecuencia, debido a la evidencia encontrada se recomienda utilizar una solución GNU/Linux o sistema integrado cuando el sistema a implementar demande como principal variable un jitter mínimo. Por ejemplo, cuando se tengan aplicaciones multimedia como video conferencias o la telefonía IP ya que dichas aplicaciones necesitan un jitter pequeño para poder funcionar adecuadamente.

Por lo anterior, se concluye que la mejor opción es GNU/Linux ya que aun cuando PacketShaper tuvo un menor uso de CPU esta solución esta limitada a su único CPU, por el contrario GNU/Linux ofrece compatibilidad para instalarse en hardware para servidores de alto desempeño disponibles en el mercado a un costo inferior al de la solución PacketShaper, actualmente existe hardware para servidor que poseen hasta cuatro CPU Quad Core con lo que nos da un total de 16 CPUs y con hasta 64 GB de memoria RAM. Con lo cual estaría superando a la solución integrada PacketShaper.

Quedo demostrado que GNU/Linux ofrece todas las herramientas y documentación para desarrollar soluciones de alta calidad que podrían ser utilizadas en lugar de soluciones comerciales. Cabe aclarar que los resultados solo son ciertos para la configuración utilizada; por lo que deben tomarse con cautela.

1. Ventajas de la solución propuesta.

Una gran ventaja de la solución propuesta es que GNU/Linux al ser software libre se tiene acceso al código fuente y adicionalmente se permite ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software para ajustarlo a las necesidades especificas de cada empresa o institución, esto a sido una de las razones fundamentales por las cuales GNU/Linux ha ido creciendo y mejorando a pasos agigantados.

2. Desventajas de la solución propuesta.

El desarrollo propuesto requiere que el administrador del cortafuegos posea conocimientos técnicos en la administración por línea de comandos de un servidor GNU/Linux y también es necesario conocimientos en redes de computadoras, se cree que esta desventaja se puede eliminar programando una interfaz gráfica de administración para este propósito, utilizando algún lenguaje también GNU como PHP, Phyton o Perl los cuales tienen una gran documentación y cada vez son mas utilizados .

3. Limitantes.

Dado el tipo de estudio que se llevó a cabo, existen una serie de aspectos relevantes que pudieran haber tenido efecto en los resultados encontrados. Es por esta razón se sugiere tomar con cautela los hallazgos del presente estudio. Los experimentos realizados fueron en ambientes asilados y controlados, por lo que si se realizan en ambientes con tráfico real los resultados podrían ser diferentes.

Adicionalmente, si se cambiara la versión de sistema operativo y/o hardware utilizado los resultados podrían ser diferentes. Se puede mencionar que existe una diferencia de 200Mhz en el CPU del sistema integrado ya que este posee un procesador a 2GHz y el procesador del equipo donde fue instalado el cortafuegos Websense y el cortafuegos GNU/Linux es de 1.8Ghz. Esta diferencia se cree que no fue la causa de haya tenido mejor desempeño el sistema integrado, ya que la diferencia en desempeño fue aproximadamente de 5.5 y 4.5 mejor en el sistema integrado, pero no es posible asegurarlo. Las soluciones utilizadas Websense y PacketShaper tienen capacidades que no fueron analizadas en la presente investigación estas capacidades son el modelado de tráfico y la contabilización de tráfico. Una funcionalidad importante y que no fue contemplada en la presente investigación es la de ver la cantidad de falsos positivos y falsos negativos en la identificación de protocolos que tienen las soluciones analizadas.

Además, solo se utilizó un flujo de VoIP en un solo sentido y las conversaciones reales utilizan dos flujos uno en cada sentido. En consecuencia, en ambientes reales de VoIP los resultados podrían ser diferentes.

Pueden existir algunas otras variables no identificadas para el presente estudio que tengan efecto en el mismo, y en consecuencia, afectar las conclusiones generadas.

4. Trabajos futuros.

Con el fin de poder obtener resultados generalizables y reales su sugiere medir el desempeño en un ambiente real, o bajo otras condiciones para ver el comportamiento de las variables involucradas en el estudio.

Una segunda investigación es el desarrollar el cortafuegos transparente en modo puente en al menos dos distribuciones diferentes de GNU/Linux y medir el desempeño para saber si la elección de Ubuntu Server 8.04.2 fue la mejor. Así como también elegir alguna otra solución de firewall de aplicación en modo transparente sistema integrado por ejemplo Fortinet, e incluso medir el desempeño de Websense en Windows contra Websense en Linux.

Una tercera propuesta consiste en utilizar un hardware para el desarrollo en GNU/Linux con un procesador a 2 GHz para identificar en ese escenario la solución con el mejor desempeño. Asimismo, se sugiere reproducir el presente estudio utilizando un flujo bidireccional de VoIP, tal como lo hacen las conversaciones reales.

El agregar las funcionalidades de modelado de tráfico y contabilidad de paquetes al desarrollo en GNU/Linux para con ello compararlas contra las soluciones Websense y PacketShaper podría resultar en un estudio que genere

mayor información a la problemática aquí estudiada. Razón por la cual, se sugiere esta línea de investigación

Por último, se sugiere elaborar una interfaz grafica de administración para el firewall para eliminar el alto conocimiento técnico en GNU/Linux que se requiere por parte del administrador del cortafuegos propuesto. De esta forma, se pueden generalizar aún más los resultados que se encontraran en esa investigación adicional.

ANEXO

En esta sección se muestra como fueron instalados y configurados los sistemas operativos, para cada uno de los tres equipos, también se detalla la instalación y configuración de las herramientas utilizadas en el experimento, finalmente se detalla el procedimiento que se siguió para la obtención de los datos de CPU, memoria RAM y jitter del flujo Multimedia.

Instalación y Configuración del equipo C.

- Receptor D-ITG -: Bitácora D-ITG, Cliente WMI y Servidor DHCP.

En esta sección se muestra como fue instalado y configurado el equipo C.
El Sistema Operativo instalado fue Ubuntu Hardy Heron (8.04 LTS).
Además se utilizó el siguiente software:

- D-ITG (Distributed Internet Traffic Generator)
- wmi-client
- dhcp3-server

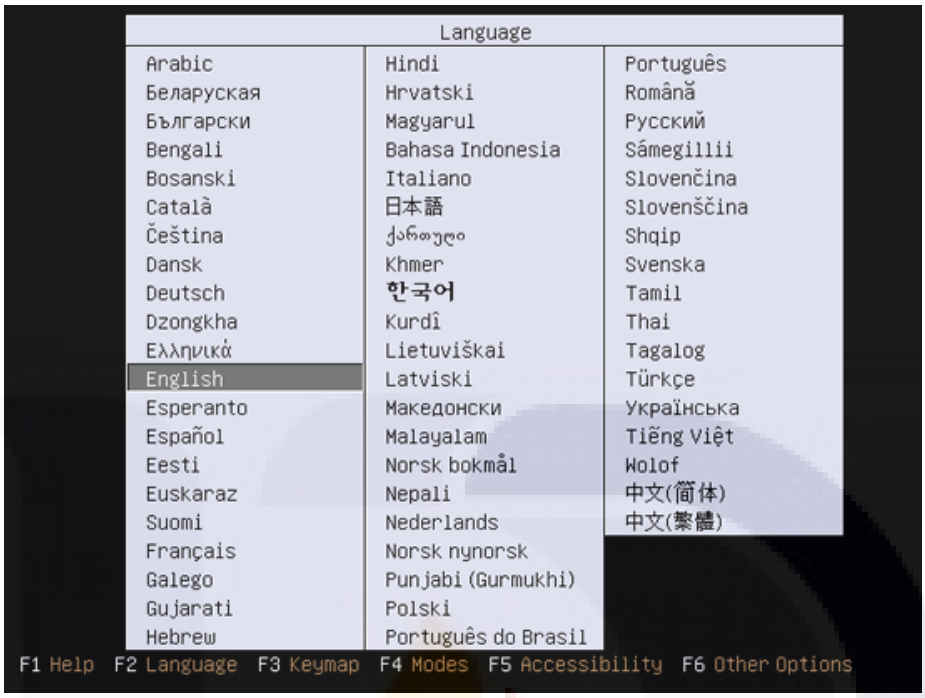
Requerimientos

Para la instalación fue necesario lo siguiente:

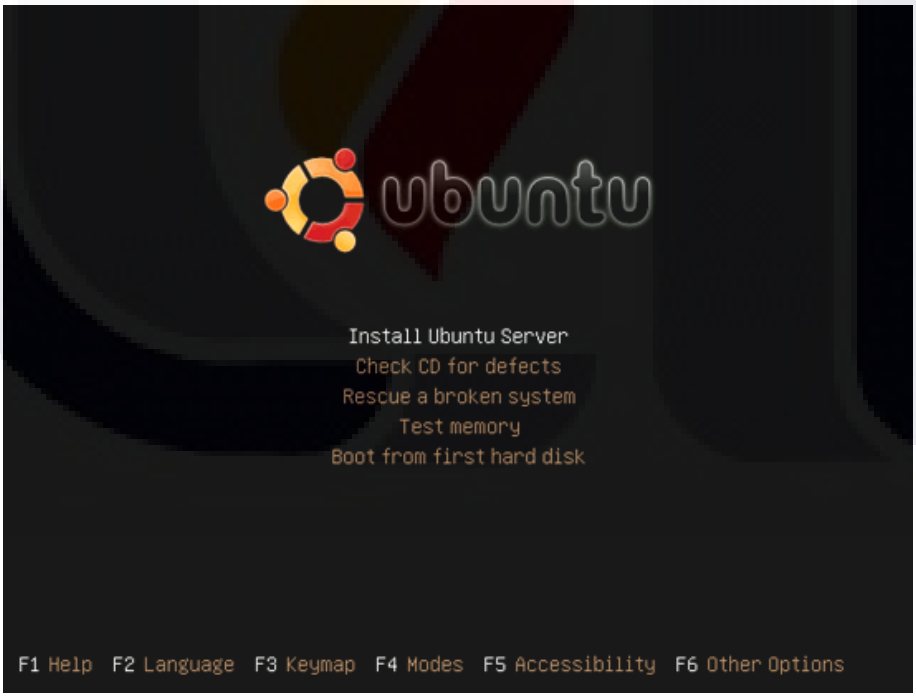
1. El CD de instalación de Ubuntu Server 8.04.2 LTS, disponible aquí: ftp://releases.ubuntu.com/releases/hardy/ubuntu-8.04.2-server-i386.iso
2. Versión estable de D-ITG, disponible aquí: http://www.grid.unina.it/software/ITG/codice/D-ITG-2.6.1d.zip
3. Una conexión de banda ancha hacia Internet.

1. Instalación de Ubuntu Server 8.04.2.

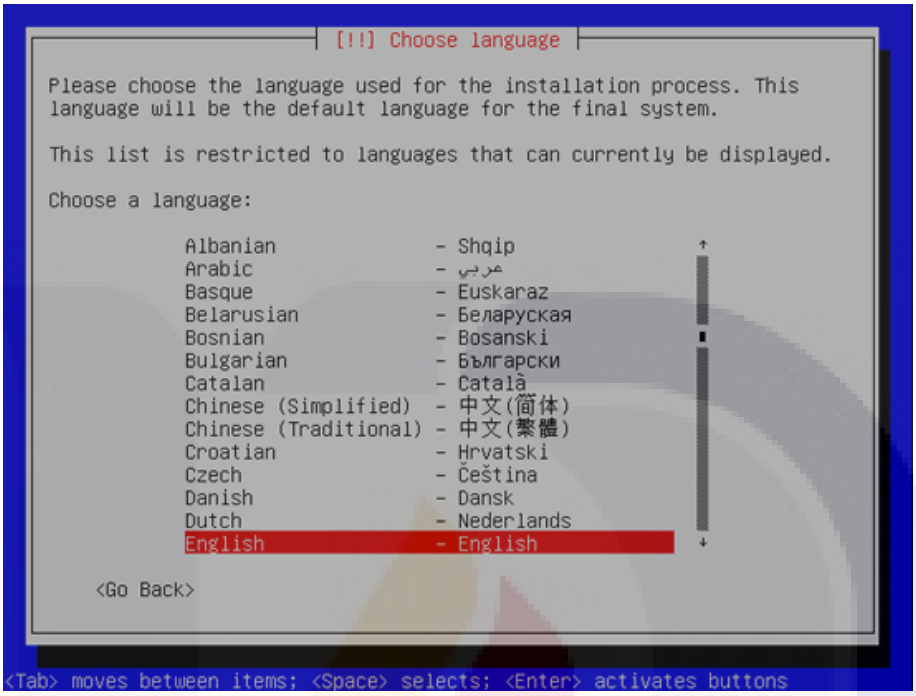
1. Se insertó el CD de instalación de Ubuntu Server 8.04.2 y se arrancó desde él.



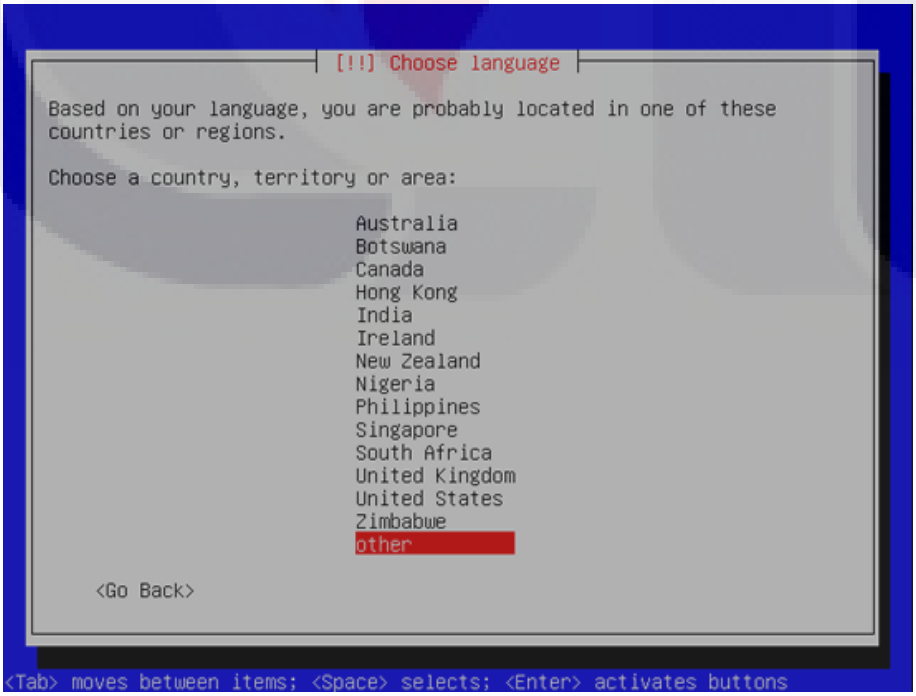
2. Se seleccionó la instalación a disco duro “Install Ubuntu Server”:



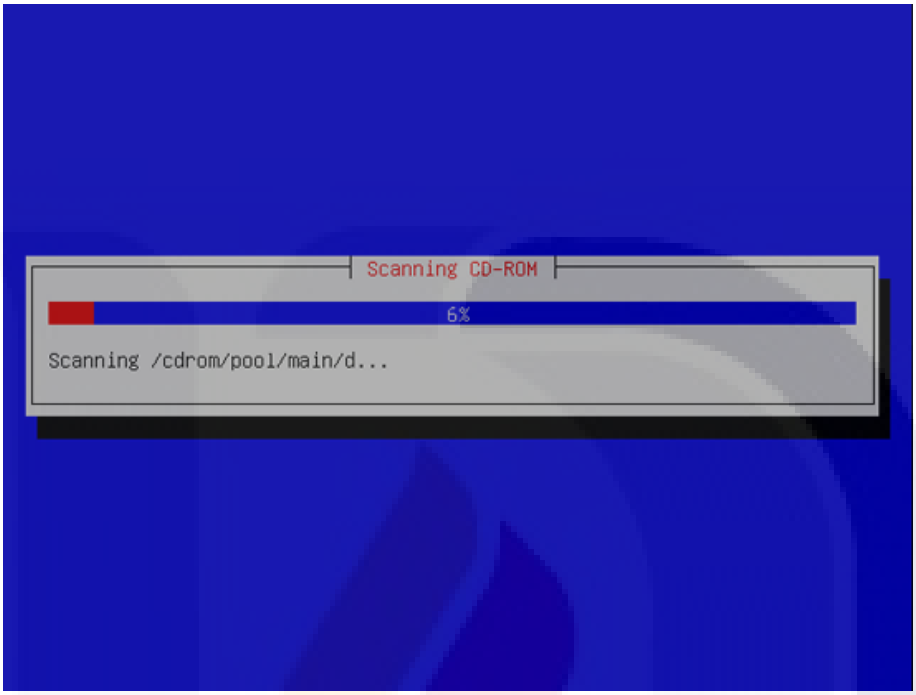
3. Al iniciar la instalación se eligió nuevamente el idioma Inglés:

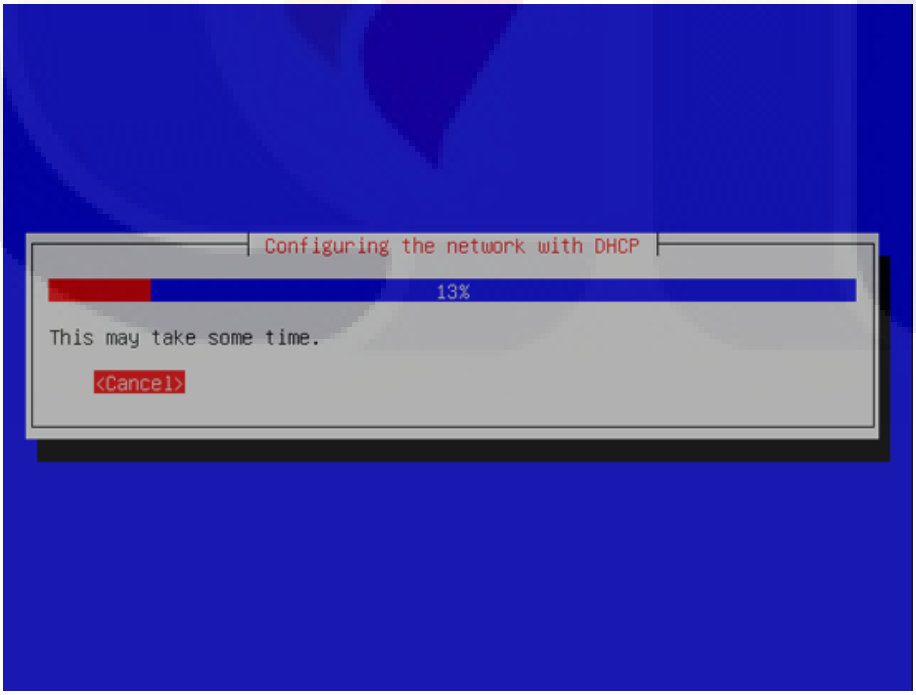


4. Se eligió la ubicación:

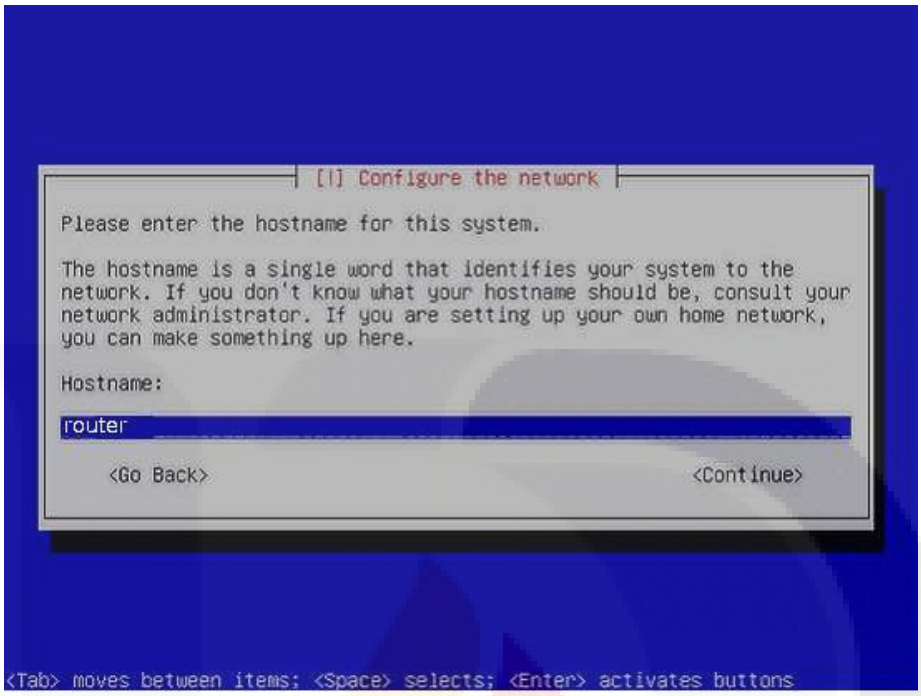


5. El instalador verificó el CD de instalación además del hardware, finalmente se configuró la red utilizando el servicio de DHCP:

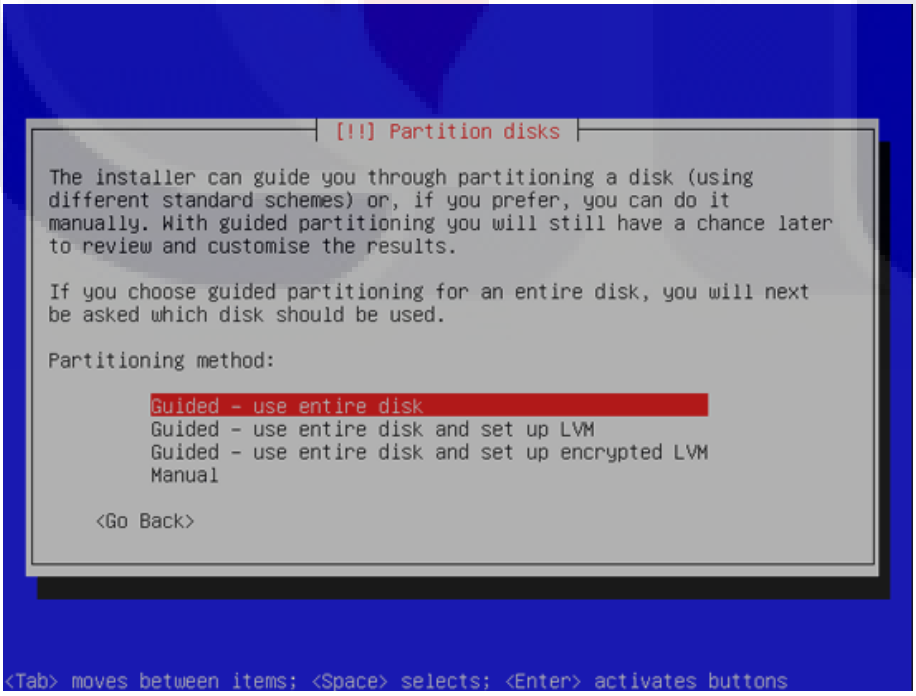




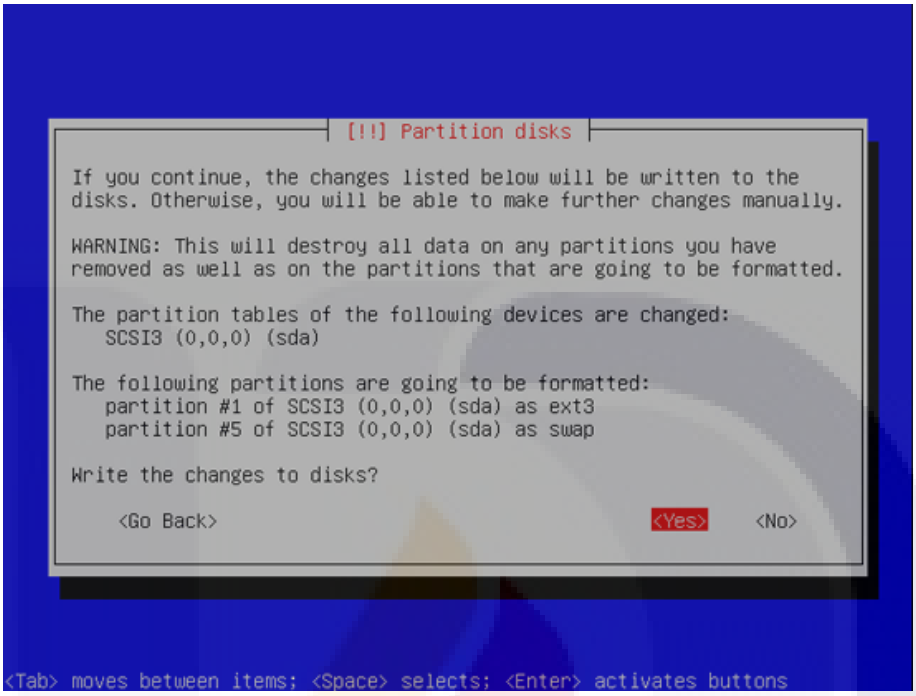
Se escribió el nombre del anfitrión, el sistema fue llamado router.uaa.mx:



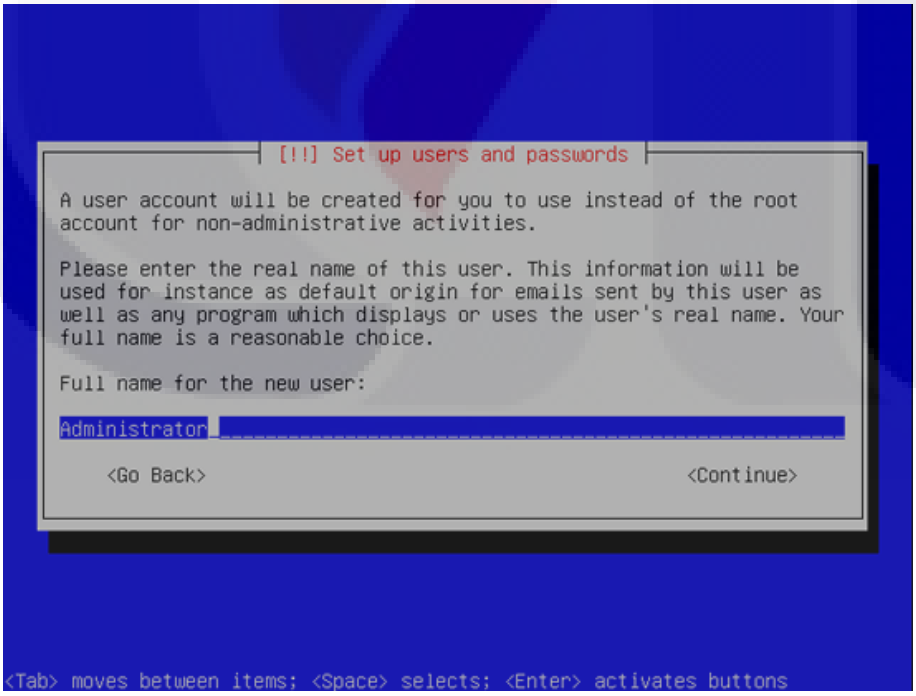
6. Por simplicidad solo se creó una gran partición (con el punto de montaje /) y una pequeña partición para área de intercambio (swap) para ello se eligió “Guided - use entire disk”:

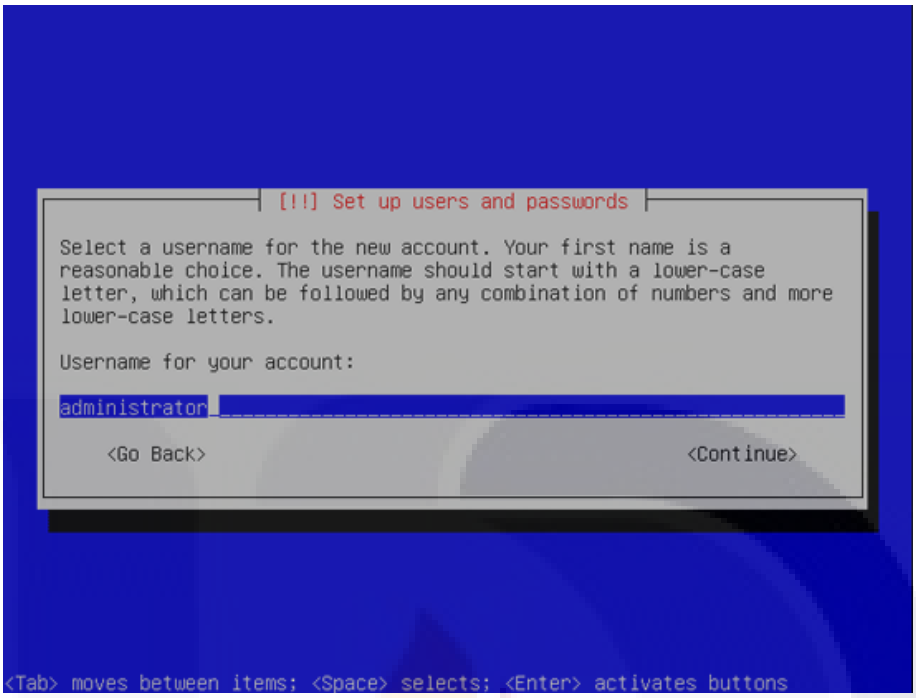


Al terminar el particionado se nos preguntó si se deseaba escribir los cambios en el disco. Al seleccionar <Yes> nuestras particiones fueron físicamente creadas:

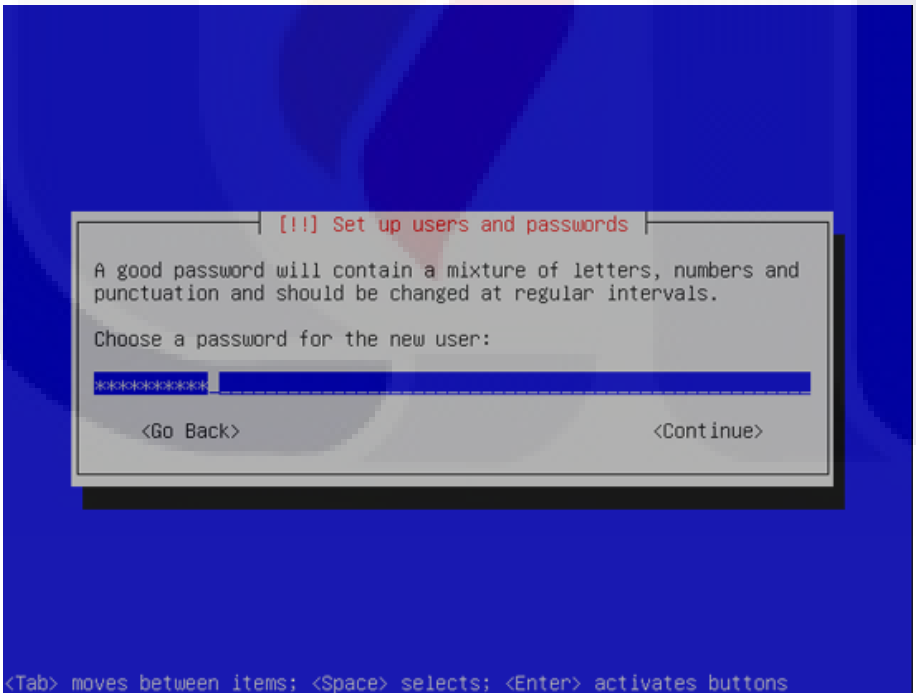


7. Se creó el usuario administrator con el nombre de usuario Administrator:

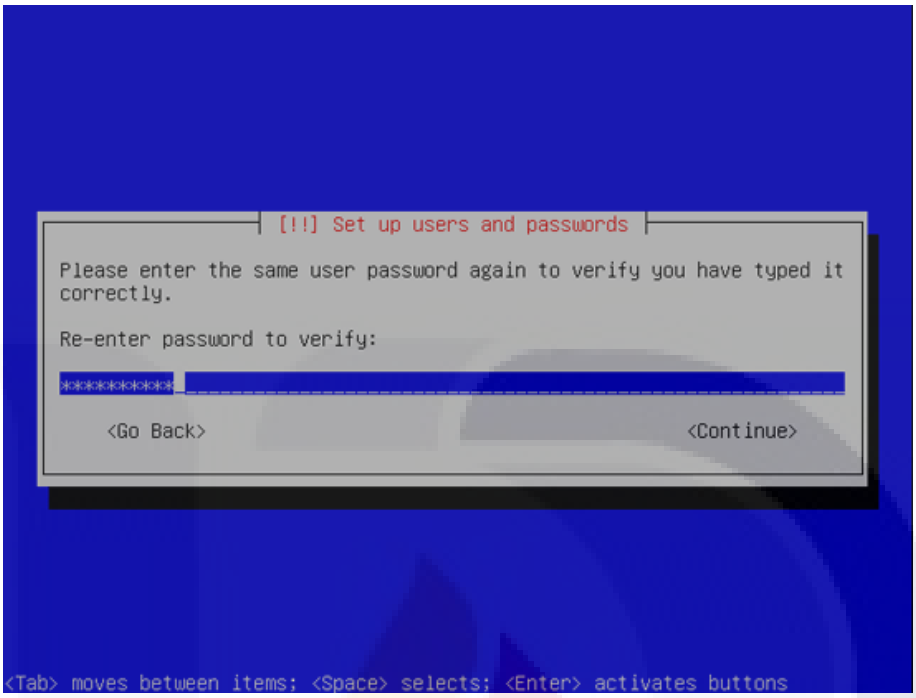




En esta pantalla se escribió la contraseña:

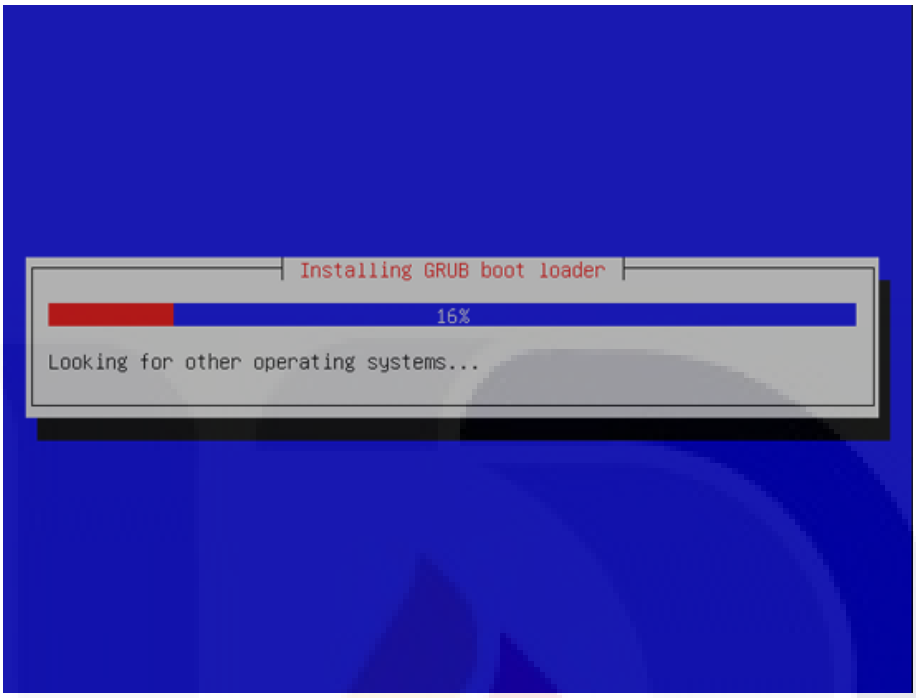


Se reescribió la contraseña elegida:

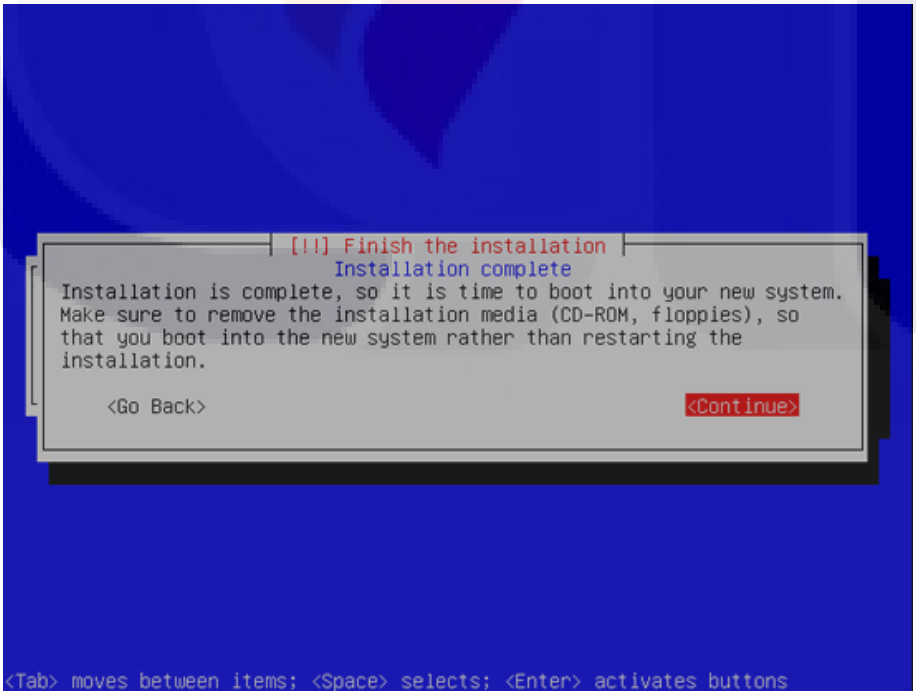


8. El único servicio seleccionado para su instalación fue OpenSSH, este permitió conectarnse inmediatamente al sistema utilizando un cliente SSH:

9. Se instaló el arrancador GRUB:



10. La instalación del sistema base finalizó, por lo que fue retirado el CD de instalación y se reinició el sistema:



11. Activación de la cuenta root: Después del reinicio solo es posible acceder con la cuenta administrator previamente creada. Debido a que los siguientes procedimientos de configuración deben de ejecutarse con la cuenta de root, fue necesario activar dicha cuenta.

Se activó la cuenta de root asignándole una contraseña:

```
root@router:~#  sudo passwd root
```

Se inició sesión con root ejecutando:

```
root@router:~#  su
```

2. Configuración de la red.

A este equipo se le instalaron 2 interfaces de red, la cuales fueron reconocidas por el sistema como eth1 y eth2, eth1 se configuró para usar con el servicio de DHCP y conectarse al Laboratorio, eth2 se configuró para conectarse a Internet.

Para lo anterior se editó el archivo /etc/network/interfaces y fue editado de la siguiente forma:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# This is a list of hotpluggable network interfaces.
# They will be activated automatically by the hotplug subsystem.
mapping hotplug
        script grep
        map eth1 eth2

#Conecta a Laboratorio
auto eth1
iface eth1 inet static
        address 172.16.1.254
        netmask 255.255.255.0
        network 172.16.1.0
        broadcast 172.16.1.255

# Conecta a Internet
auto eth2
iface eth2 inet static
        address 192.168.1.1
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.254
```

El servicio de red fue reiniciado con el siguiente comando:

```
root@router:~#  /etc/init.d/networking restart
```

Se verificó que la configuración anterior se halla activado correctamente ejecutando el comando *ifconfig*:

```
root@router:/# ifconfig
eth1      Link encap:Ethernet  HWaddr 00:14:d1:13:e5:54
          inet addr:172.16.1.254  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::214:d1ff:fe13:e554/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8623 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10812 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2099984 (2.0 MB)  TX bytes:10212415 (9.7 MB)
          Interrupt:5 Base address:0xd000

eth2      Link encap:Ethernet  HWaddr 00:0a:e6:d9:60:d6
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20a:e6ff:fed9:60d6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15497 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8038 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10457291 (9.9 MB)  TX bytes:1965610 (1.8 MB)
          Interrupt:5 Base address:0xd400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

3. Actualización del sistema.

1. Configuración del nombre a nuestro equipo:

Se editó el archivo /etc/hosts. Para dejarlo de la siguiente forma:

```
127.0.0.1       localhost.localdomain   localhost
192.168.1.1     router.uaa.mx     router
```

2. Se editó el archivo /etc/apt/sources.list y se actualizó la instalación de Ubuntu, Para ello fue necesario lo siguiente;

- Comentar o remover la línea de la instalación desde CD
- De comentar las líneas para los repositorios universe y multiverse.

El archivo /etc/apt/sources.list quedó de la siguiente manera:

```
deb http://de.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy main restricted
## Major bug fix updates produced after the final release of the distribution.
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu team, and may not be under a free l
deb http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
## Uncomment the following two lines to add software from the 'backports' repository.
## N.B. software from this repository may not have been tested as extensively as that contained in the main release, although
### it includes newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository. This software is not part of Ubuntu, but is offered by Canonical and the respective vendors as a
### service to Ubuntu users.
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

3. Se actualizó la base de datos de paquetes de apt:

```
root@router:~#  apt-get update
```

4. Se actualizaron los paquetes instalados:

```
root@router:~#  apt-get upgrade
```

4. Instalación del cliente WMI.

El Cliente WMI fue necesario para leer los valores de CPU y memoria RAM del equipo B Cortafuegos en Windows.

1. Se actualizaron los paquetes disponibles:

```
root@router:~#  apt-get update
```

2. Se Instaló el cliente WMI:

```
root@router:~#  apt-get install wmi-client
```

5. Instalación y configuración de DHCP.

El servicio de DHCP fue instalado para facilitar el laboratorio ya que asigna el IP al equipo B Cortafuegos en Windows así como al equipo C, para el equipo B Cortafuegos en GNU/Linux y equipo B sistema integrado no fue necesario ya que funcionan en modo Puente.

1. Se instaló el servicio de DHCP:

```
root@router:/# aptitude install dhcp3-server
```

2. Se configuró el servicio de DHCP:

Como se mencionó el servicio de DHCP fue configurado en eth1, por lo que fue necesario activar el servicio para eth1, para lo cual se editó el archivo /etc/default/dhcp3-server y se dejó de la siguiente forma:

```
# Defaults for dhcp initscript  sourced by /etc/init.d/dhcp  installed at /etc/default/dhcp3-server by
the maintainer scripts

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"
```

El archivo principal del servicio DHCP se configuró en el archivo /etc/default/dhcpd.conf, este archivo se dejó de la siguiente forma:

```
# Defaults for dhcp initscript
# sourced by /etc/init.d/dhcp
# installed at /etc/default/dhcp3-server by the maintainer scripts


# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="eth1"
root@router:/# cat /etc/dhcp3/dhcpd.conf
#dhcpd.conf

option domain-name "maestria.uaa.mx";
option domain-name-servers 200.56.200.1,192.168.1.254;
option netbios-name-servers  172.16.1.254;

#dns-update-style none;
default-lease-time 604800;
max-lease-time 720000;

authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

subnet 172.16.1.0 netmask 255.255.255.0 {
        option broadcast-address 172.16.1.255;
        option routers 172.16.1.254;
        range 172.16.1.10 172.16.1.20;
      }
```

Se puede observar que el laboratorio quedó trabajando en el segmento 172.16.1.10 al 172.16.1.20

6. Configuración de NAT para que los equipos del laboratorio salieran a Internet.

Con la conexión hacia Internet fue posible instalar los paquetes necesarios por los equipos B y el equipo C. Para lograr lo anterior se creo un script de arranque

llamado S98tesis y almacenado dentro de la carpeta /etc/rc2.d/ el contenido de dicho script es:

```
/sbin/modprobe ip_conntrack_ftp
/sbin/modprobe ip_nat_ftp
echo 1 > /proc/sys/net/ipv4/ip_forward
/sbin/iptables -F
/sbin/iptables -X
/sbin/iptables -Z
/sbin/iptables -P INPUT ACCEPT
/sbin/iptables -P OUTPUT ACCEPT
/sbin/iptables -P FORWARD ACCEPT
/sbin/iptables -t nat -F
/sbin/iptables -t nat -A POSTROUTING -o eth2 -j MASQUERADE
```

7. Instalación de D-ITG (Distributed Internet Traffic Generator).

En este equipo solo se utilizó ITGRecv ya que este equipo fue el receptor, posteriormente al finalizar el experimento fue utilizado el programa ITGDec para obtener los datos del jitter del primer flujo VoIP.

Se ingresó al directorio /usr/src

```
root@router:~# cd /usr/src
```

Se Descargó el código fuente de D-ITG:

```
root@router:/usr/src# wget http://www.grid.unina.it/software/ITG/codice/D-ITG-2.6.1d.zip
```

Se Desempaquetó el archivo:

```
root@router:/usr/src# unzip D-ITG-2.6.1d.zip
```

Se ingresó el directorio /usr/src/d-itg-2.6.1d:

```
root@router:/usr/src# cd d-itg-2.6.1d
```

Se compiló y se instaló:

```
root@router:/usr/src/d-itg-2.6.1d# make
root@router:/usr/src/d-itg-2.6.1d# make install
```

Los ejecutables quedaron en la carpeta bin, se crearon enlaces lógicos hacia la carpeta /usb/bin de los programas ITGRecv, ITGLog e ITGDec para ejecutarlos desde cualquier carpeta:

```
root@router:/usr/src/d-itg-2.6.1d#  cd /usr/bin/
root@router:/usr/bin#  ln -s / usr/src/d-itg-2.6.1d/bin/ITGRec  .
root@router:/usr/bin#  ln -s / usr/src/d-itg-2.6.1d/bin/ITGLog  .
root@router:/usr/bin#  ln -s / usr/src/d-itg-2.6.1d/bin/ITGDec  .
```

Instalación y Configuración del equipo B

Cortafuegos en Windows –: Windows2003 y Websense Express.

En esta sección se muestra como fue instalado y configurado el equipo B.
– Cortafuegos en Windows –. El Sistema Operativo instalado fue Windows 2003.
Además se utilizó el siguiente software:

- Websense Express.

El equipo utilizado fue el mismo donde se instalo el cortafuego de GNU/Linux, las características de hardware fueron las siguientes:

- CPU : Intel(R) Pentium(R) 4 CPU 1.80GHz, tamaño de cache : 256 KB
- memoria DDR-1 480 MB
- Tarjetas de Red
  - eth2 Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
  - eth3 Intel Corporation 82557/8/9/0/1 Ethernet Pro 100 (rev 02)
  - eth4 VIA Technologies, Inc. VT6102 [Rhine-II] (rev 74)

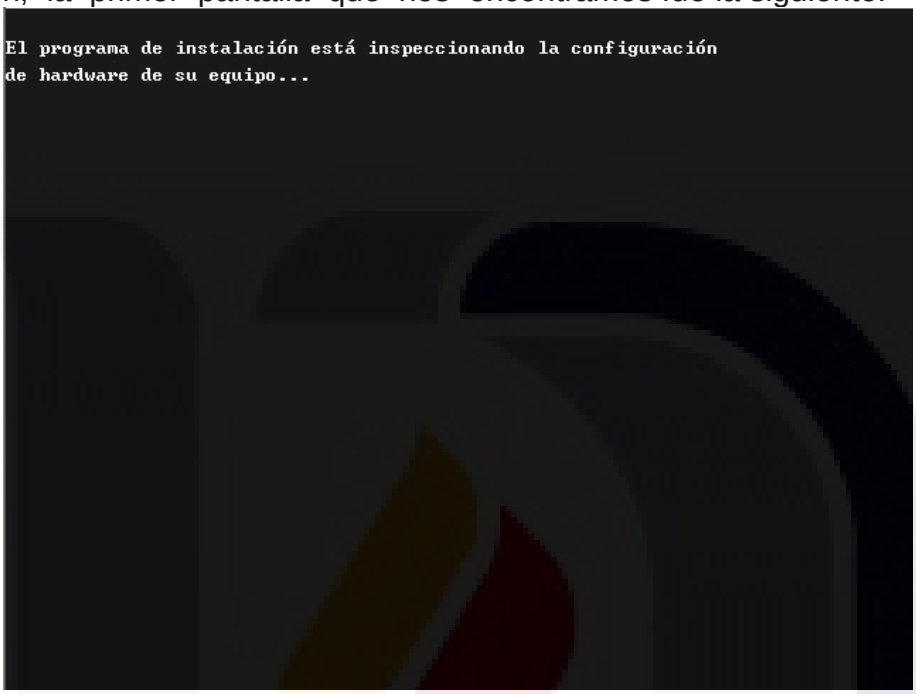
Requerimientos

Para la instalación fue necesario lo siguiente:

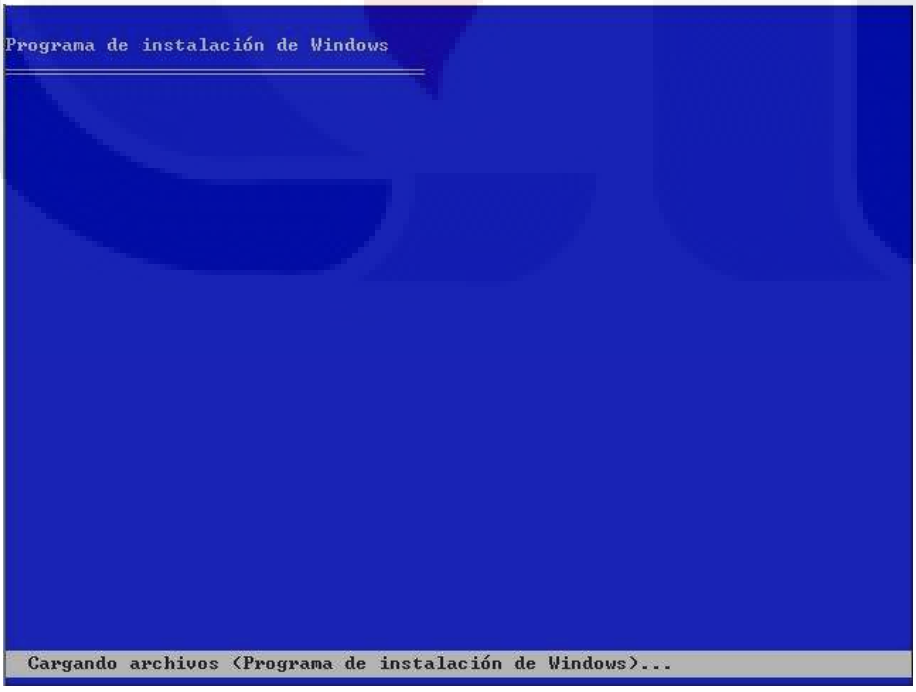
4. El CD de instalación de Windows Server 2003:
5. Websense Express.

1. Instalación de Windows Server 2003.

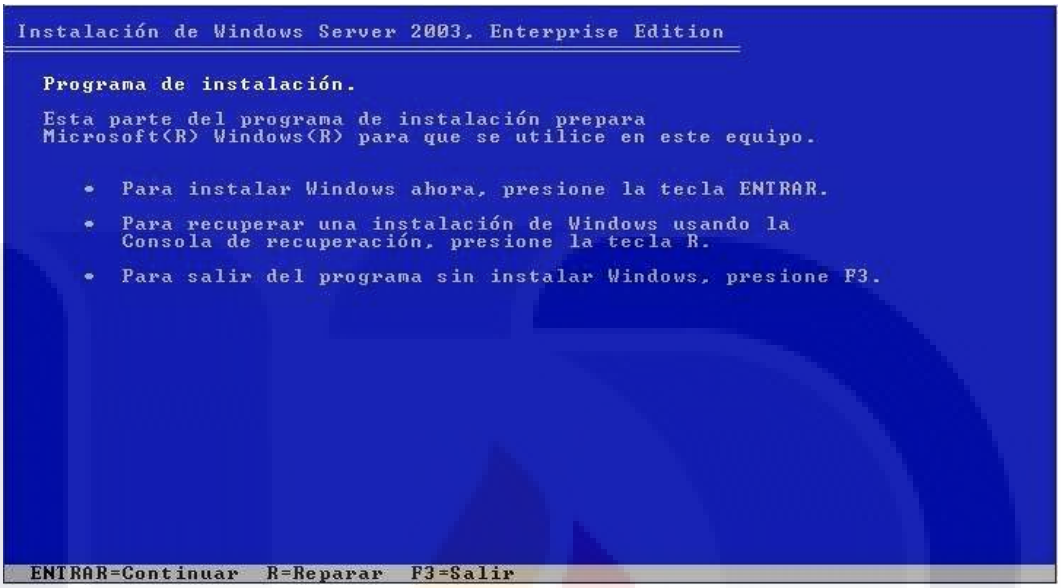
1 Insertó el CD de instalación de Windows Server 2003: Se encendió la PC y se colocó el CD de Windows Server 2003, al arrancar desde el CD se inició la instalación, la primer pantalla que nos encontramos fue la siguiente:



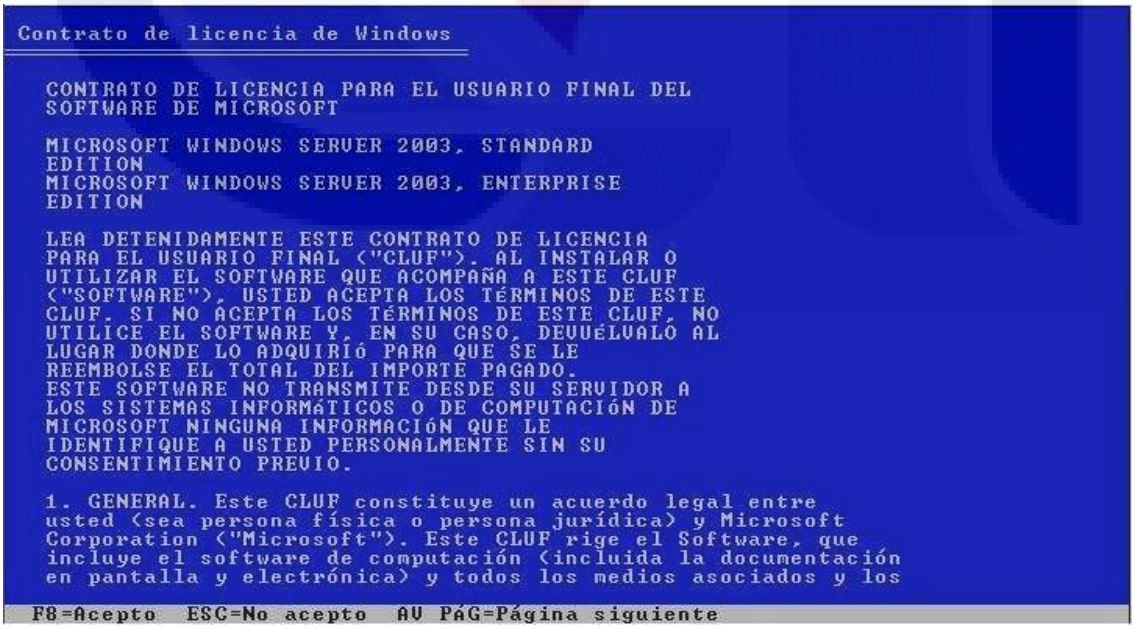
2. Se inició la carga de los archivos necesarios para iniciar la instalación.



3. Una vez cargados los archivos necesarios se presionó la tecla Enter.



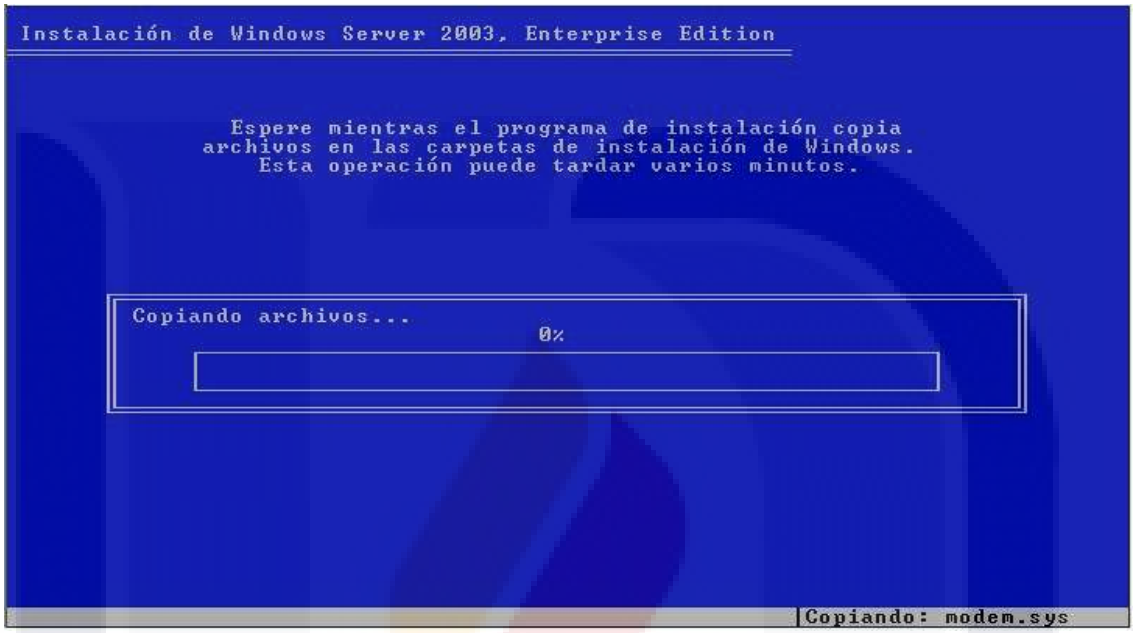
4. Se aceptó la licencia presionando F8



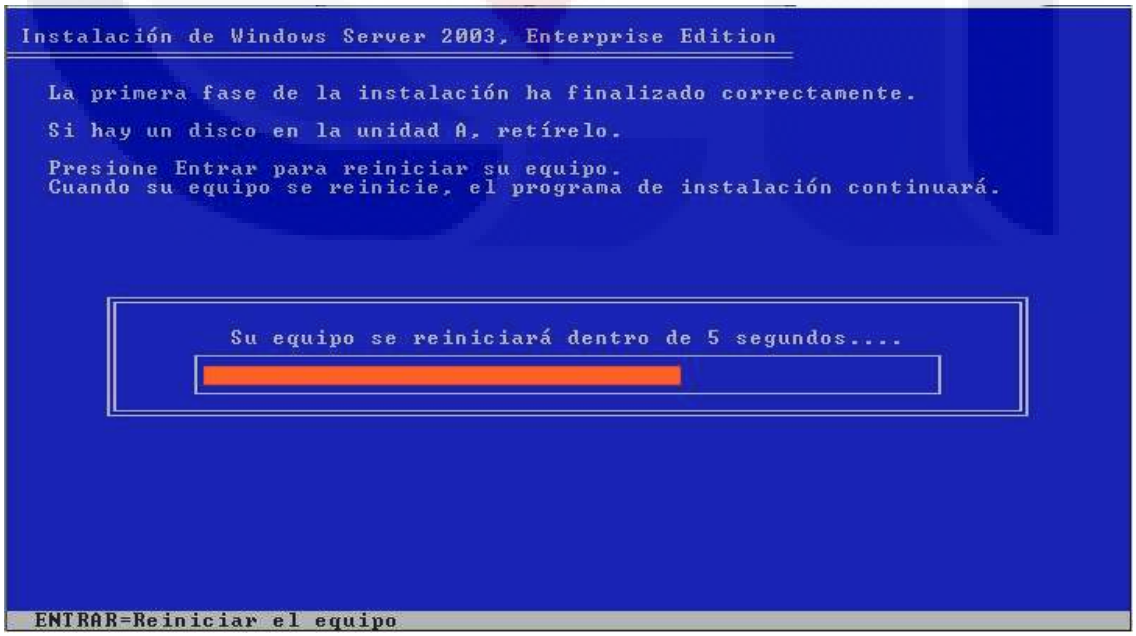
5. Se seleccionó el espacio no particionado para la instalación:

6. Una vez creada la partición se seleccionó la tercer opción: “Formatear la partición utilizando el sistema de archivos NTFS” y se presionó la tecla Enter para continuar.

7. Se nos presentó:
   - EL porcentaje de formateo.
   - Se examinó el disco duro.
8. Se inició el copiado de archivos a disco duro:



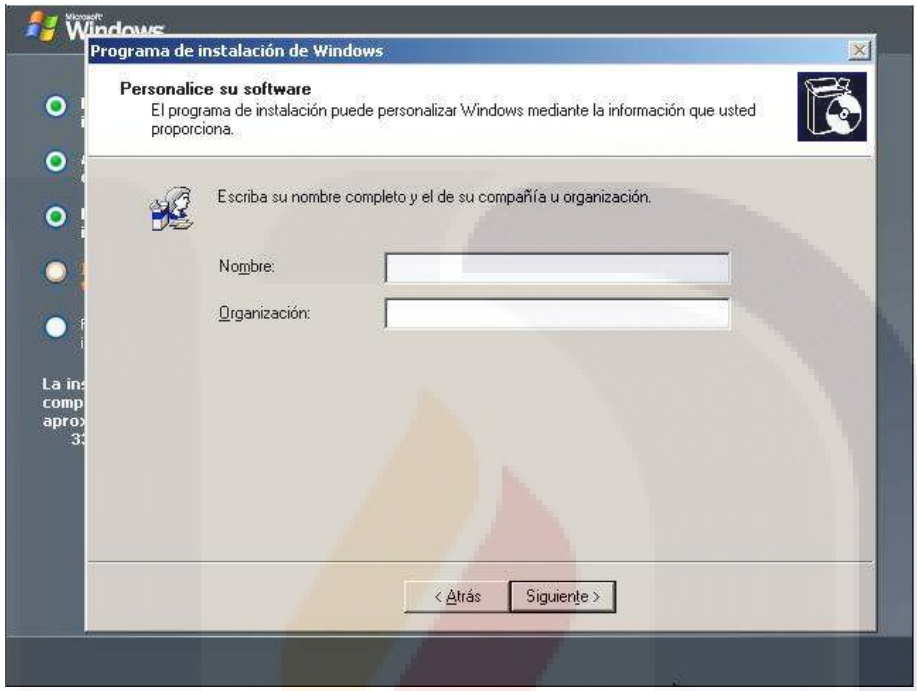
9. Al terminar el copiado de archivos a disco duro, el quipo se reinició automáticamente.



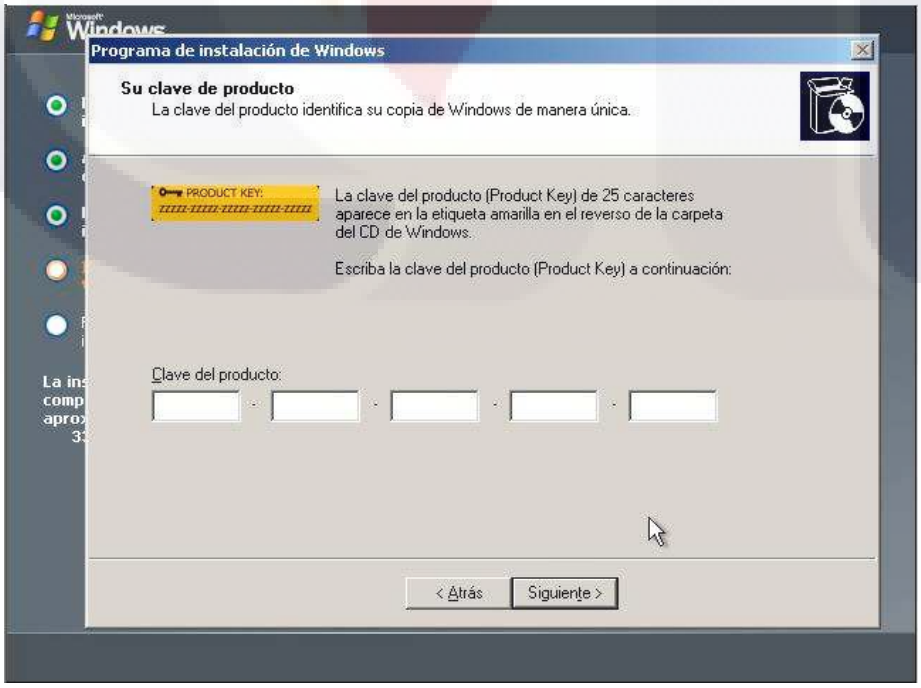
10. Al reiniciar el equipo este fue arrancado desde disco duro no desde el CD, la siguiente imagen se nos presentó al estar arrancando por primera vez desde disco duro.

11. Se inició el proceso de instalación gráfico:

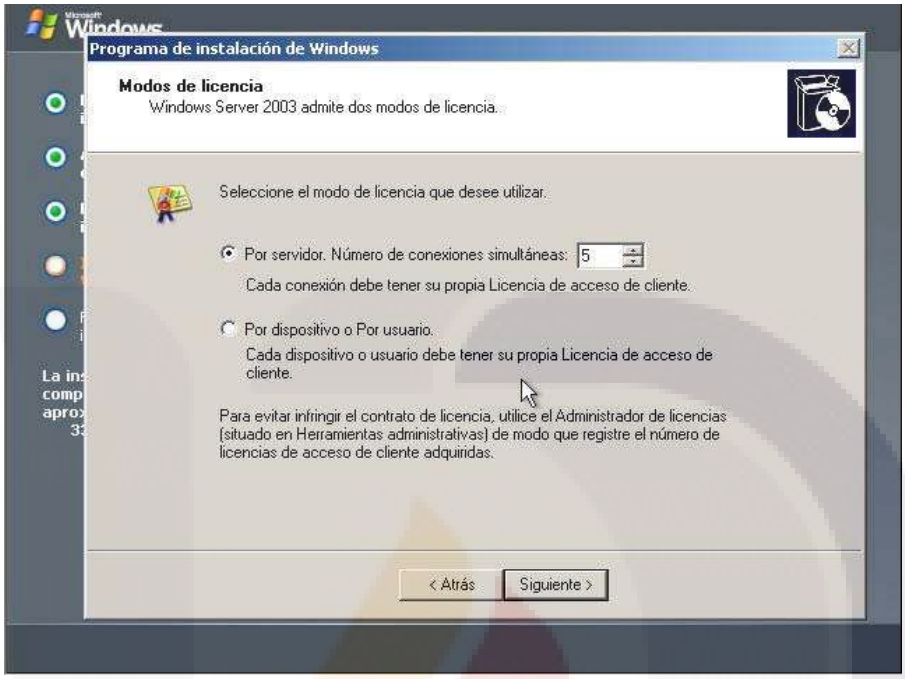
12. Se iniciaron las configuraciones:
    - Lo primero que se configuró fue la “Configuración regional y de idioma” hay se seleccionó español.
    - Seguidamente se configuró el nombre de usuario y el de la organización.



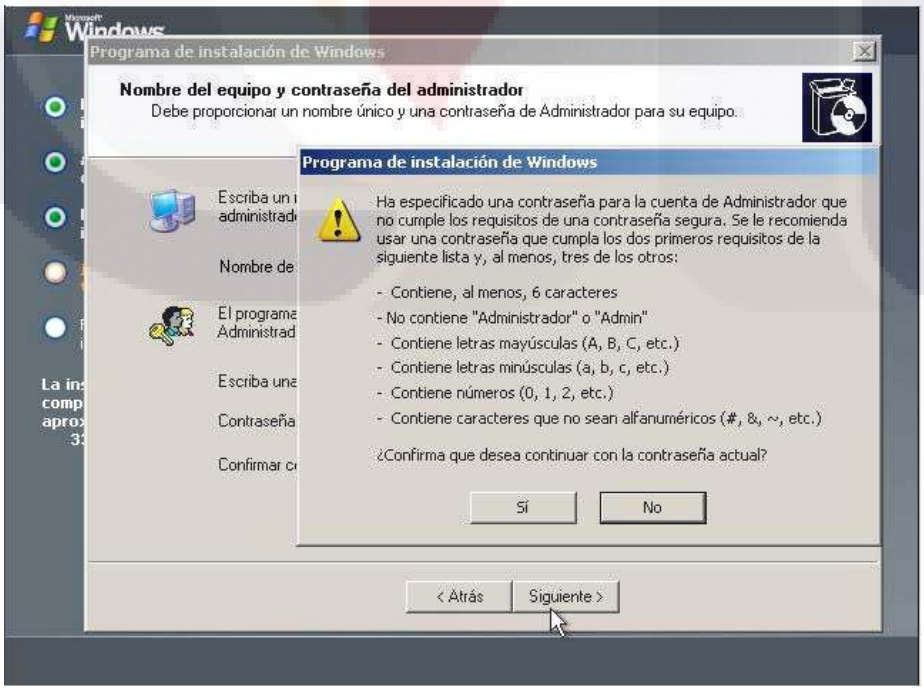
13. Se Introdujo la clave de instalación de Windows.



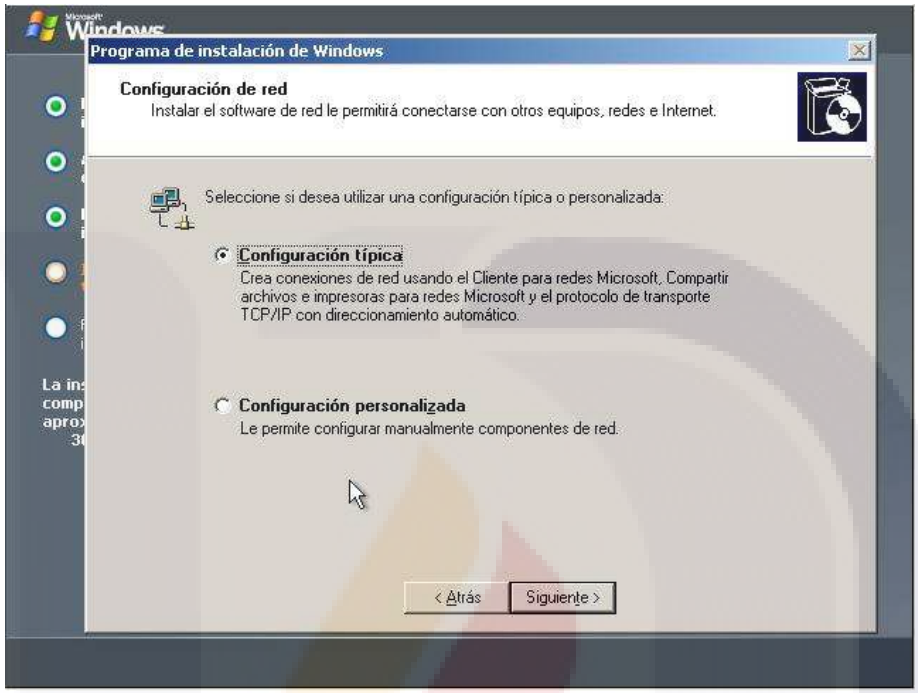
14. Se nos preguntó el tipo de licenciamiento a usar para conectarnos a los servicios instalados, se seleccionó la primera opción y se presionó Siguiente.



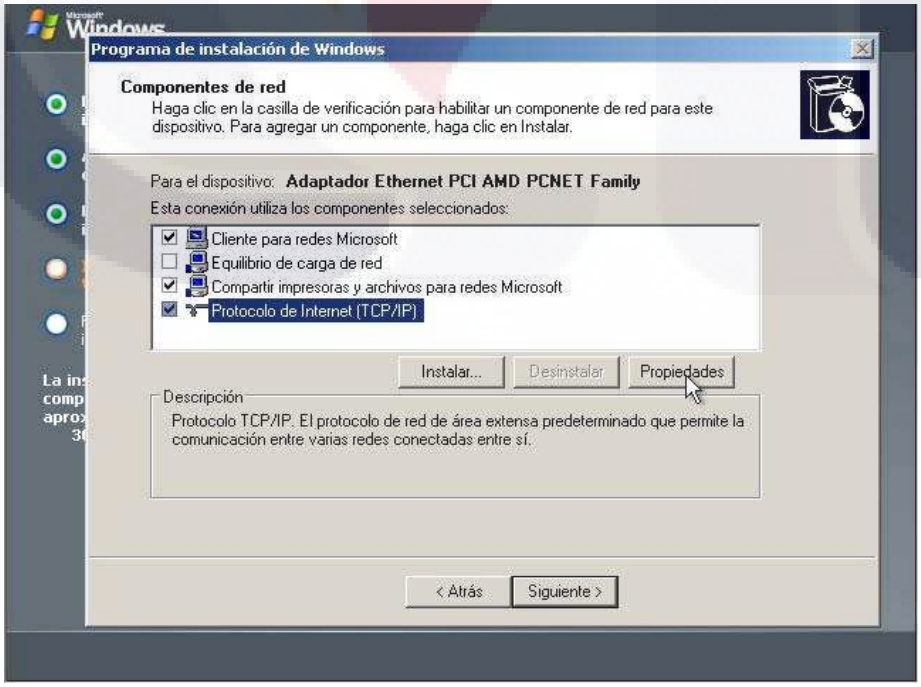
15. Se le puso un nombre a nuestro equipo, también se estableció la contraseña del Administrador.



16. Configuración de la red: Se seleccionó “Configuración típica” y se presionó Siguientes.

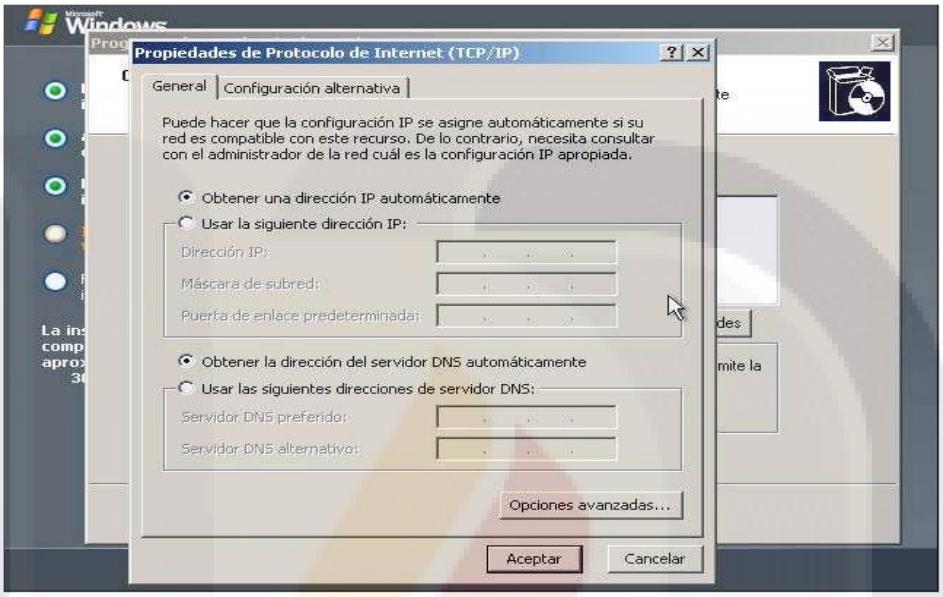


17. Se seleccionó “Protocolo de Internet (TCP/IP)” y se dio clic en “Propiedades”:

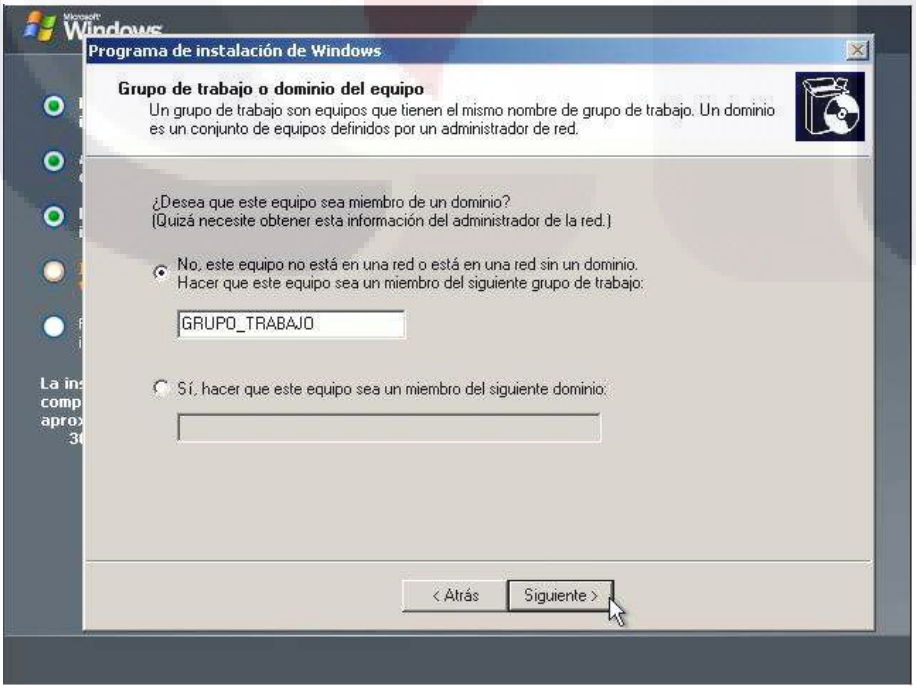


18. Debido a que en el equipo A fue un servidor DHCP solo se verificó la configuración para:

- Obtener una dirección IP automáticamente.
- Obtener la dirección del servidor DNS automáticamente.

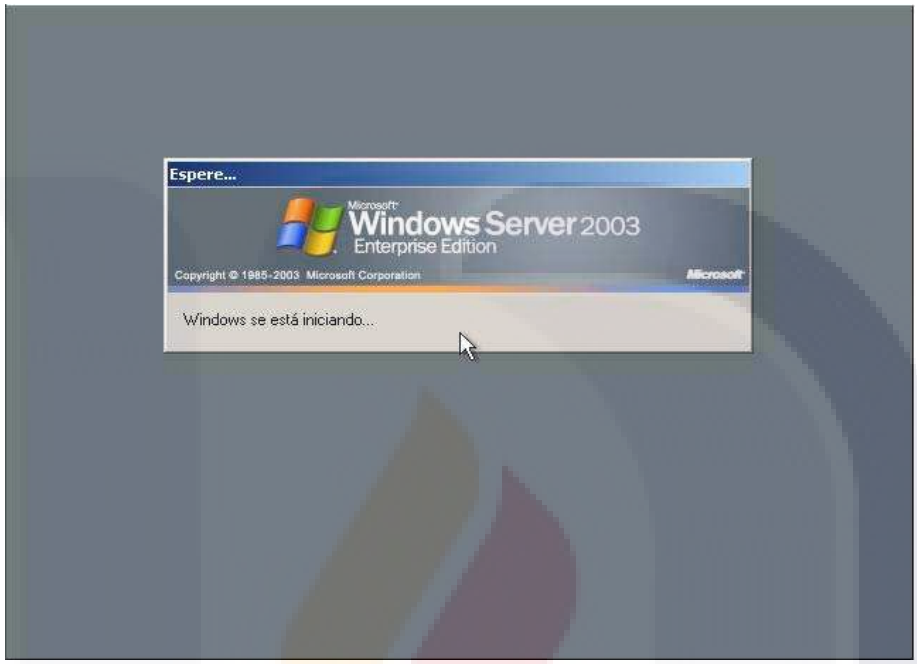


19. El grupo de trabajo no fue indispensable en nuestra configuración por los que se seleccionó la opción por defecto y se presionó Siguiente:

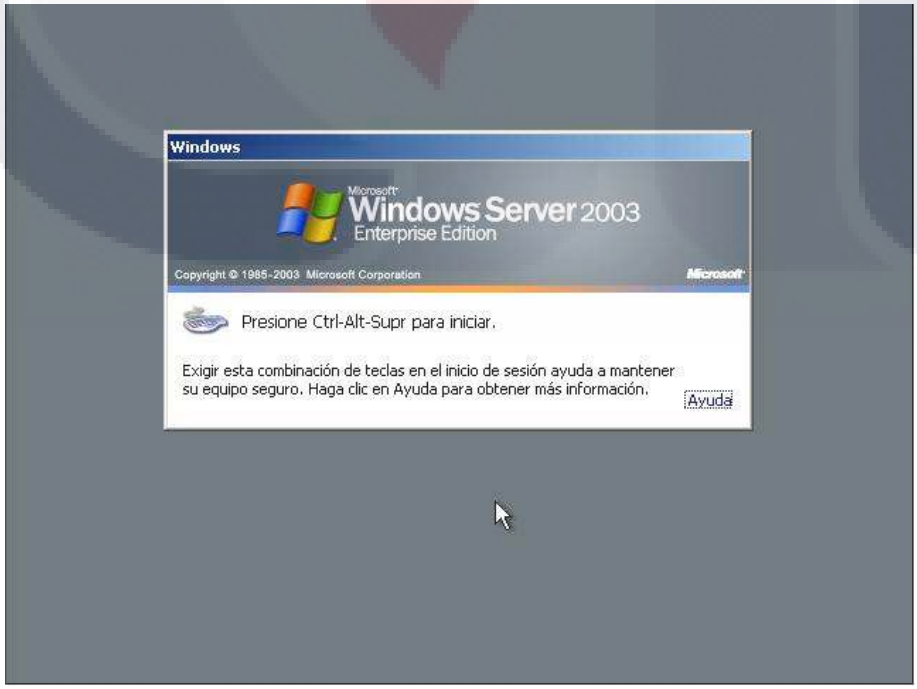


20. Finalizó la instalación con lo siguiente:

- Se guardó la configuración.
- Se borraron los archivos temporales usados en la instalación y reinició el equipo:



21. Windows Server 2003 fue instalado, pulsando la secuencia de teclas indicada en la imagen se inició sesión.



2. Instalación de WebSense.

1. Solicitud de la licencia y descarga del software de instalación:

Se consiguió una licencia de prueba por 30 días de Websense Express en el siguiente enlace: http://www.websense.com/evaluations/ La licencia de prueba me fue enviada por correo, a continuación se muestra el correo con la licencia:

FREE 30-DAY EVALUATION: WEBSENSE EXPRESS

Dear Americo,

Thank you for evaluating Websense Express free for 30 days.

To begin, click on the link below to access the download page for Websense Express. Please save this email. You will need the evaluation key to activate your software and to receive technical support throughout your evaluation period.

*** Click here to begin download**
* Evaluation key: EVAGKAPANE2CH5B4
* Expiration date: March 8, 2009

Your evaluation key will be activated within one business day of your request. Please note, inaccurate contact information may cause the key to be disabled.

Websense offers free technical support during your evaluation period. To access technical support, you must register for a FREE MyWebsense account. A MyWebsense account also lets you evaluate multiple products using a single evaluation key, and offers unlimited access to the Websense Support Portal and Knowledge Base for product-specific documentation, tutorials, and Knowledge Base articles during your evaluation period.

* Register for a MyWebsense account: www.mywebsense.com
* Visit our online support center: www.websense.com/supportportal

System Requirements

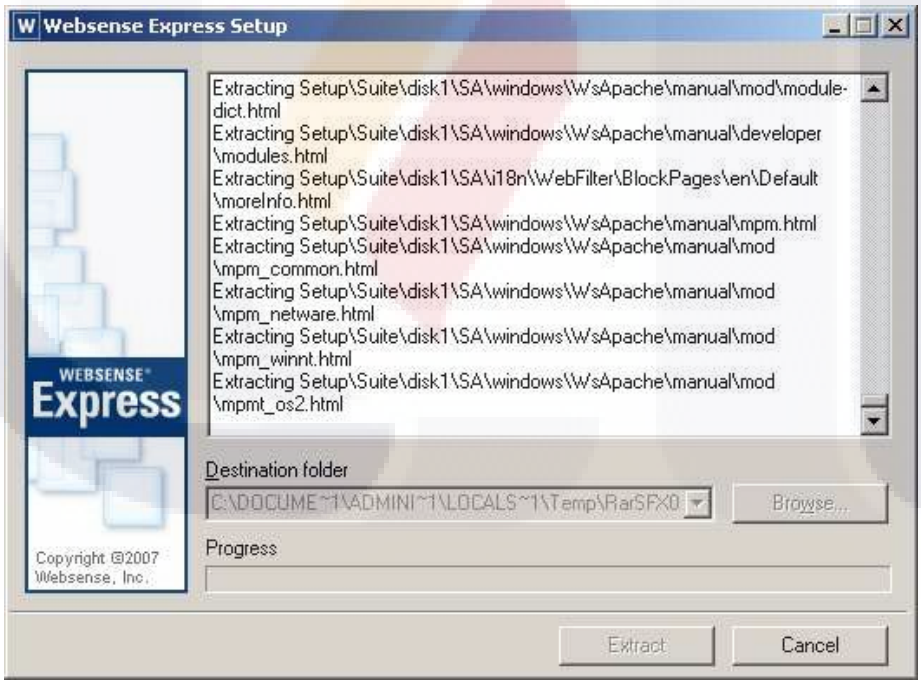
* Dual-Core Xeon 3050, 2.13 GHz
* 85 GB of free disk space
* 2 GB of RAM
* CD-ROM
* 2 network interface cards (NICs)
* Microsoft Windows 2003 Server SP1 or later (32 bit)
* Microsoft Windows 2003 Server R2 or later (32 bit)
* Internet Explorer 6 or later
* Adobe Acrobat Reader 6 or later

El enlace para descargar Websense Express fue proporcionado en el correo anterior, una vez descargado se inició el proceso de instalación.

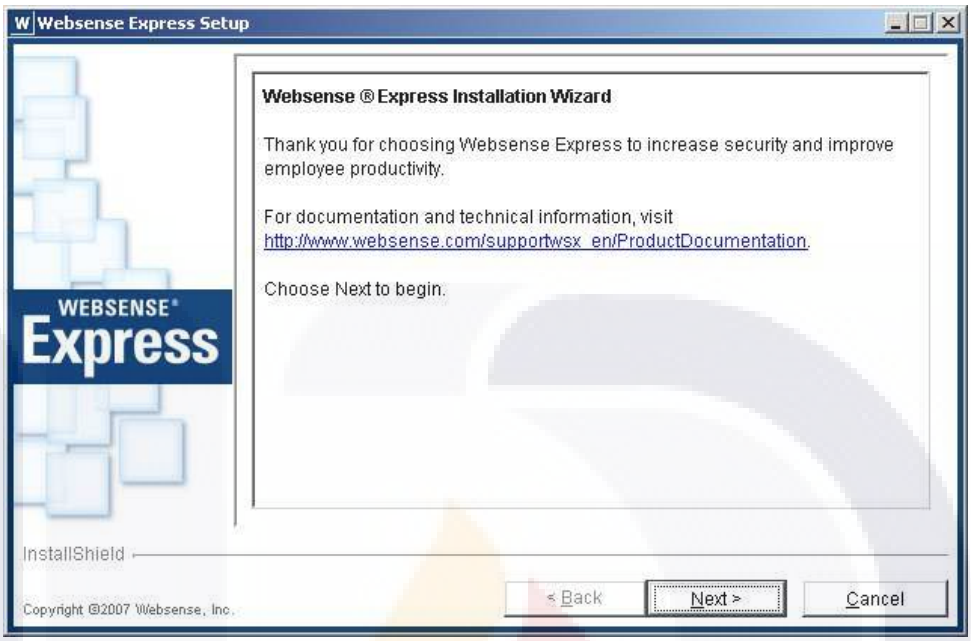
2. Instalación de Websense Express: Inició la instalación dando doble clic en el archivo WebsenseExpress10_Setup.exe descargado:

WSX_User_Guide.pdf	2,284 KB	Adobe Acrobat Doc...	06/02/2009 04:21 p.m.
WSX_QuickSetupCard.pdf	172 KB	Adobe Acrobat Doc...	06/02/2009 04:19 p.m.
WSX_Installation.pdf	307 KB	Adobe Acrobat Doc...	06/02/2009 04:20 p.m.
WebsenseExpress10_Setup.exe	118,023 KB	Application	06/02/2009 03:43 p.m.
WBSN_WebSecurity_Web.pdf	671 KB	Adobe Acrobat Doc...	06/02/2009 03:45 p.m.
Licencia.txt	2 KB	Text Document	06/02/2009 04:27 p.m.
Express_datasheet_FNL.pdf	216 KB	Adobe Acrobat Doc...	06/02/2009 03:45 p.m.
Explorer_Bus_Examples.pdf	1,960 KB	Adobe Acrobat Doc...	06/02/2009 04:22 p.m.
datasheet_wsx_es.pdf	757 KB	Adobe Acrobat Doc...	06/02/2009 03:45 p.m.
datasheet_web_product_family_es.pdf	10,388 KB	Adobe Acrobat Doc...	06/02/2009 03:47 p.m.

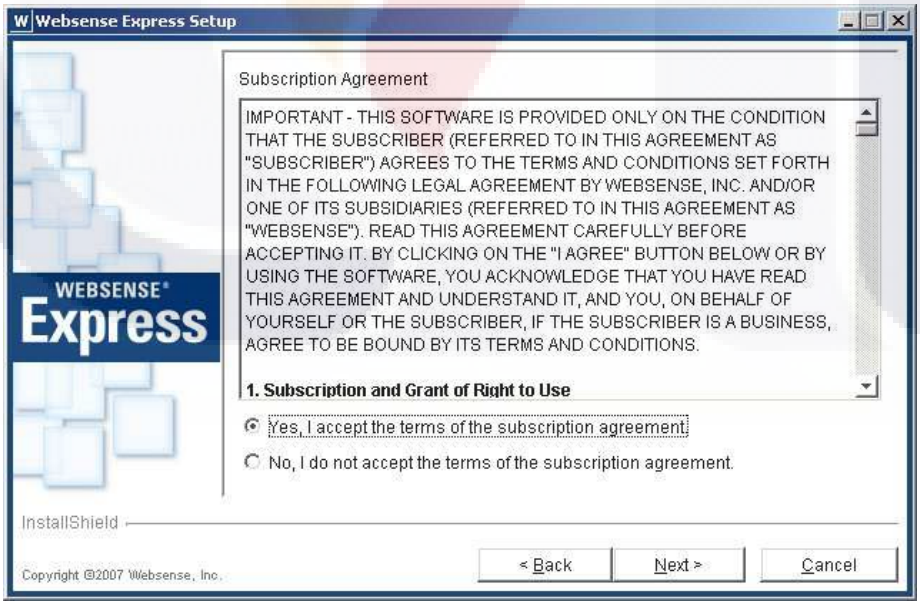
Se inició la descompresión del archivo:



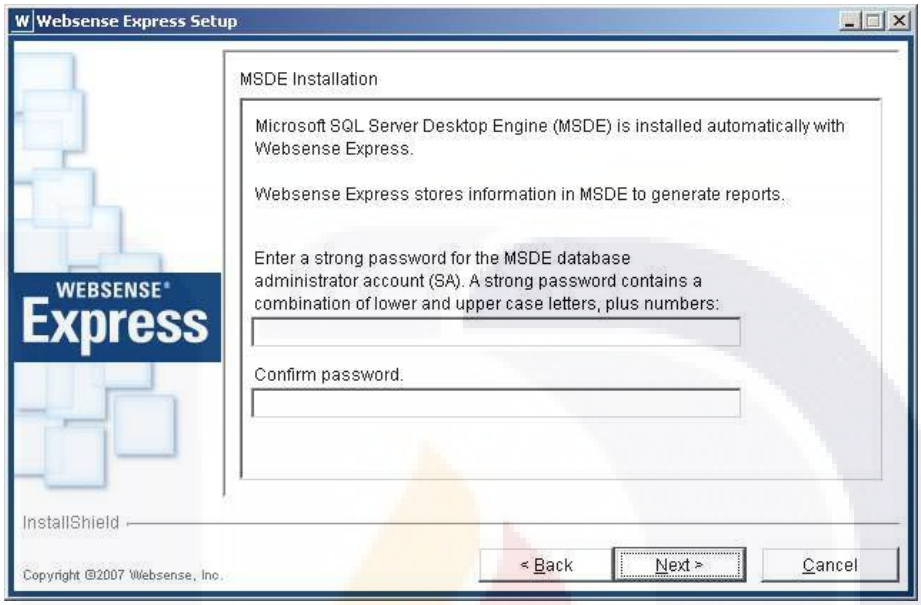
Se presionó el botón Next para continuar con la instalación:



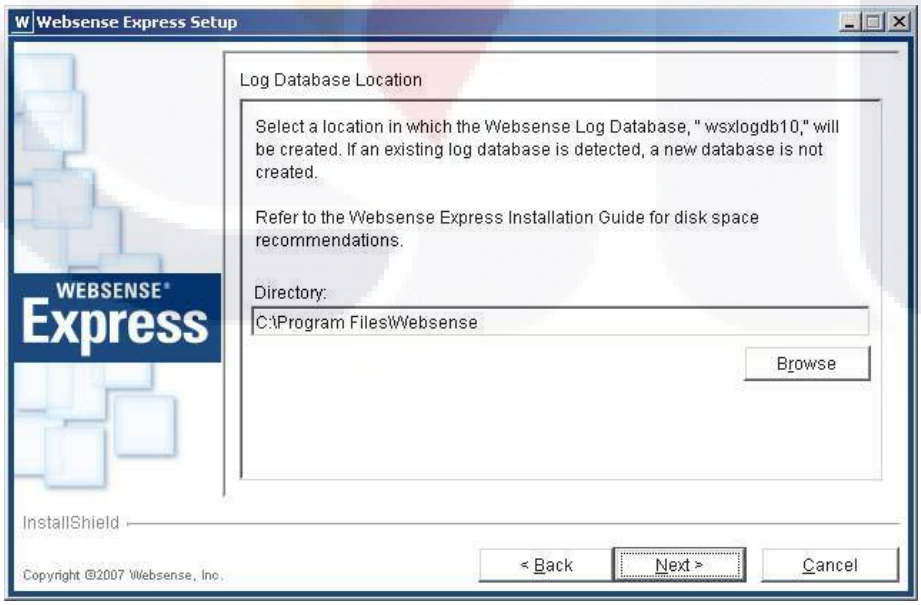
Se aceptaron los términos de uso del software y se presionó el botón Next:



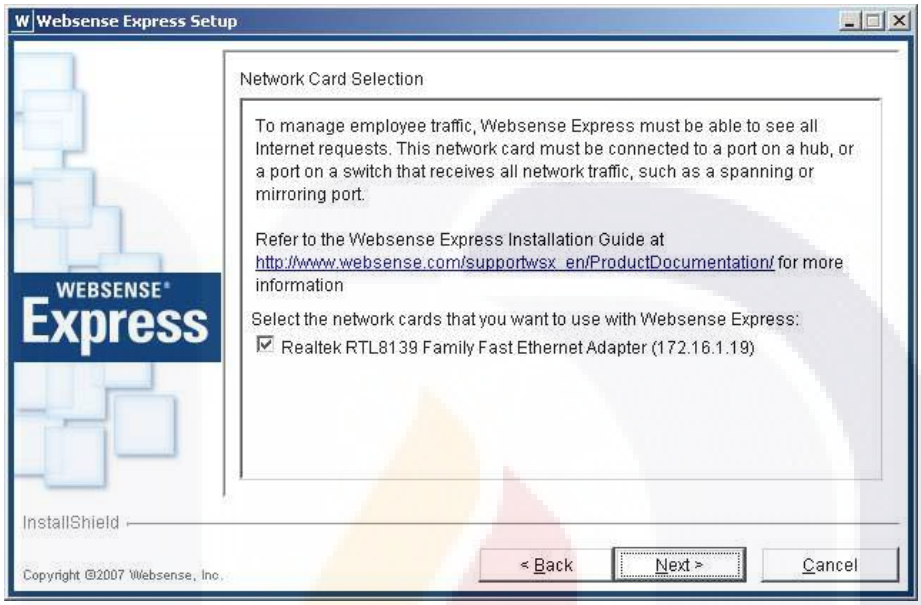
Automáticamente fue instalado el motor de base de datos Microsoft SQL Server Desktop Engine (MSDE). Y se le asignó una contraseña a MSDE:



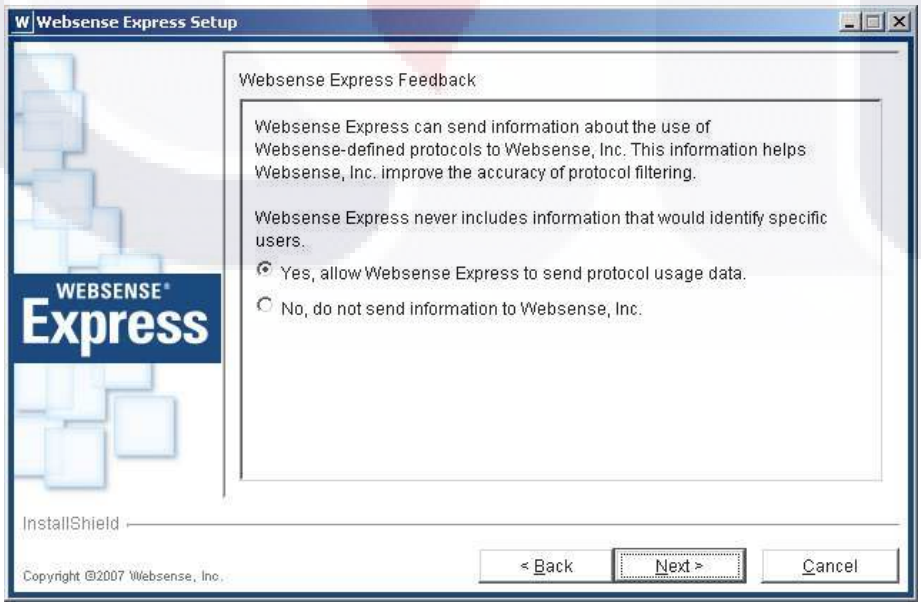
Se aceptó el directorio de instalación que trae por defecto:



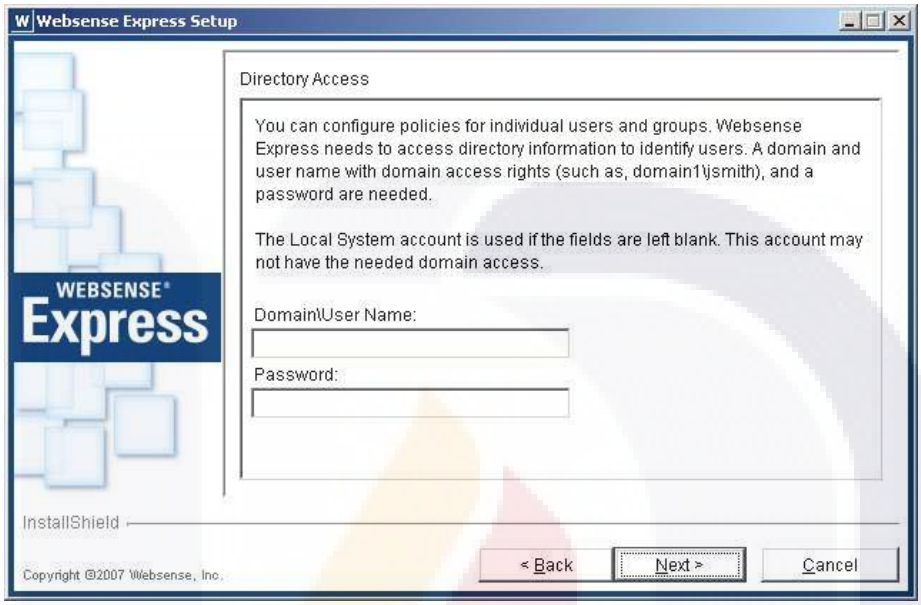
Se seleccionó la interfaz de red para el bloqueo de aplicaciones, el ip de la interfaz fue 172.16.1.19:



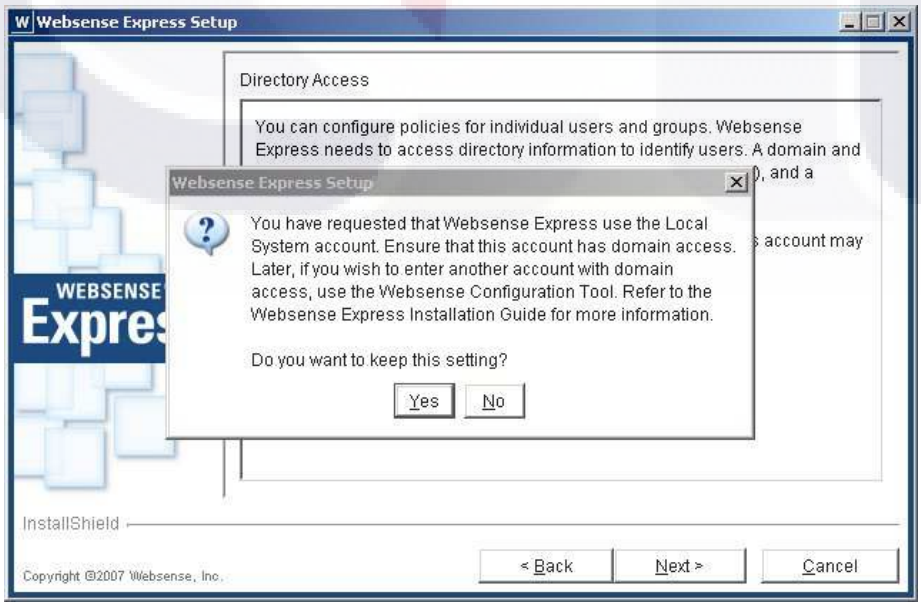
Se aceptó el envío de información de uso de esta instalación a Websense:



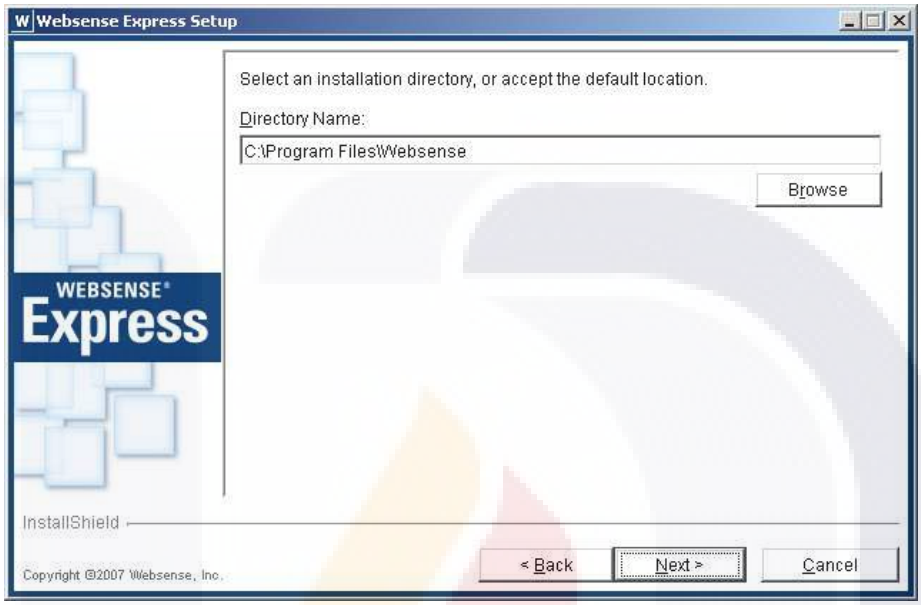
En esta instalación no fue necesario el bloqueo por usuario, así que se dejo en blanco la configuración siguiente:



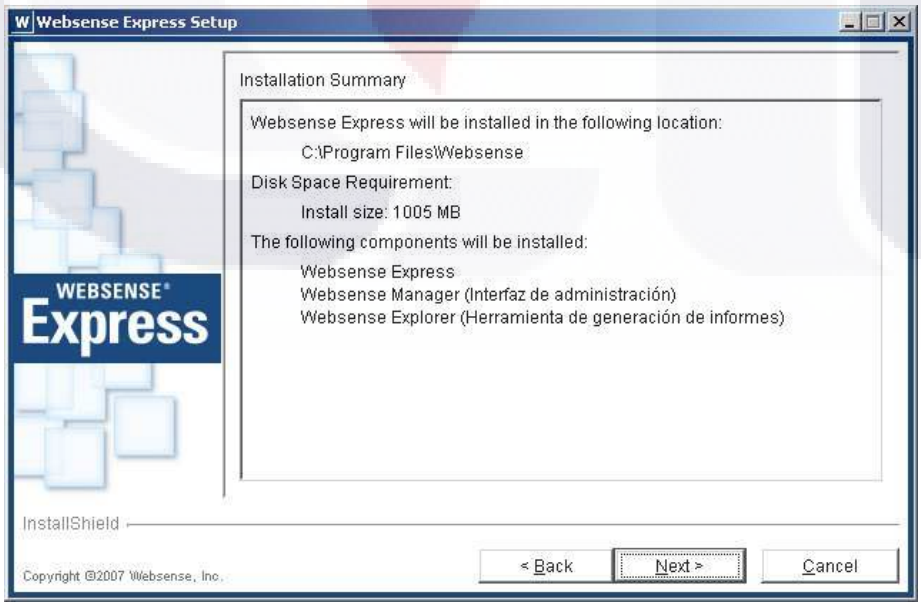
Se confirmó nuestro deseo de no bloquear por usuario, por lo que se presionó el botón Yes:



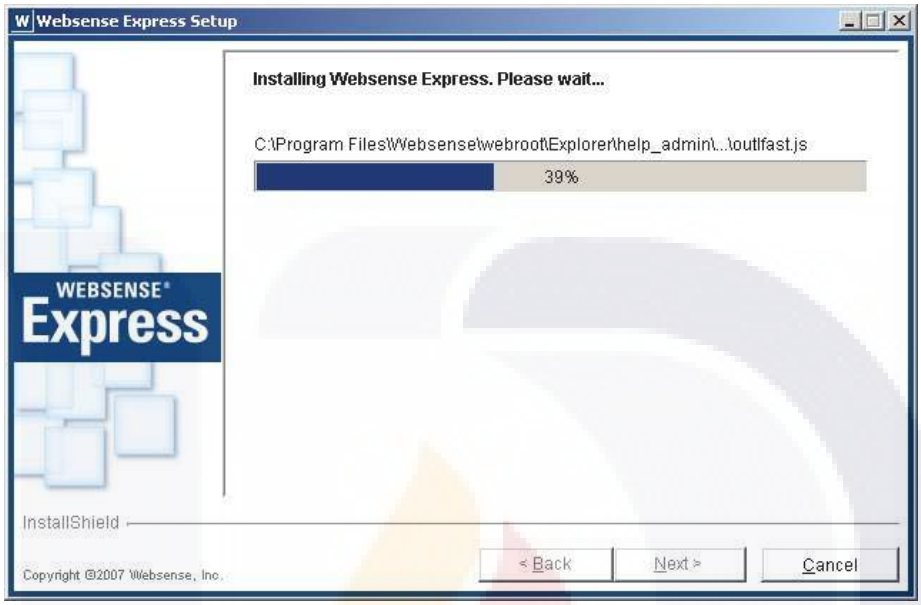
Se aceptó el directorio de instalación que trae por defecto y se presiono el botón Next:



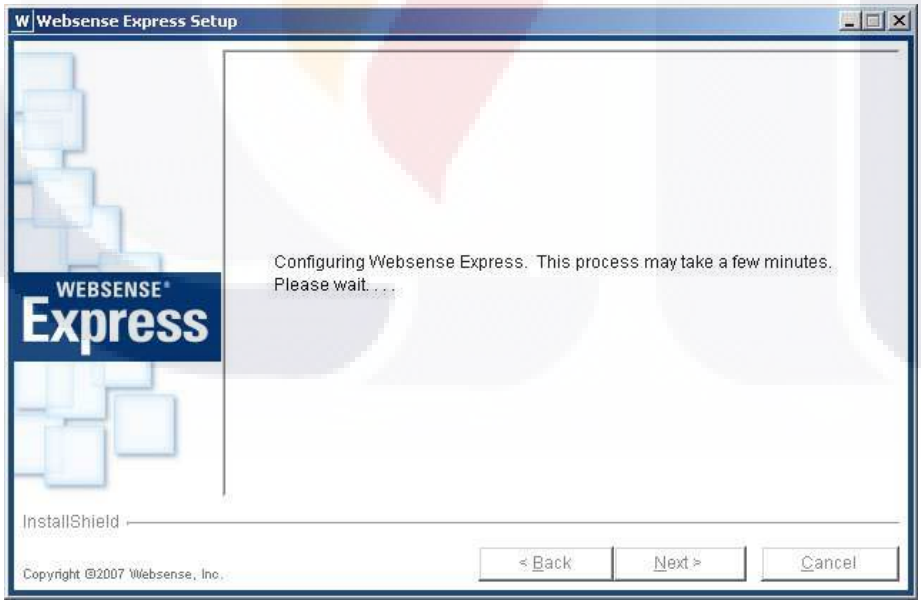
Se nos presentó un resumen de la instalación, para continuar se presionó el botón Next:



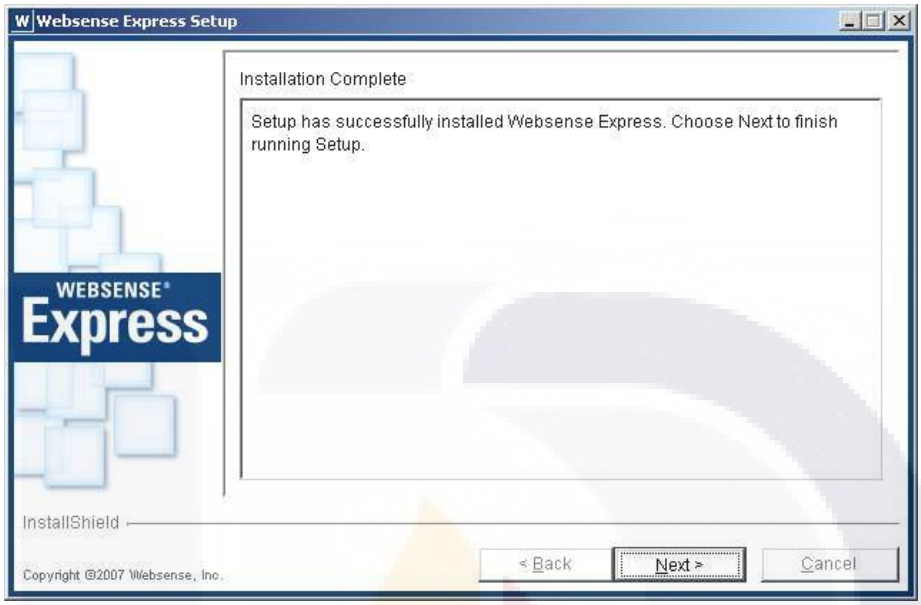
Se inició la instalación a disco duro de Websense Express:



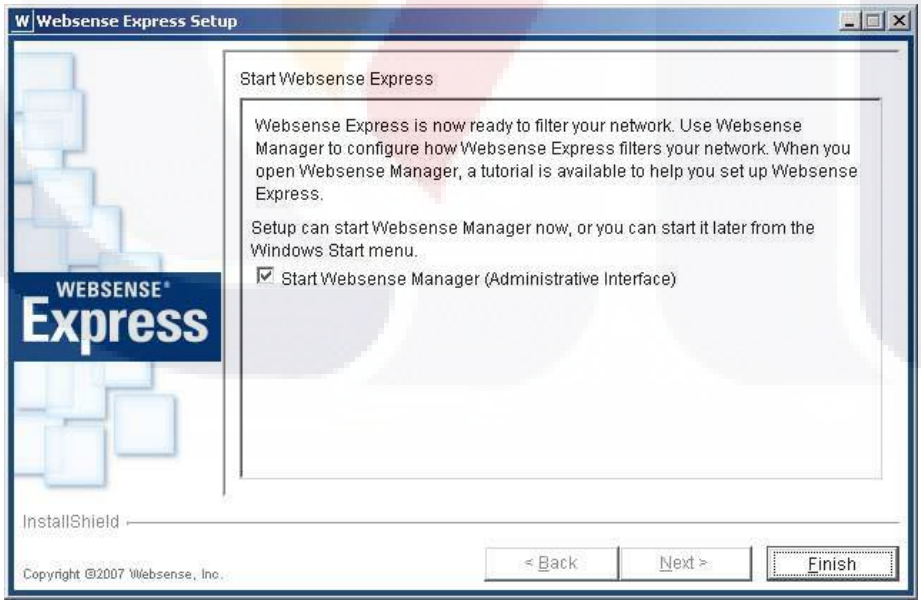
Se inició la configuración de Websense Express por parte del instalador:



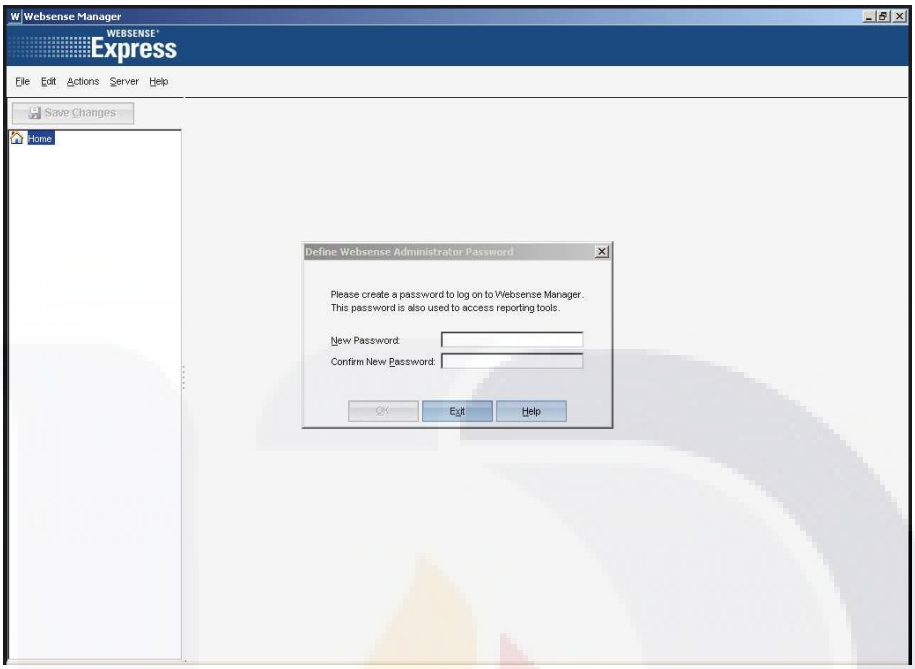
La instalación y configuración de Websense Express terminó correctamente:



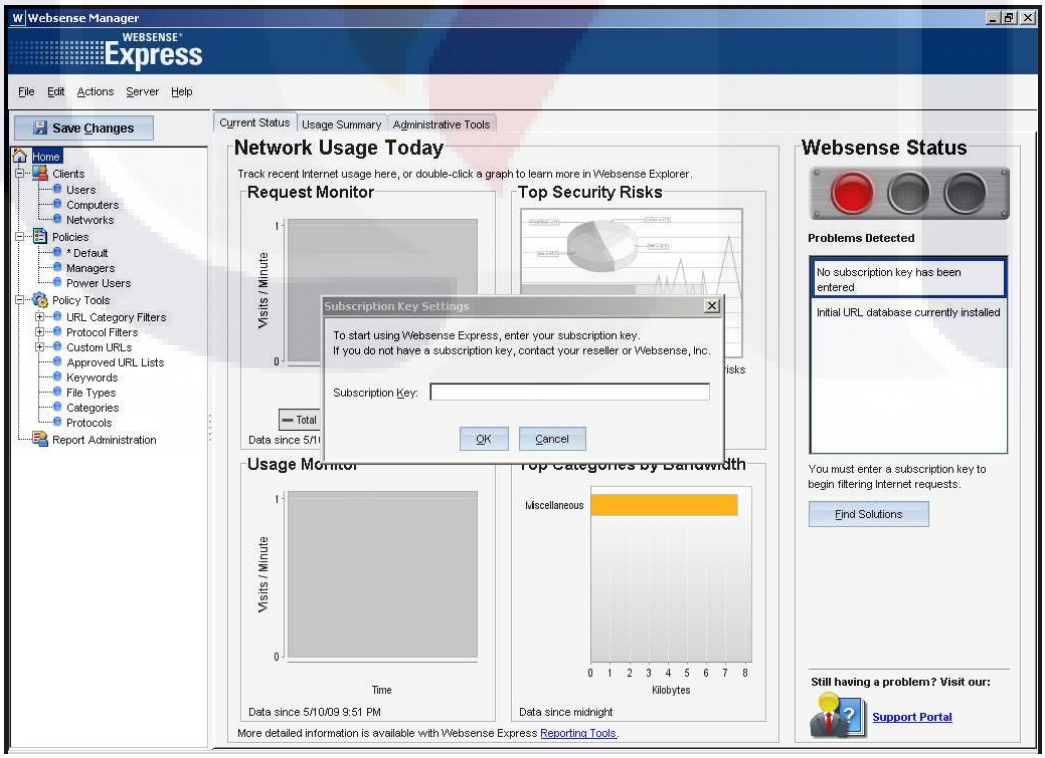
Se presionó el botón Finish para iniciar la interfaz gráfica de administración:



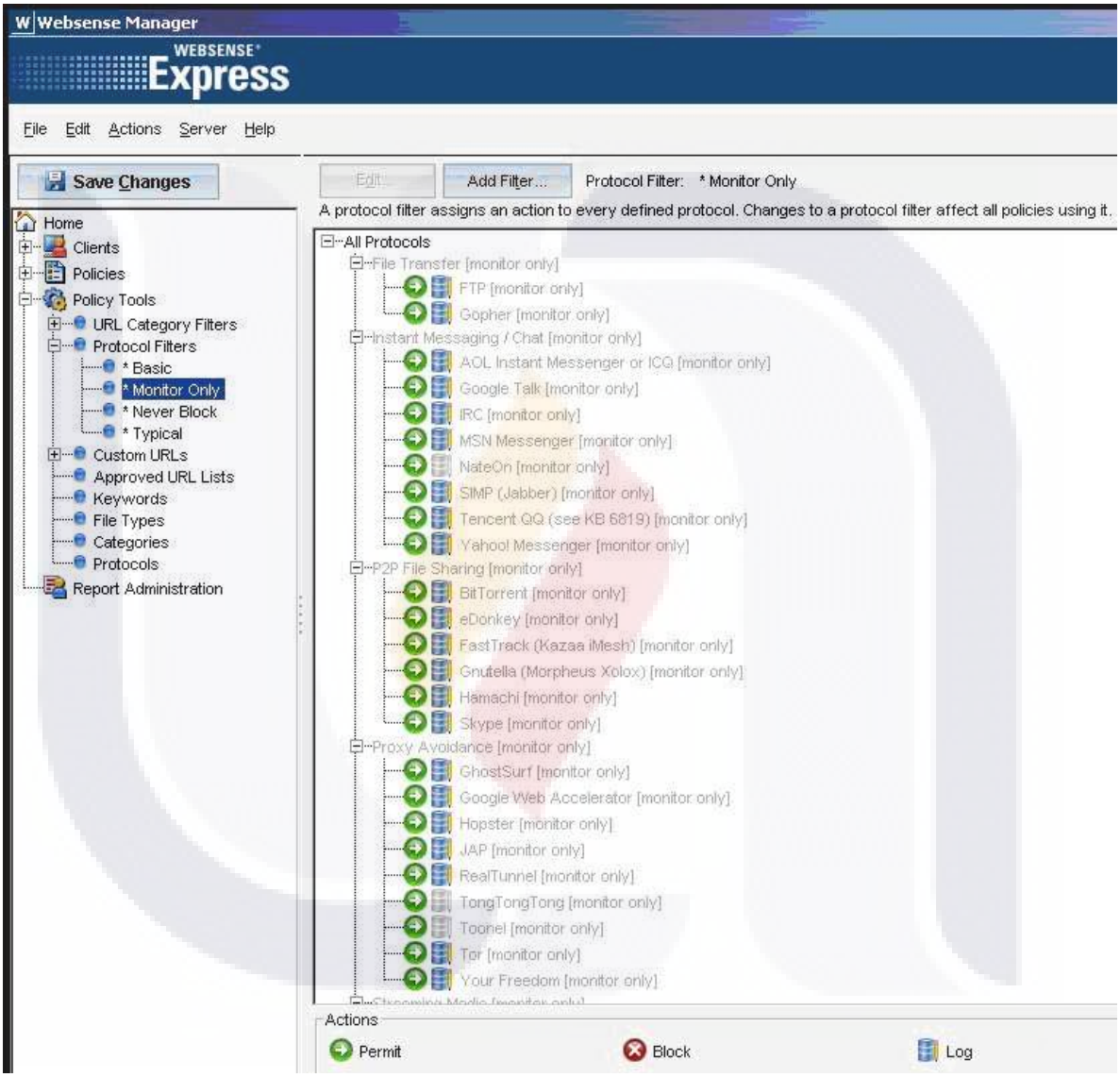
Al iniciar la interfaz de administración se solicitó la contraseña:



Ya dentro del sistema de administración se introdujo la clave de suscripción proporcionada:



3. Configuración de Websense Express: Se configuró para que solo se monitoreara los protocolos que puede bloquear Websense, para lo anterior se seleccionó *Monitor Only:*



Instalación y Configuración del equipo B

Cortafuegos en GNU/Linux –: Ubuntu Server, Netfilter Iptables, l7filter, brigde y sysstat.

En esta sección se muestra como fue instalado y configurado el equipo B.

– Cortafuegos GNU/Linux –. El Sistema Operativo instalado fue Ubuntu Hardy Heron (8.04 LTS).

Además se utilizó el siguiente software:

- Netfilter
- l7filter (Modulo de Netfilter).
- Brigde-utils
- Sysstat

Requerimientos

Para la instalación fue necesario lo siguiente:

1. El CD de instalación de Ubuntu Server 8.04.2 LTS, disponible en el siguiente enlace: ftp://releases.ubuntu.com/releases/hardy/ubuntu-8.04.2-server-i386.iso
2. Una conexión de banda ancha hacia Internet.

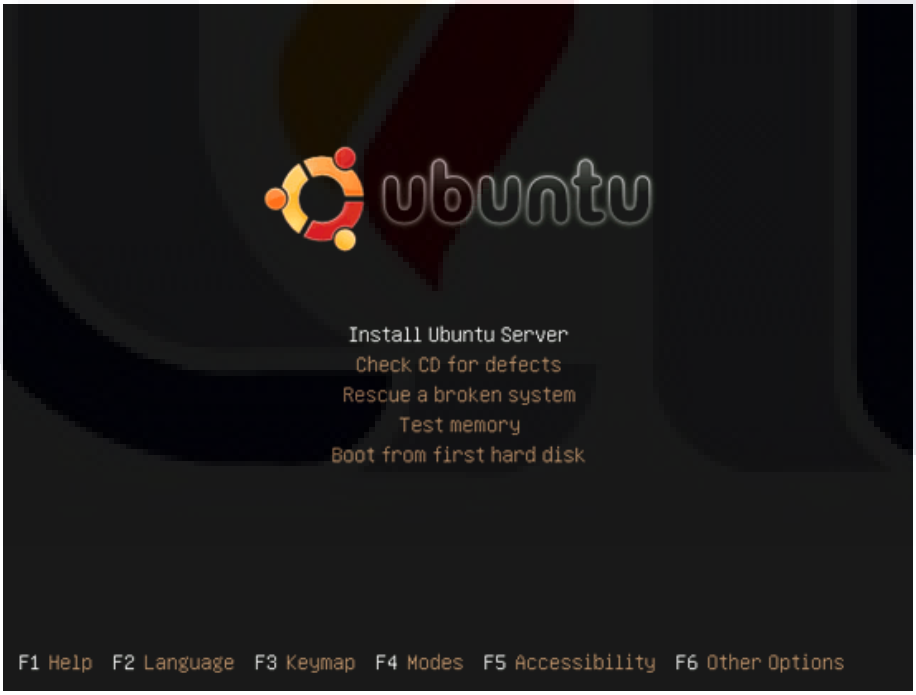
Nota Preliminar:

En este procedimiento fue utilizado como nombre de anfitrión firewall.uaa.mx

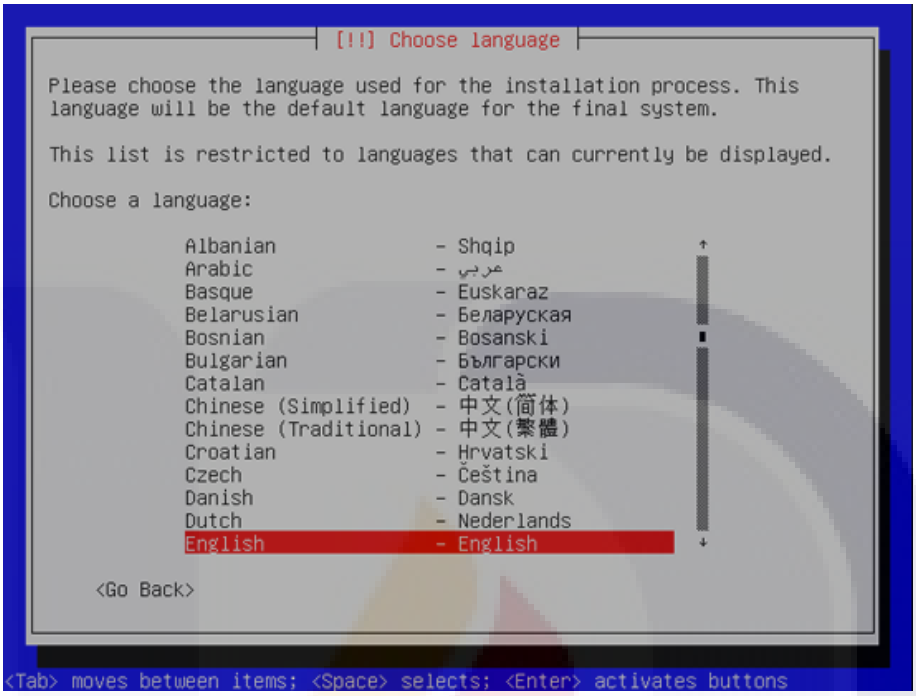
1. Instalación de Ubuntu Server 8.04.2.

1. Se insertó el CD de instalación de Ubuntu Server 8.04.2 y se arrancó desde el. Debido a la gran cantidad de información y documentación de Ubuntu en Inglés se eligió ese idioma:

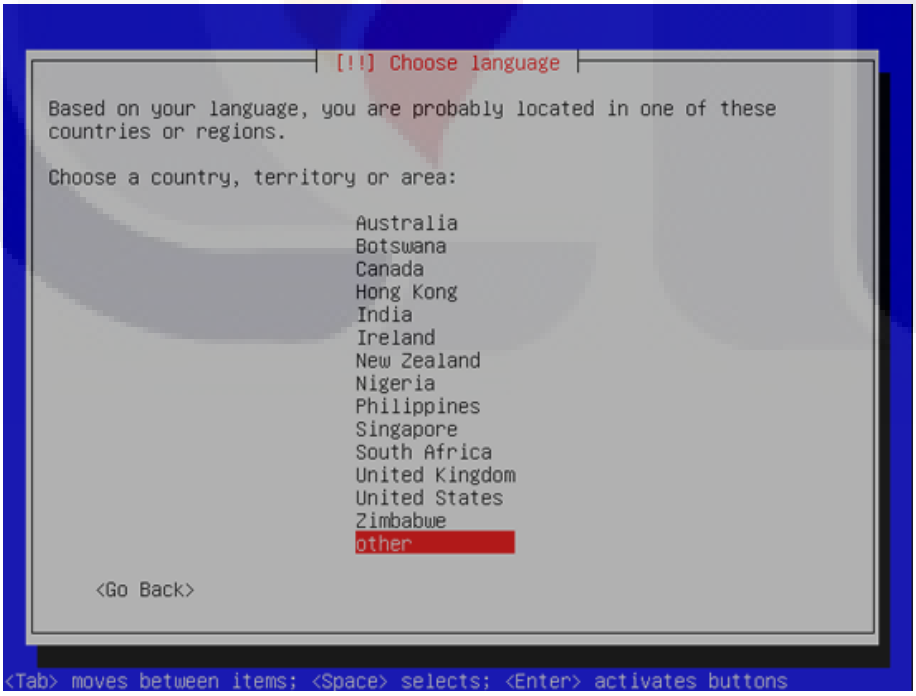
2. Se seleccionó la instalación a disco duro “Install Ubuntu Server”:



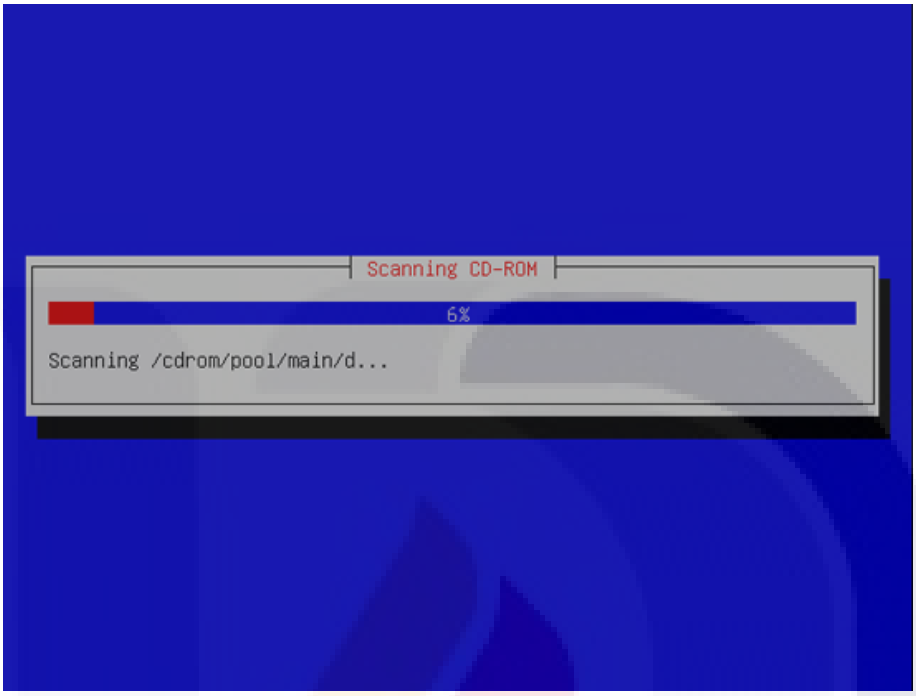
3. Al iniciar la instalación se eligió nuevamente el idioma Inglés:

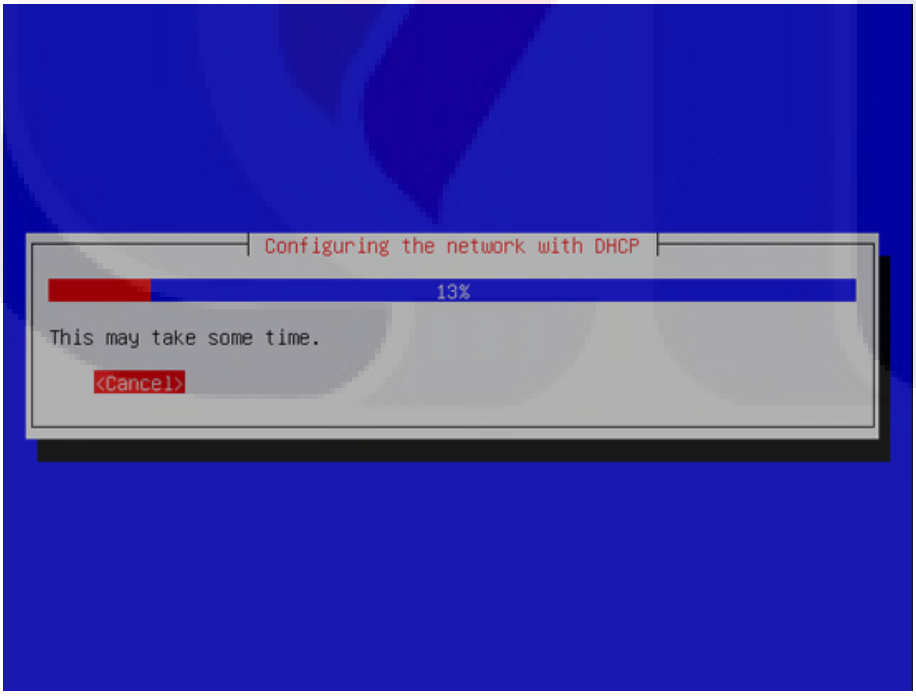


4. Se eligió nuestra ubicación:

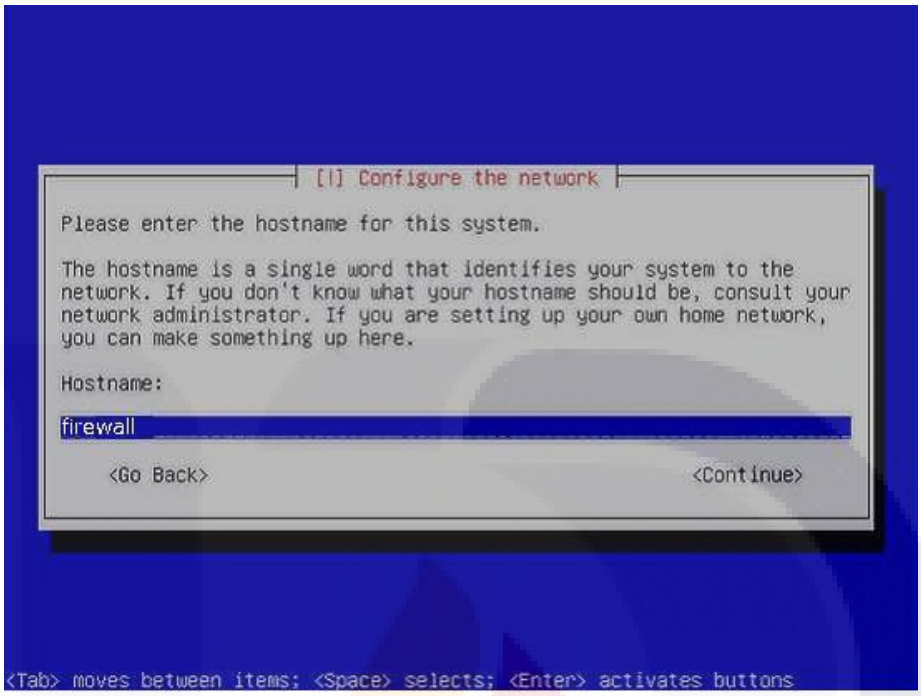


5. El instalador verificó el CD de instalación además del hardware, finalmente se configuró la red utilizando el servicio de DHCP:

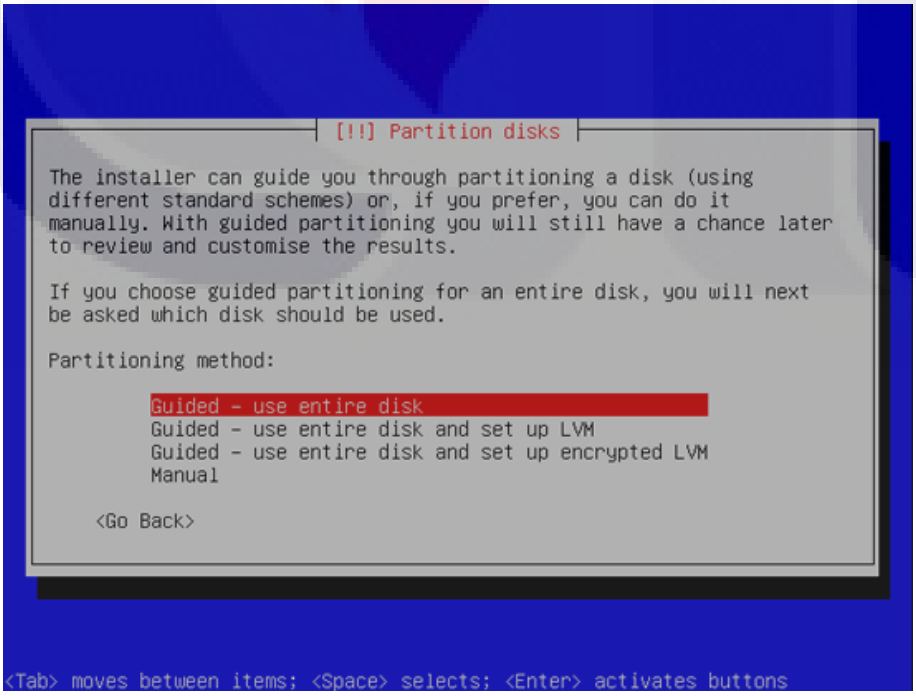




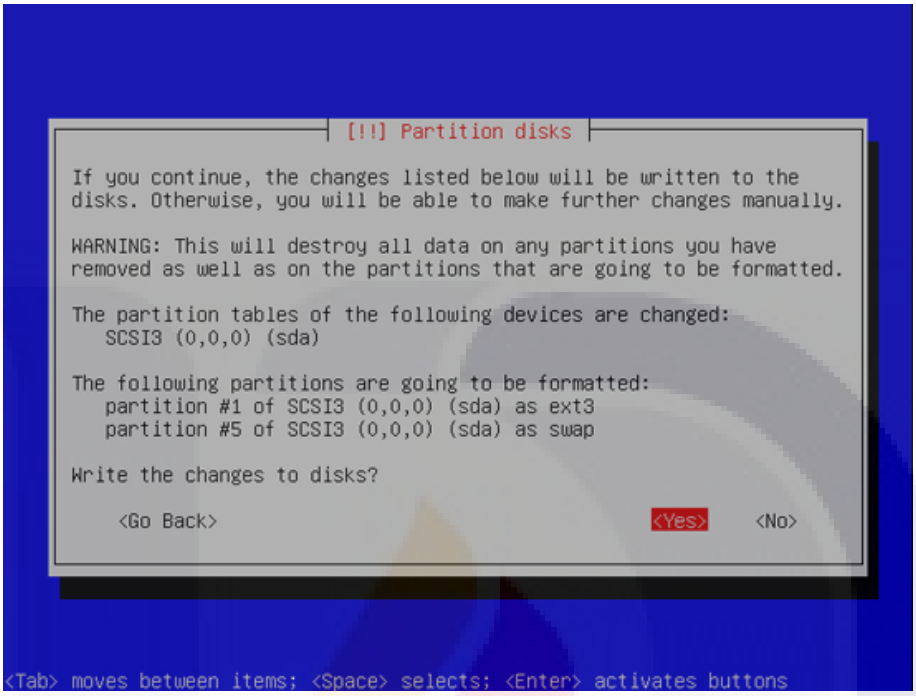
Se escribió el nombre del anfitrión, el sistema fue llamado firewall.uaa.mx:



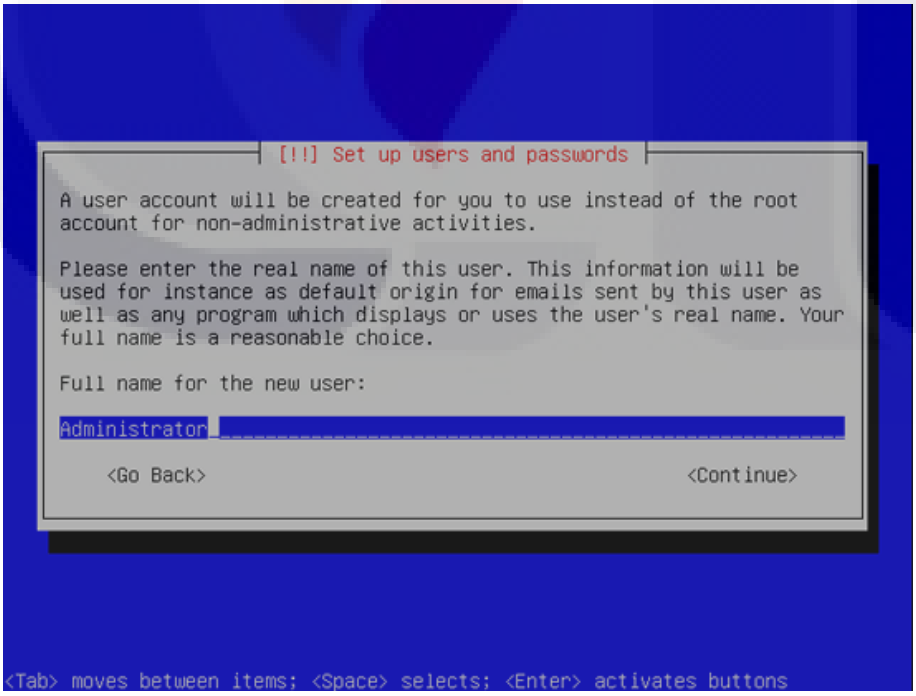
6. Por simplicidad solo se creo una gran partición (con el punto de montaje /) y una pequeña partición para área de intercambio (swap) para ello se eligió “Guided - use entire disk”:

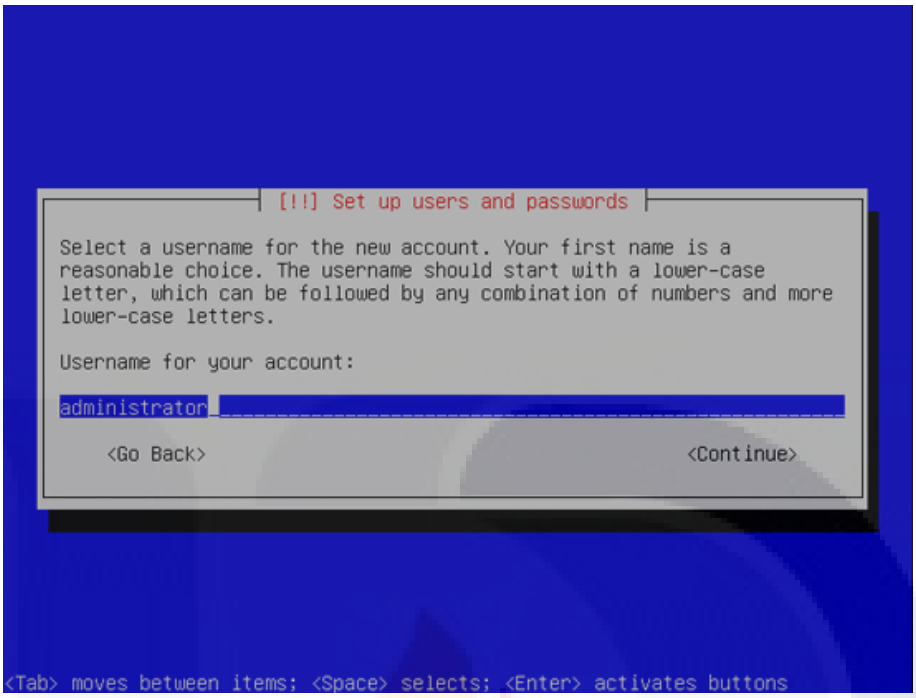


Al terminar el particionado se nos preguntó si se deseaba escribir los cambios en el disco, Al seleccionar <Yes> nuestras particiones fueron físicamente creadas:

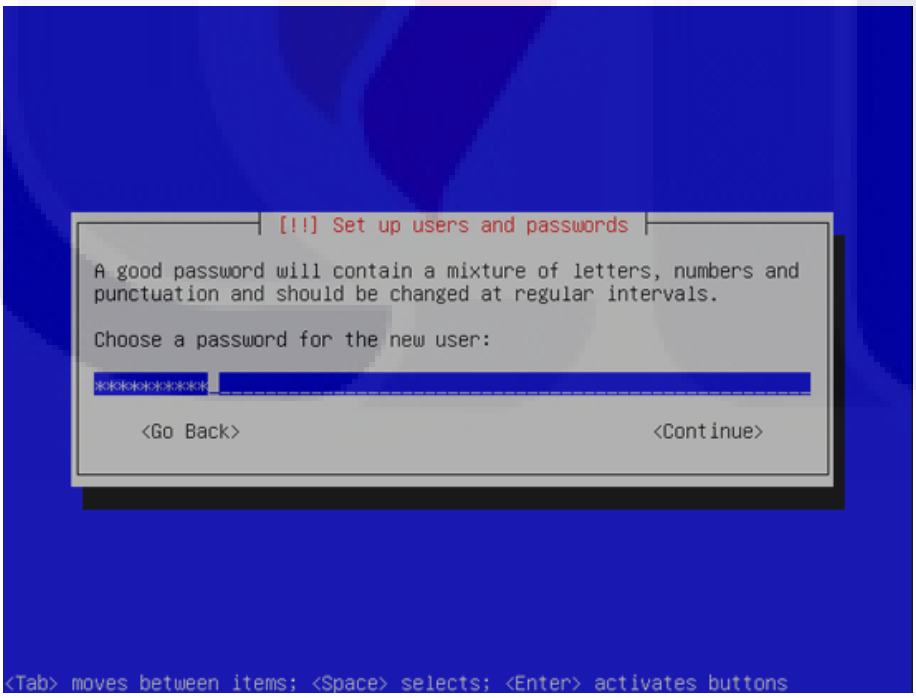


7. Se creo el usuario administrator con el nombre de usuario Administrator:

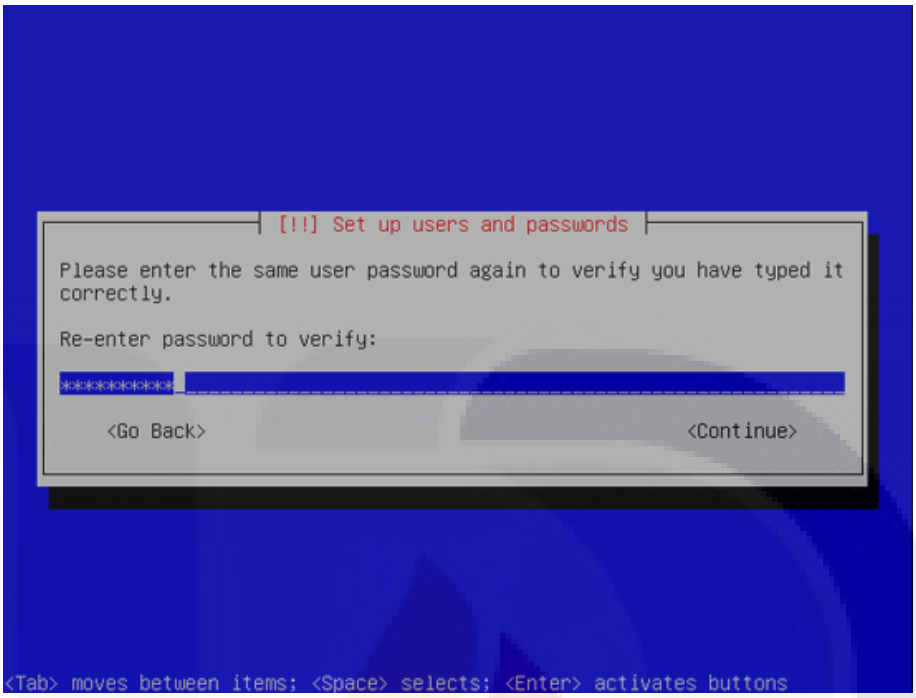




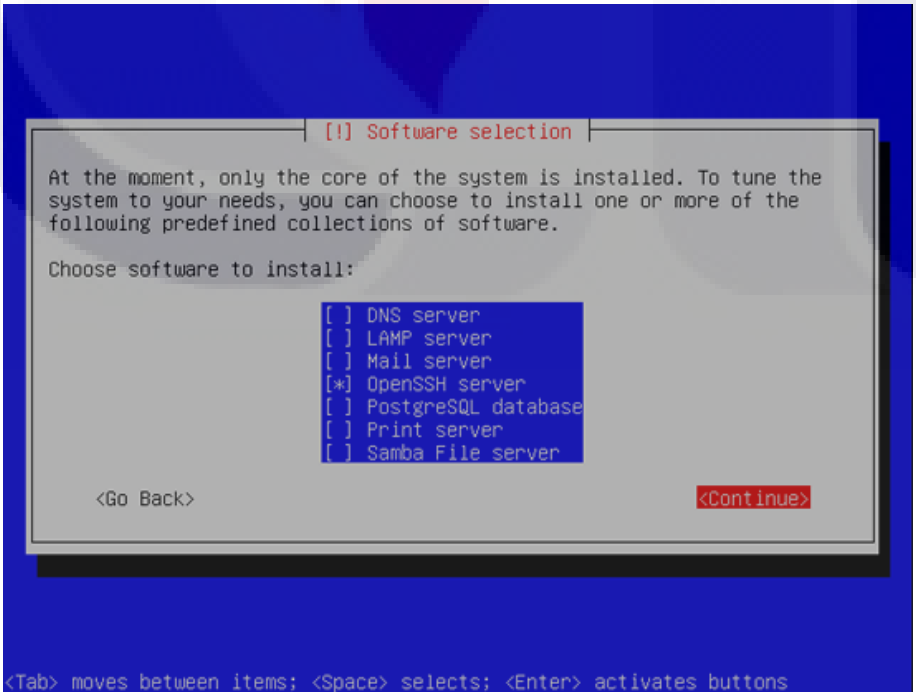
En esta pantalla se escribió la contraseña:



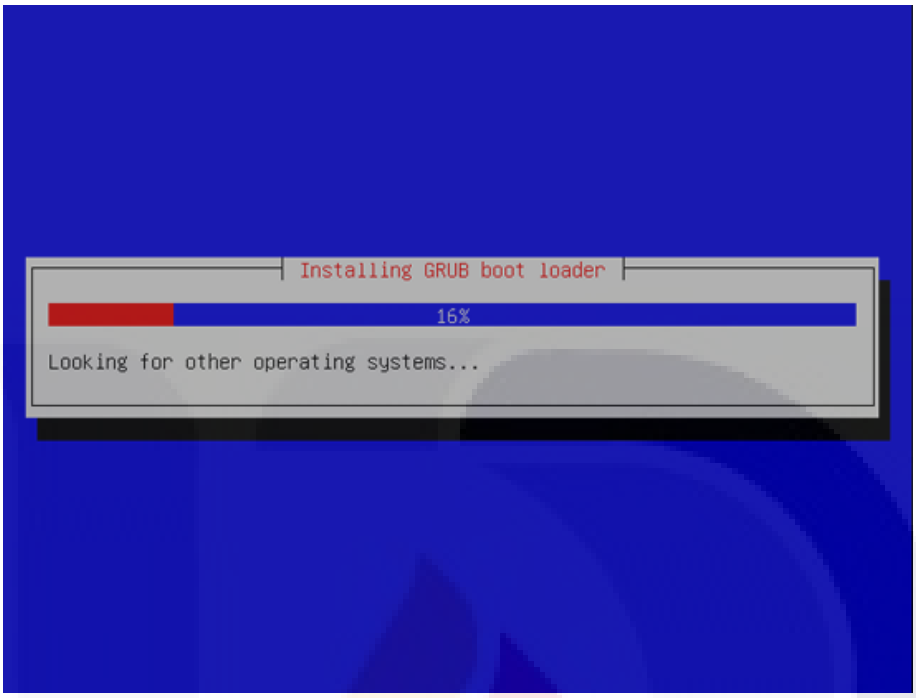
Se reescribió la contraseña elegida:



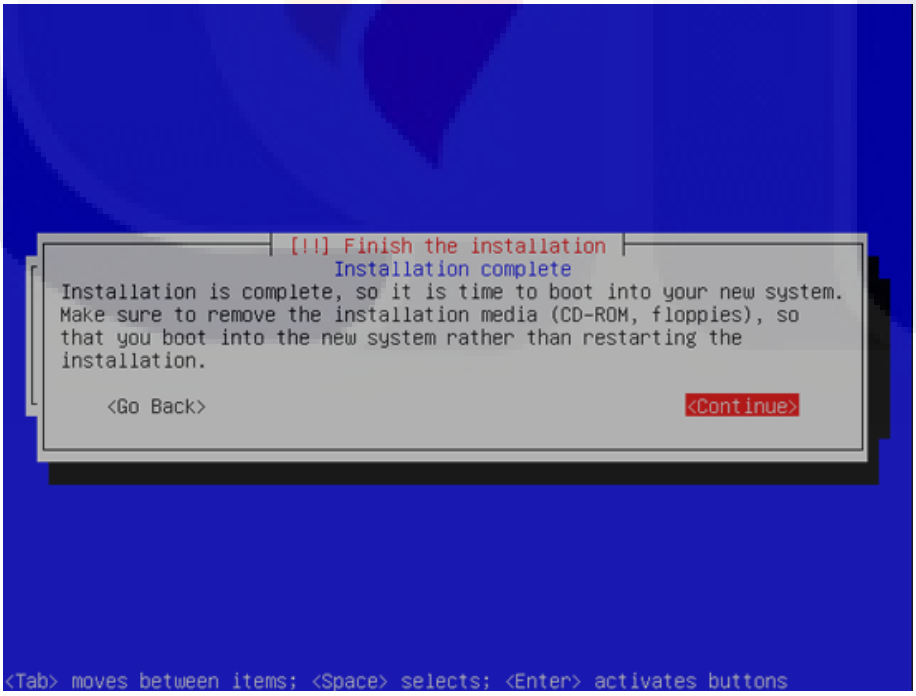
8. El único servicio seleccionado para su instalación fue OpenSSH, este nos permitió conectarnos inmediatamente al sistema utilizando un cliente SSH:



9. Se instaló el arrancador GRUB:



10. La instalación del sistema base finalizó, por lo que fue retirado el CD de instalación y se reinició el sistema:



11. Activación de la cuenta root: Después del reinicio solo es posible acceder con la cuenta administrator previamente creada. Debido a que los siguientes procedimientos de configuración deben de ejecutarse con la cuenta de root, fue necesario activar dicha cuenta.

Se activó la cuenta de root asignándole una contraseña:

```
root@firewall:~#  sudo passwd root
```

Se inició sesión con root ejecutando:

```
root@firewall:~#  su
```

2. Configuración de la red.

A este equipo se le instalaron 3 interfaces de red, reconocidas por el sistema como eth2, eth3 y eth4. eth2 fue configurada con un IP de la red utilizada en el laboratorio, eth3 y eth4 fueron configuradas en puente.

Para configurar la interfaz eth2 ejecutamos el comando ifconfig:

```
root@firewall:~# dhclient3 eth2
```

Se verificó que la configuración anterior se activo correctamente ejecutando el comando ifconfig:

```
root@firewall:~# ifconfig
eth2      Link encap:Ethernet  HWaddr 00:14:d1:13:f2:a5
          inet addr:172.16.1.11  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::214:d1ff:fe13:f2a5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26247 (25.6 KB)  TX bytes:19562 (19.1 KB)
          Interrupt:10 Base address:0xec00
```

3. Actualización del sistema.

1. Se editó el archivo /etc/apt/sources.list y se actualió la instalación de Ubuntu. Para ello fue necesario;

- Comentar o remover la línea de la instalación desde CD
- De comentar las líneas para los repositorios universe y multiverse.

El archivo /etc/apt/sources.list quedó de la siguiente manera:

```
deb http://de.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy main restricted

## Major bug fix updates produced after the final release of the distribution.
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu team.
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse

## Uncomment the following two lines to add software from the 'backports' repository.
## N.B. software from this repository may not have been tested as extensively as that contained in the main release, although ### it includes newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository. This software is not part of Ubuntu, but is offered by Canonical
### service to Ubuntu users.
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

2. Se actualizó la base de datos de paquetes con el comando apt-get:

```
root@firewall:~#  apt-get update
```

3. Se instalaron las últimas actualizaciones de los paquetes ya instalados:

```
root@firewall:~#  apt-get upgrade
```

4. Instalación de Netfilter Iptables.

Ubuntu Server 8.4.2 por defecto trae instalado y compilado el modulo de Iptables, por lo que solo fue necesario verificar.

1. Se verificó que iptables estuviera instalado y funcionando

```
root@firewall:~# iptables
iptables v1.3.8: no command specified
Try `iptables -h' or 'iptables --help' for more information.
```

5. Instalación y configuración de L7filter.

http://kuscsik.blogspot.com/2008/02/how-to-userspace-l7-filter-on-ubuntu.html

Para instalar el modulo de L7filter de una manera sencilla se siguió el manual de Zoltan

1. Se agregó al final del archivo /etc/apt/sources.list los repositorios de Zoltan:

```
deb http://ppa.launchpad.net/kuscsik/ubuntu hardy main
deb-src http://ppa.launchpad.net/kuscsik/ubuntu hardy main
```

2. Se actualizaron los paquetes disponibles con el comando apt-get:

```
root@router:~#  apt-get update
```

3. Se instalaron l7-filter-userspace y l7-protocols:

```
root@router:~#  apt-get install l7-filter-userspace  l7-protocols
```

4. Configuración de l7-filter:

Se editó el archivo /etc/l7_filter.conf y se agregó el protocolo ssh, http:

```
ssh 5
http 5
```

Se utilizó iptables para redireccionar los paquetes al modulo de l7filter:

```
root@firewall:~# iptables -A FORWARD -j NFQUEUE --queue-num 0
```

La lista de los paquetes que puede identificar l7iflter se puede consultar en el siguiente enlace: http://l7-filter.sourceforge.net/protocols

6. Instalación y configuración de brigde-utils.

Esta herramienta permitió que nuestro cortafuegos se configurará en modo puente y así se logró hacer un cortafuegos transparente.

1. Se actualizaron los paquetes disponibles con el comando apt-get:

```
root@router:~#  apt-get update
```

2. Se instaló bridge-utils:

```
root@router:~#  apt-get aptitude install bridge-utils
```

Como se mencionó en el punto 2 *configuración de la red* el equipo tiene 3 interfaces de red eth2, eth3 y eth4 para ver la información de las interfaces se utilizó el comando ifconfig -a

```
root@firewall:~# ifconfig -a
eth2      Link encap:Ethernet  HWaddr 00:14:d1:13:f2:a5
          inet addr:172.16.1.11  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::214:d1ff:fe13:f2a5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:361 errors:0 dropped:0 overruns:0 frame:0
          TX packets:187 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:71138 (69.4 KB)  TX bytes:23192 (22.6 KB)
          Interrupt:10 Base address:0xec00
```

```
eth3      Link encap:Ethernet  HWaddr 00:a0:c9:f2:c2:4f
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth4      Link encap:Ethernet  HWaddr 00:0a:e6:a2:65:52
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:11 Base address:0xcd00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

3. Configuración del cortafuegos en puente:

Se asignó la dirección IP 0.0.0.0 a la interfaz eth3:

```
root@router:~#  ifconfig eth3 0.0.0.0
```

Se asignó la dirección IP 0.0.0.0 a la interfaz eth4:

```
root@router:~#  ifconfig eth4 0.0.0.0
```

Se creó el puente con el nombre tesis:

```
root@router:~#  brctl addbr tesis
```

Se agregó la interfaz eth3 al puente:

```
root@firewall:~# brctl addif tesis eth3
```

Se agregó la interfaz eth4 al puente:

```
root@firewall:~# brctl addif tesis eth4
```

Se activó el puente:

```
root@firewall:~# ifconfig tesis up
```

Se verificó que el puente fue creado correctamente con el comando *brctl show*:

```
root@firewall:~# brctl show
bridge name     bridge id               STP enabled     interfaces
tesis           8000.000ae6a26552       no              eth3
                                                        eth4
```

7. Automatización de los puntos 5 y 6 anteriores.

Se creó el script ***puente*** para automáticamente hacer lo descrito en el punto 5 *Instalación y configuración de L7filter* así como lo descrito en el punto 6 *Instalación y configuración de brigde-utils*. El contenido del script es el siguiente:

```
ifconfig eth3 0.0.0.0
ifconfig eth4 0.0.0.0
brctl addbr tesis
brctl addif tesis eth3
brctl addif tesis eth4
ifconfig tesis up
iptables -A FORWARD -j NFQUEUE --queue-num 0
sleep 2
l7-filter -f /etc/l7  filter.conf
```

8. Instalación de sysstat.

Esta herramienta contiene comandos para monitorizar el sistema operativo GNU/Linux, los comandos que contiene son: sar, sadf, iostat, mpstat y pidstat. El comando que utilizaremos es ***mpstat*** este nos permite obtener información del uso de CPU.

1. Se instaló sysstat:

```
root@firewall:~# apt-get install sysstat
```

2. Se probó que **mpstat** funcionará correctamente:

```
root@firewall:/opt# mpstat
Linux 2.6.24-23-server (firewall)       05/02/2009

09:36:04 PM  CPU   %user   %nice    %sys %iowait    %irq   %soft  %steal   %idle    intr/s
09:36:04 PM    all      0.08      0.00      0.08      0.08      0.00      0.00      0.00   99.75     43.06
```

Instalación y Configuración del equipo B

Cortafuegos en Sistema Integrado –: Packetshaper 7500.

En esta sección se muestra como fue instalado y configurado el equipo B.

– Cortafuegos en Sistema Integrado –

Requerimientos

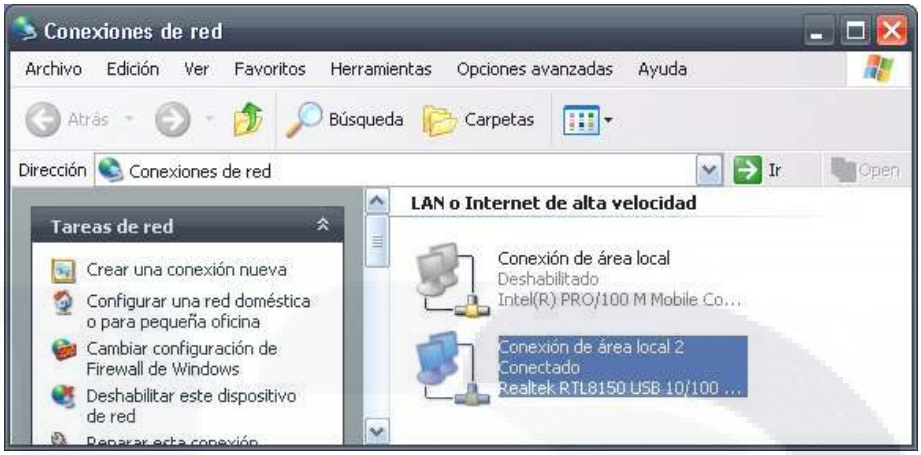
Para instalación fue necesario lo siguiente:

- Sistema Integrado PacketShaper
  - Version: PacketShaper v8.3.3g1 2008-11-07
  - Product: PacketShaper 7500
  - Part Number: 123-0001-01 REV C
  - Serial Number: 175-10010384
- Equipo PC configurado en la red 207.78.98.X para llevar a cabo la configuración del sistema integrado PacketShaper 7500.

1. Configuración del equipo PC para configuración.

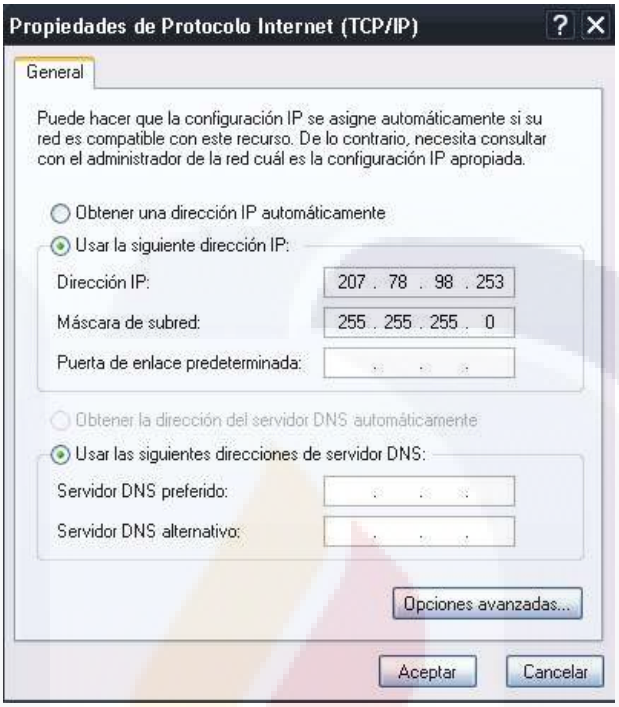
La consola de administración de PacketShaper 7500 por defecto esta configurada en el IP 207.78.98.254 por lo que fue necesario que el equipo PC utilizado estuviera en la misma red para poder acceder a la consola de administración, el equipo PC que se utilizó tenia el sistema operativo Windows XP, por lo que el procedimiento de configuración de la red se describe a continuación:

1. Se configuró la *conexión de área local 2*, para lo cual dentro del menú de *Conexiones de red* se modificaron las propiedades de dicha conexión:



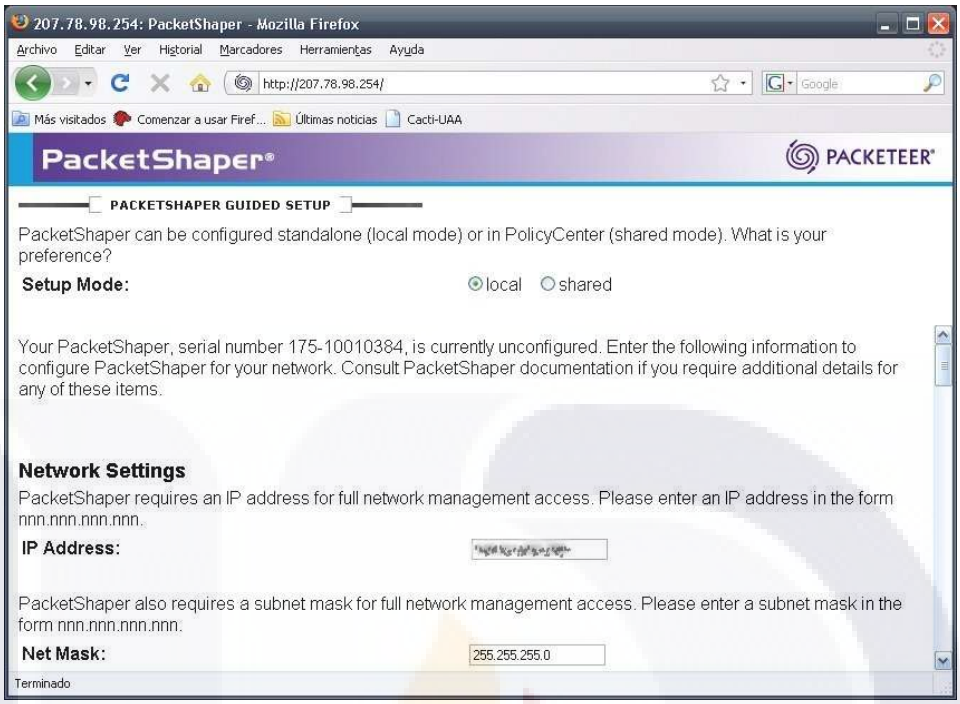
2. Dentro de las propiedades de dicha conexión se modificó el *Protocolo Internet (TCP/IP)* para poder modificar dicho protocolo se seleccionó este y se dio clic en Propiedades, la imagen siguiente muestra esta parte:

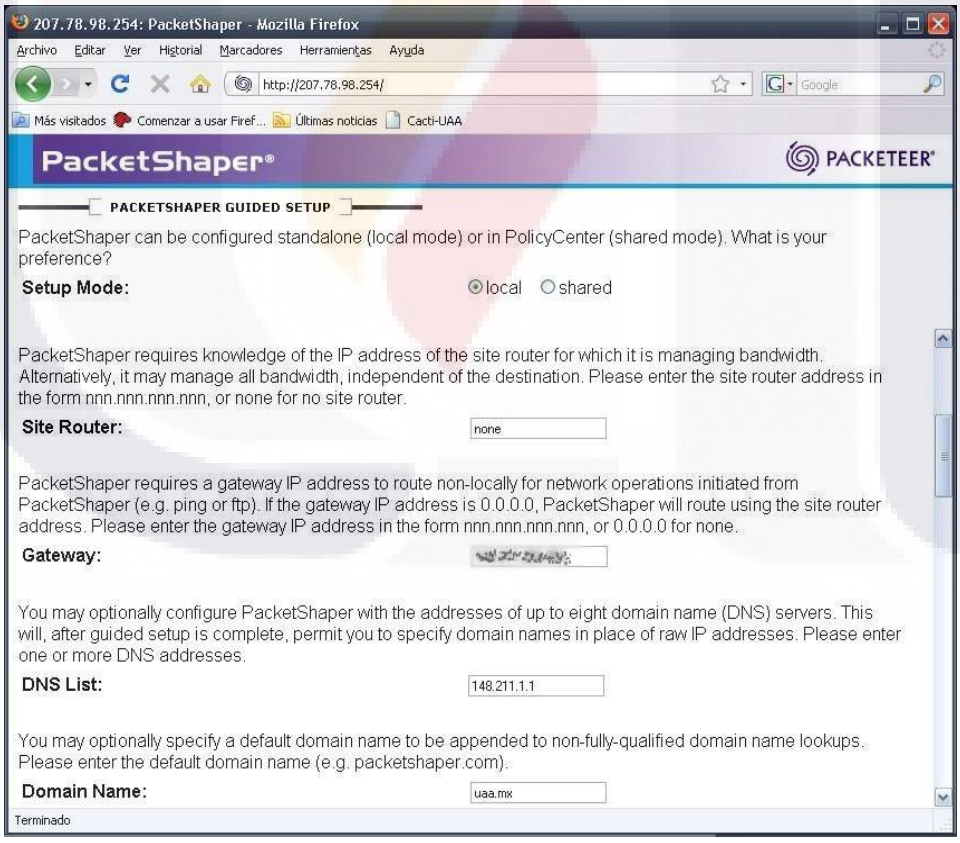
3. Dentro de las propiedades de *Protocolo Internet (TCP/IP)* en la parte Usar la siguiente dirección IP se configuró para que tuviera el IP 207.78.98.253 tal como se muestra a continuación:

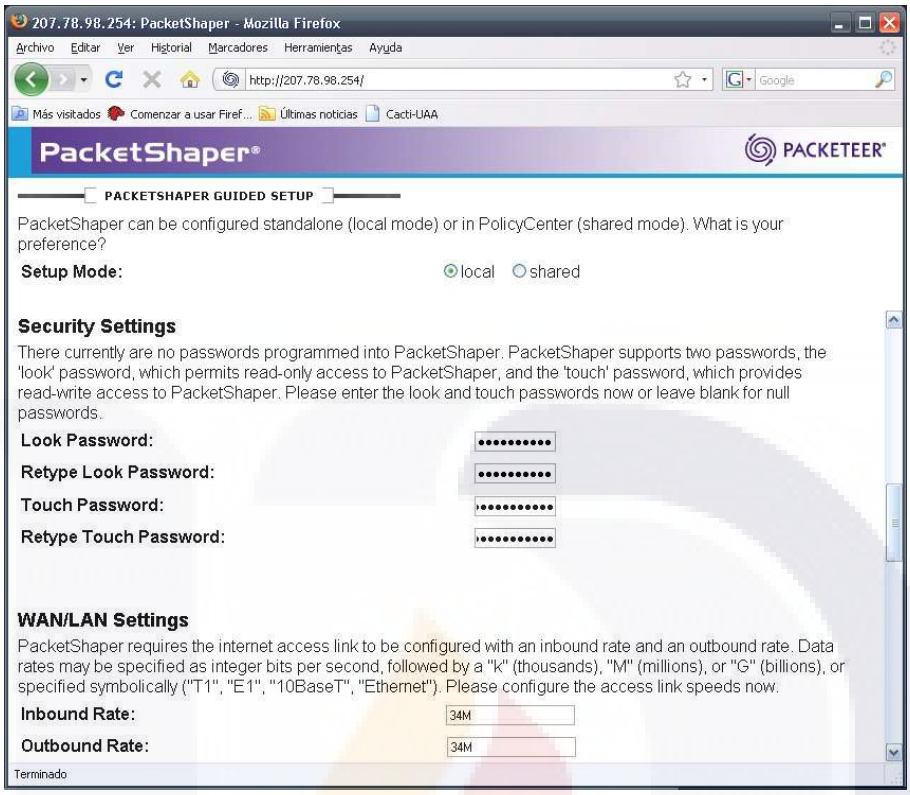


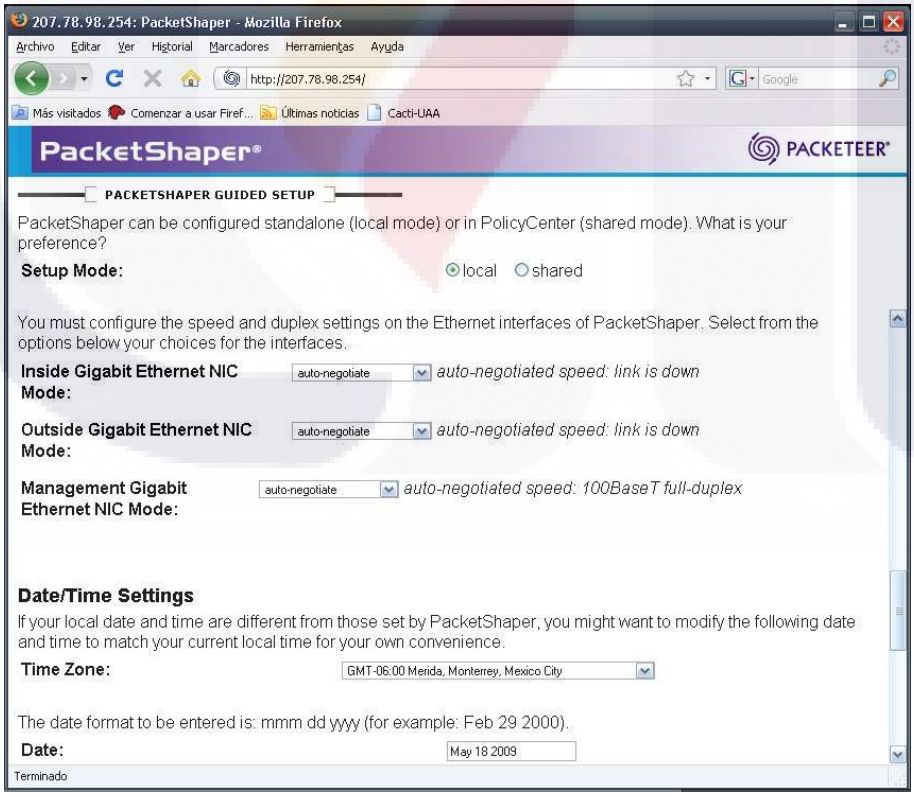
2. Configuración del sistema integrado.

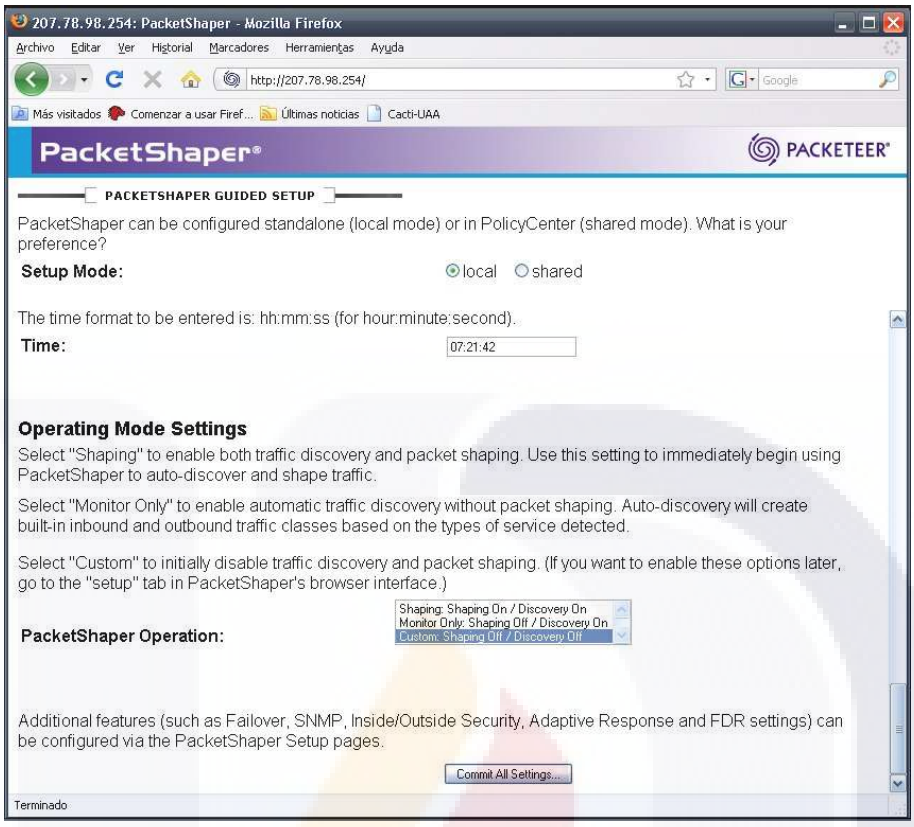
1. Ya con el IP 207.78.98.253 configurado se abrió la página de configuración de PacketShaper http://207.78.98.254 la configuración realizada se aprecia en las siguientes imágenes:





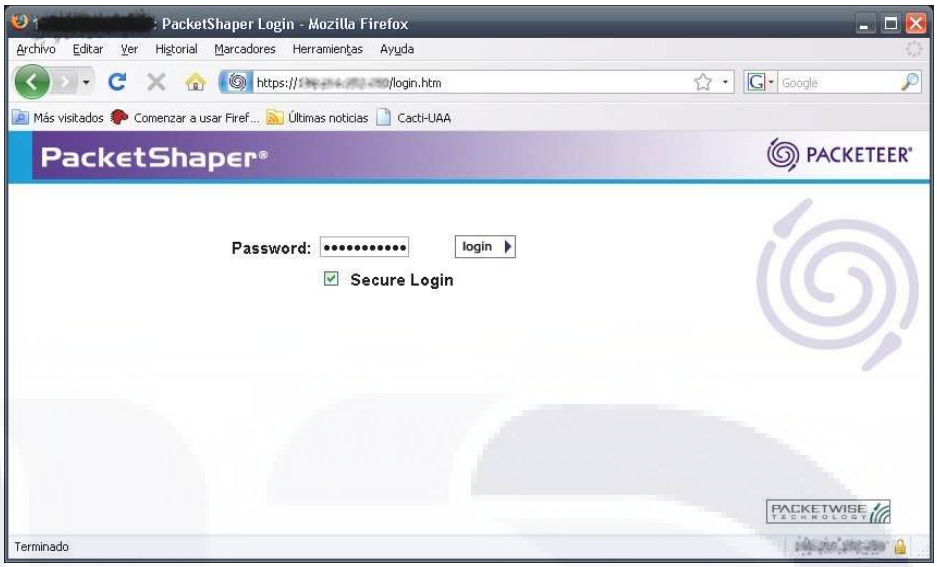


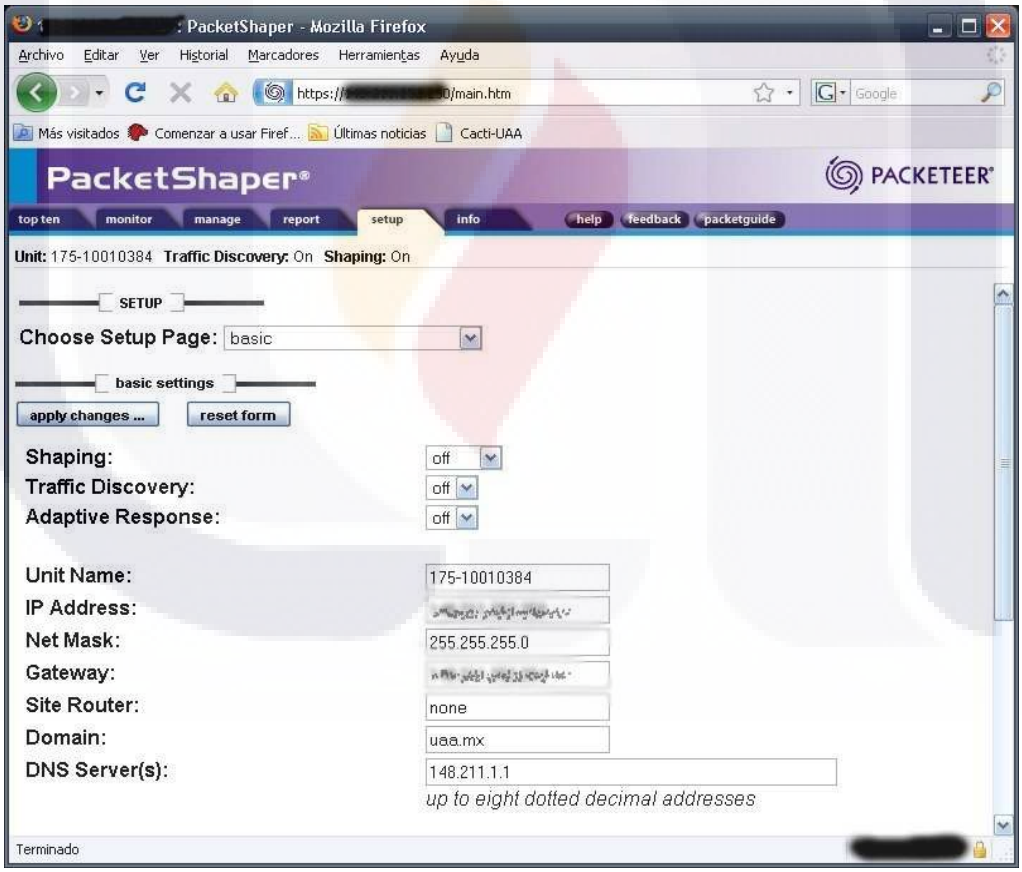




2. Una vez terminada la configuración y haber presionado el botón *Commit All Settings* se confirmó la configuración en la siguiente ventana:

3. Ya configurado PacketShaper se inició sesión para desactivar el *shaping y traficc discovery,* ver las siguientes imágenes:





PacketShaper quedó listo para iniciar el tercer escenario.

Instalación y Configuración del equipo A

Trasmisor D-ITG –: Windows XP, D-ITG y itggui-0911.

En esta sección se muestra como fue instalado y configurado el equipo C. – Trasmisor D-ITG –. El Sistema Operativo instalado fue Windows XP Professional.

Además se utilizó el siguiente software:

- D-ITG modulo ITGSend
- Itggui-0911

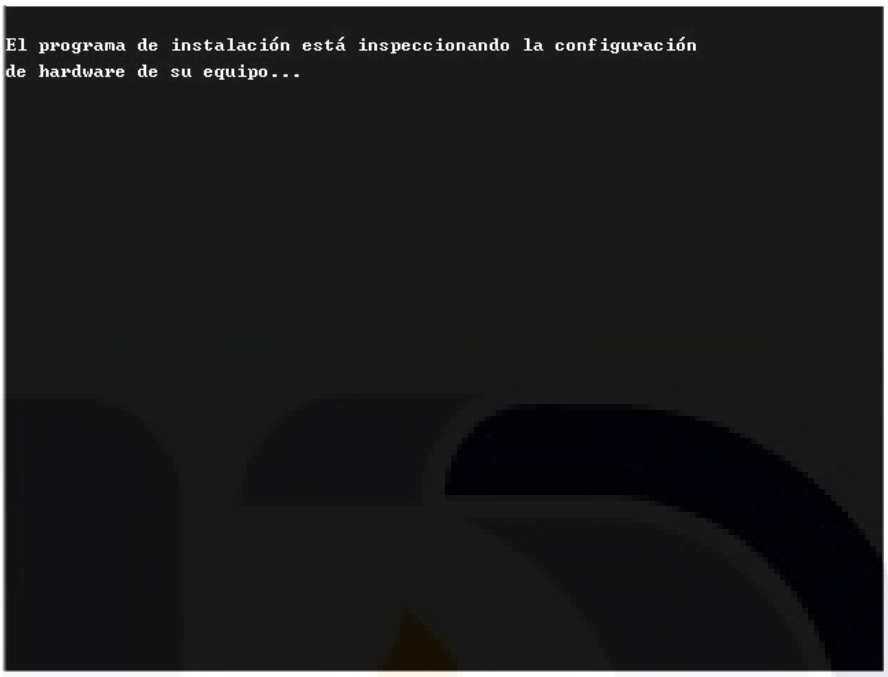
Requerimientos

Para la instalación fue necesario lo siguiente:

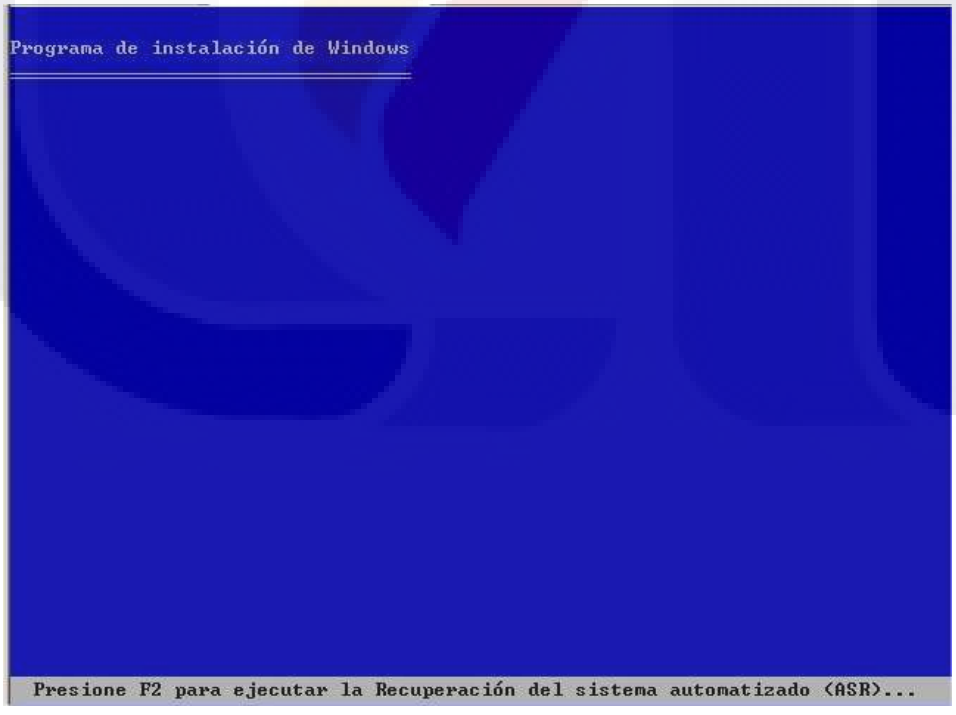
3. El CD de instalación de Windows XP Professional.
4. D-ITG versión estable para Windows, disponible aquí: http://www.grid.unina.it/software/ITG/codice/D-ITG-2.6.1d-WINbinaryIPv4.zip
5. Itggui-0911 Interfaz gráfica para D-ITG, disponible aquí: http://www.semken.com/downloads/itggui-0911.zip

1. Instalación de Windows XP Professional.

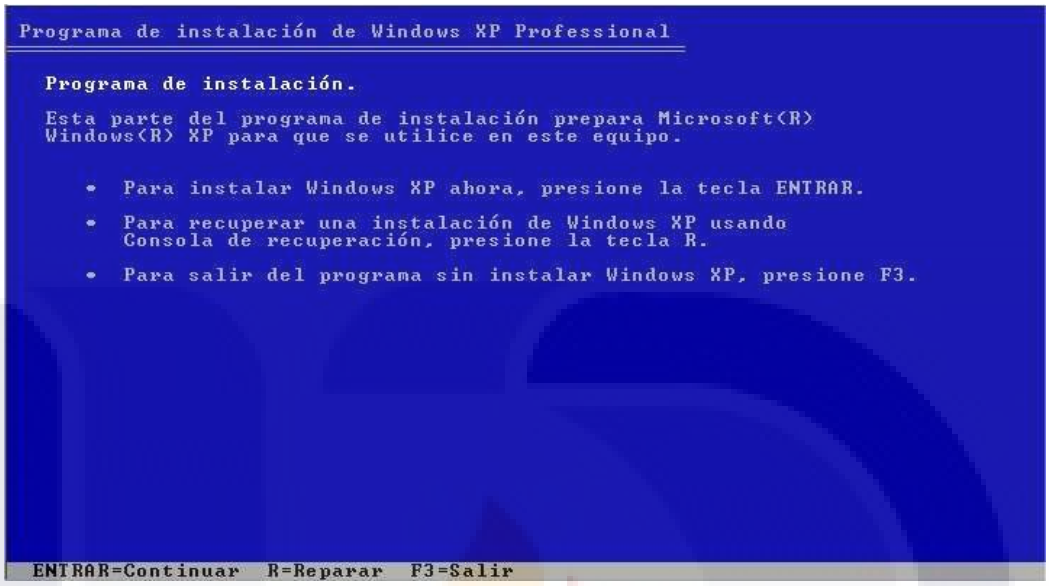
1. Se insertó el CD de instalación de Windows XP Professional, se arrancó desde el CD y aparecer el mensaje" presione cualquier tecla para iniciar desde CD", se presionó la tecla Enter para iniciar la instalación:



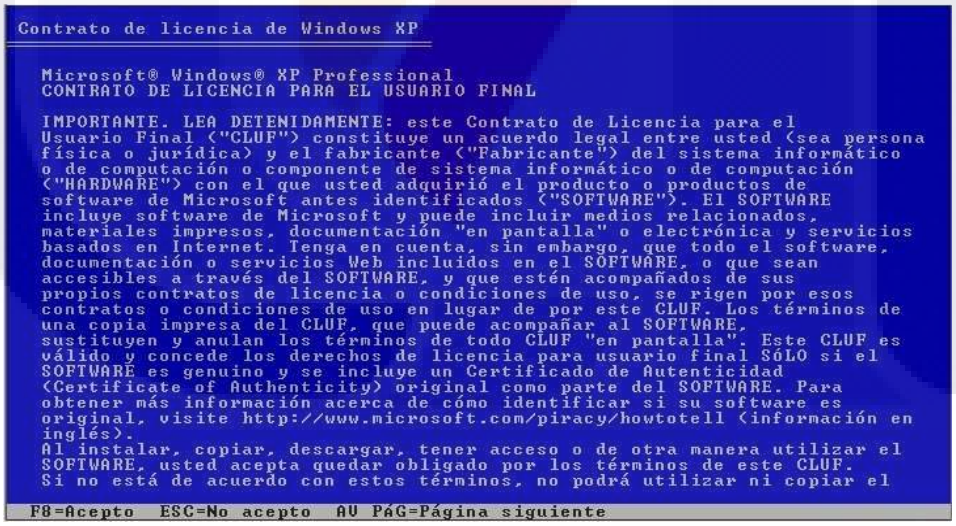
2. Se inició el "Programa de instalación de Windows":



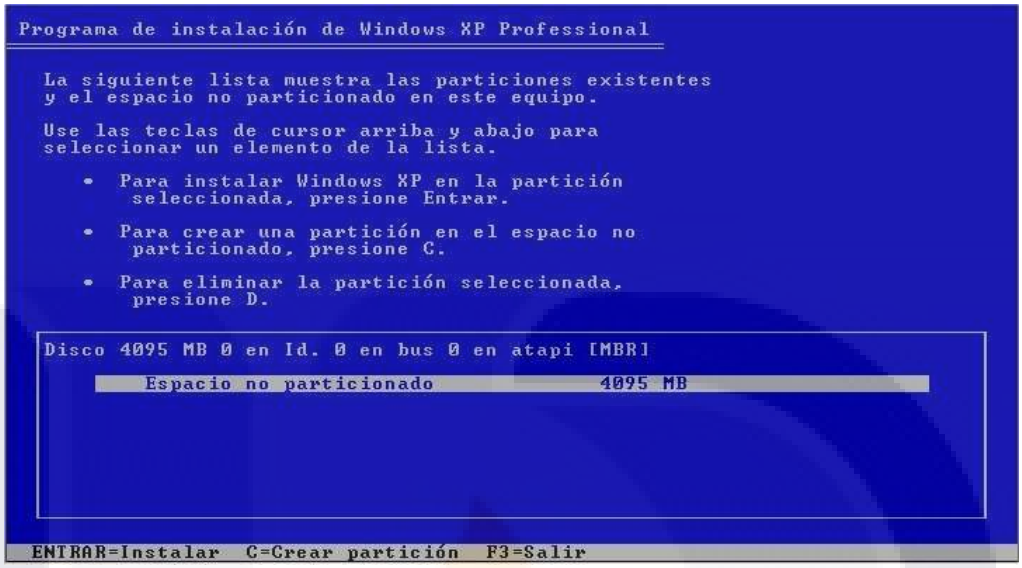
3. En la siguiente pantalla se presionó la tecla Enter para instalar Windows XP.



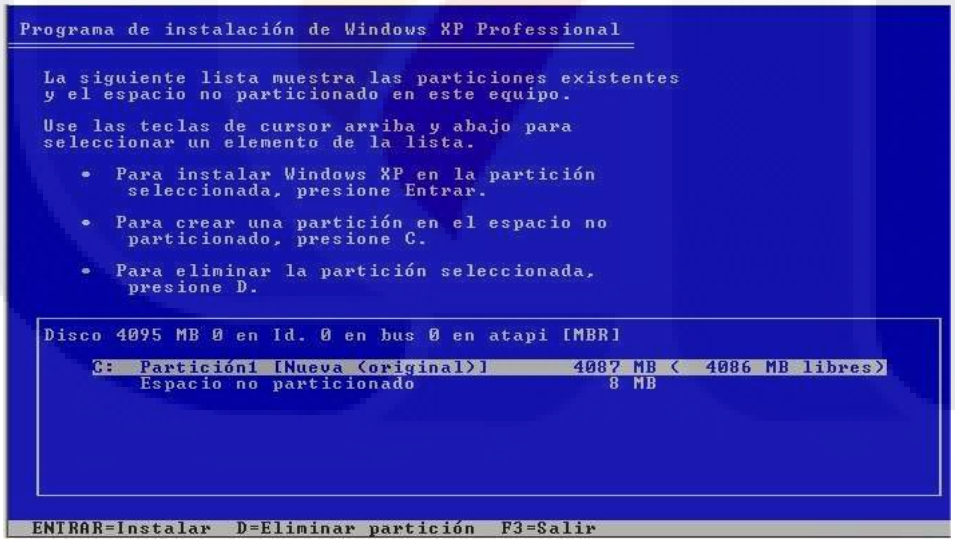
4. Se acepto el contrato de licencia para el usuario final presionado la tecla F8:



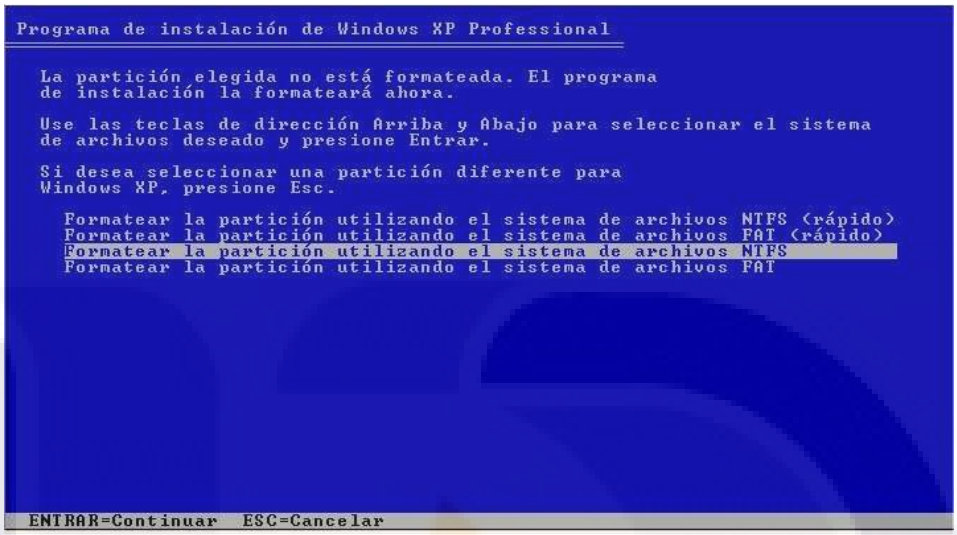
5. Se seleccionó el “espacio no particionado” y se presionó la tecla Enter:



6. Se seleccionó la nueva partición y se presionó la tecla Enter:

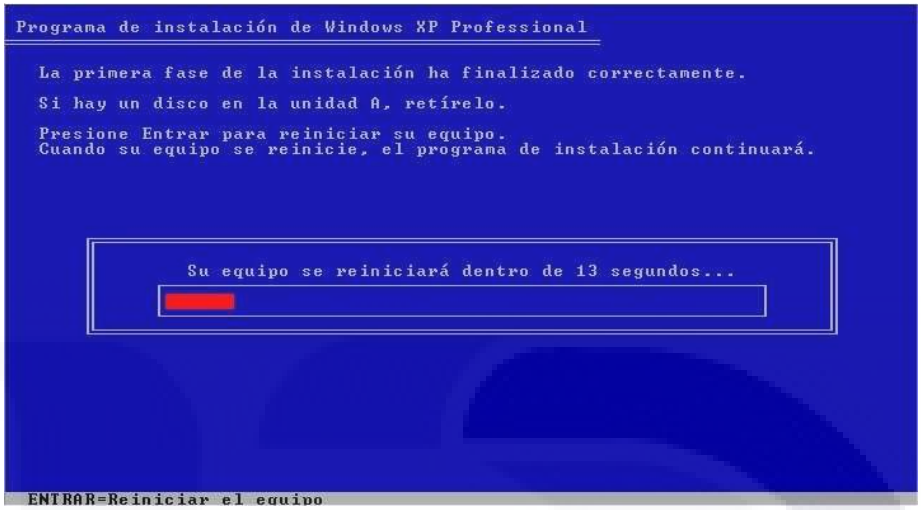


7. Se formateó usando la tercer opción: “Formatear la partición utilizando el sistema de archivos NTFS” y se presionó la tecla Enter:

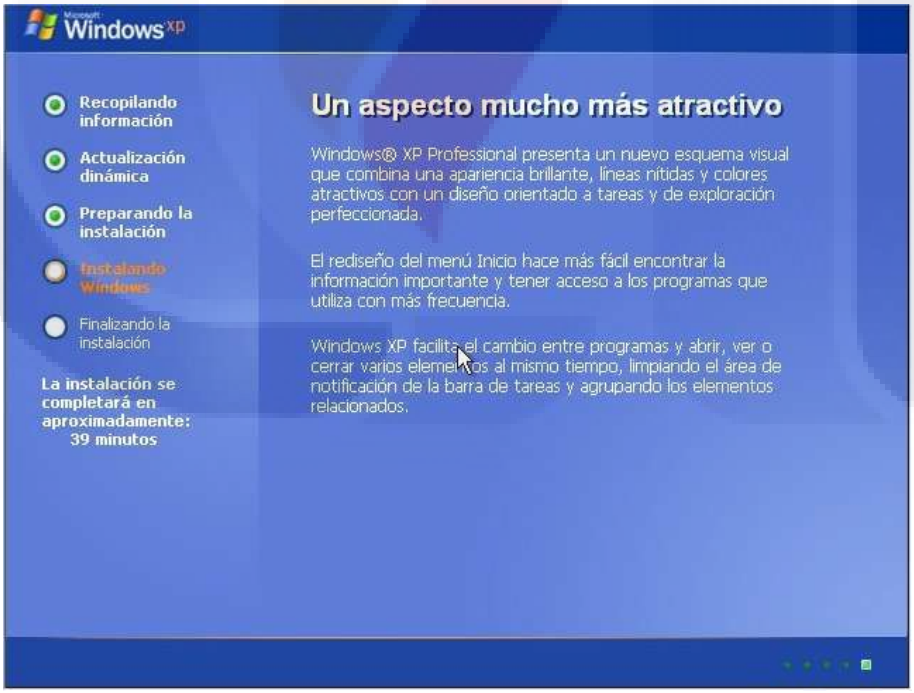


8. Se inició la copia de archivos del CD al disco duro:

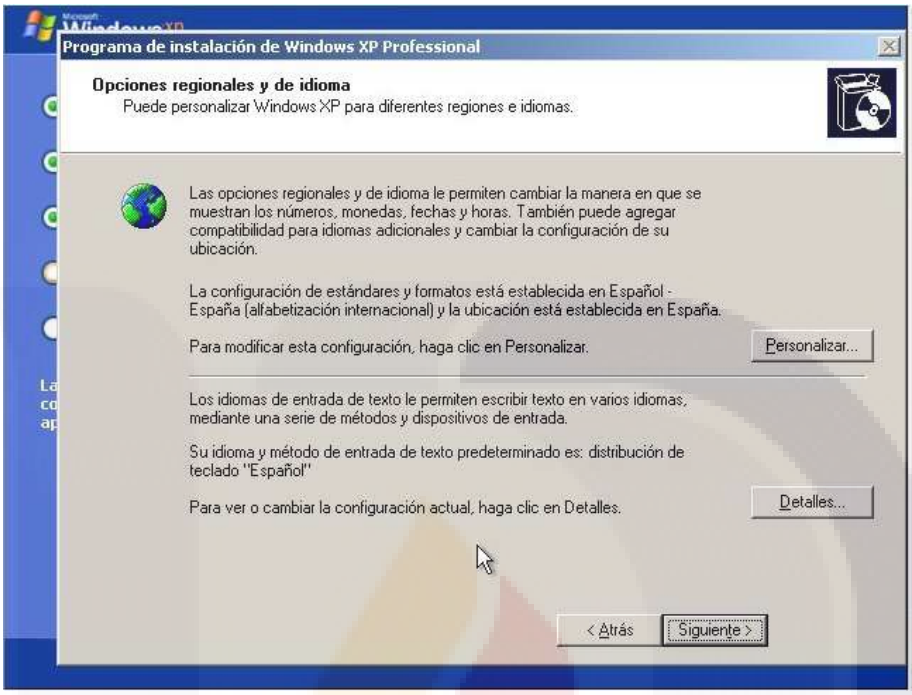
9. Terminada la copia de archivos el equipo se reinició automáticamente:



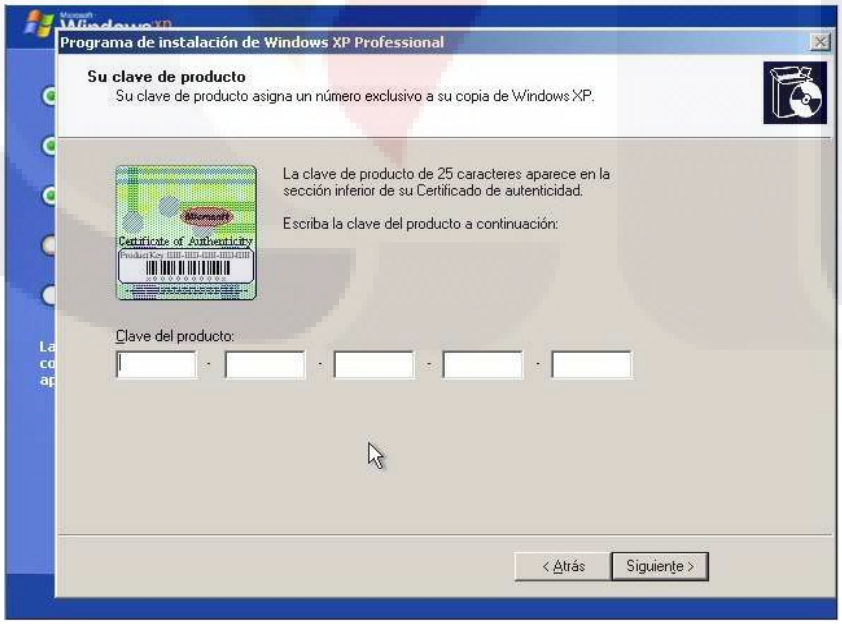
10. Al arrancar el equipo nuevamente no se presionó ninguna tecla, ya que ahora se arrancó desde los archivos copiados al disco duro, aquí comenzó la instalación en modo gráfico:



11. Se nos presentó las opciones regionales de idioma, se eligió la opción por defecto (Alfabetización internacional) y se presionó el botón Siguiente:



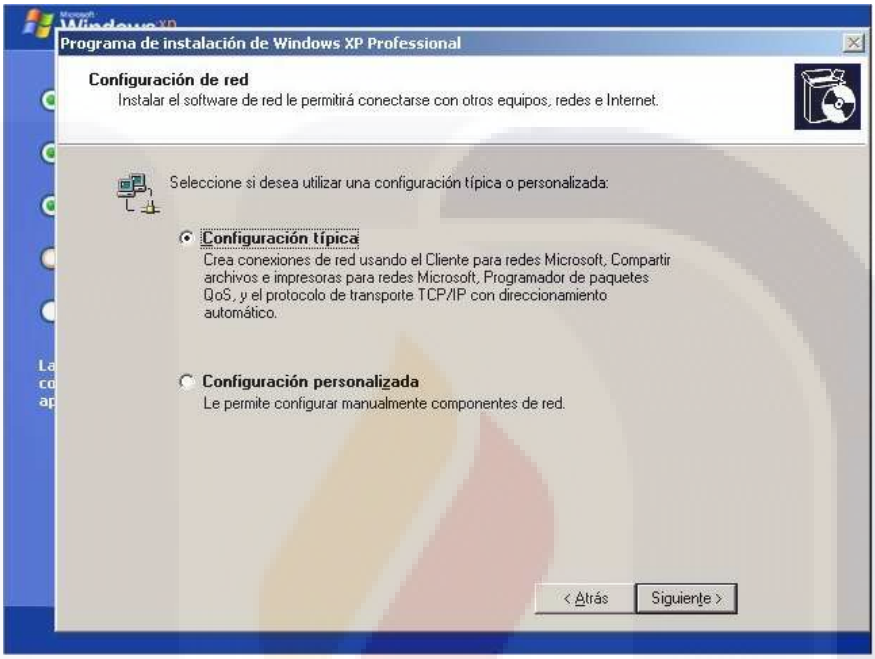
12. Se introdujo la clave para activar Windows XP y se presionó el botón Siguiente:



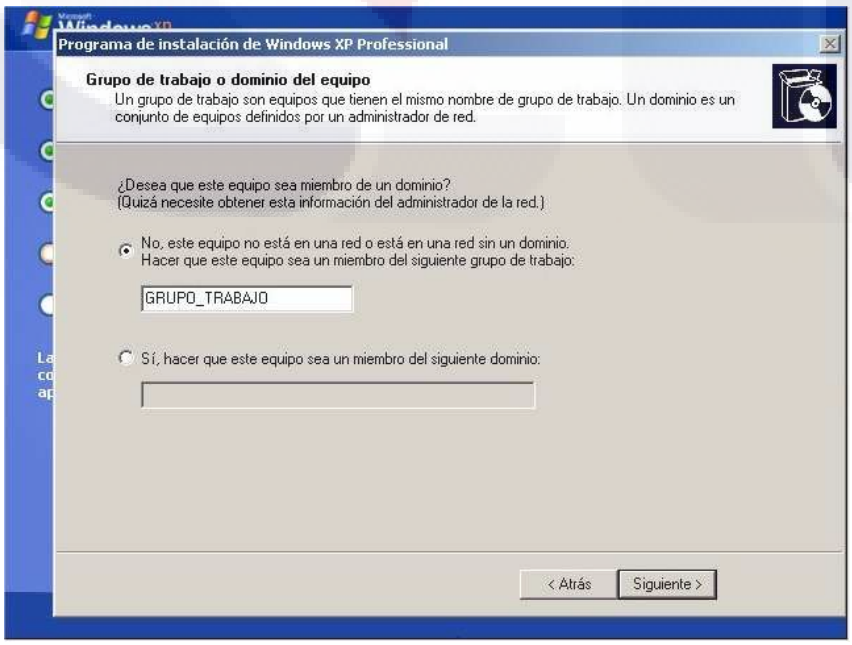
13. A continuación se configuró:

- El nombre de equipo y se estableció la contraseña de administrador.
- Se estableció la fecha y la hora del equipo.

14. En la instalación de la red se seleccionó "Configuración típica" y se presionó el botón Siguiente:



15. No fue necesario que el equipo perteneciera a un dominio, por lo que se seleccionó la primera opción y se presionó el botón Siguiente:



16. La instalación continuó con los siguientes procesos automatizados:

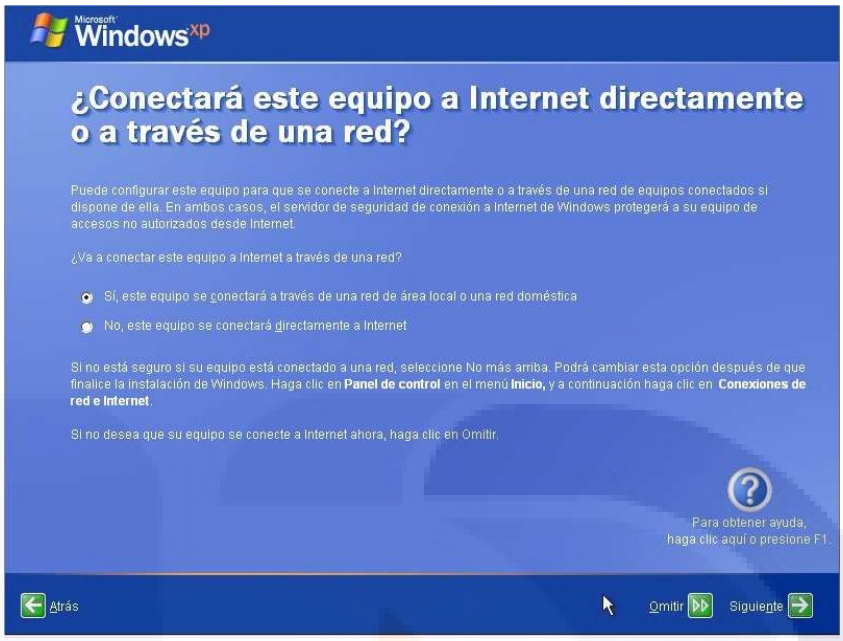
- “Copia de archivos”.
- “Completando de instalación”.
- “Instalando elementos del menú de inicio”.
- “Registrando componentes”.
- “Guardando configuración”.

El equipo se reinició automáticamente, en este momento el CD de instalación de Windows XP fue retirado de la unida de CD. Se inició Windows XP Professional por primera ocasión desde el disco duro:

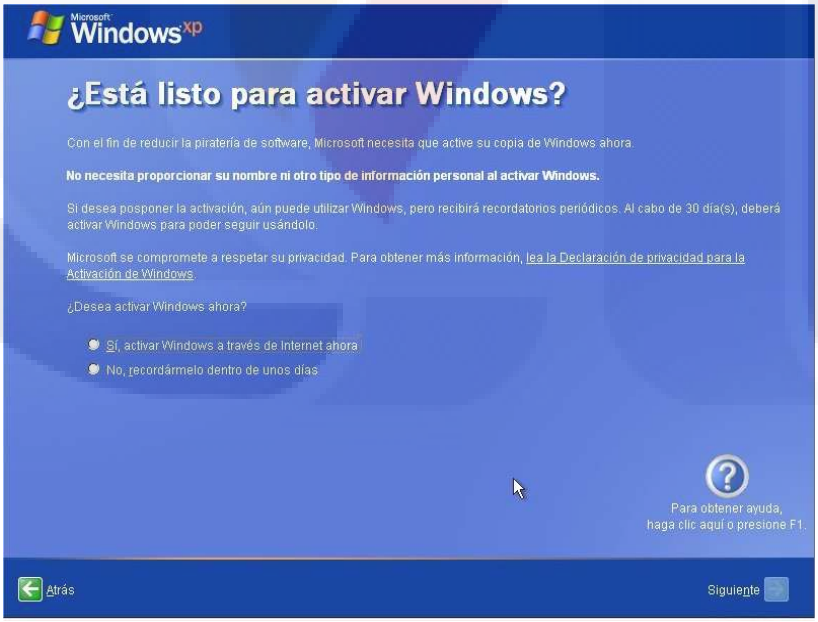
17. La instalación continuó con los siguientes procesos:

- “Configuración de la resolución de pantalla”.
- “Comprobación de su conexión a Internet”, la cual fue omitida.

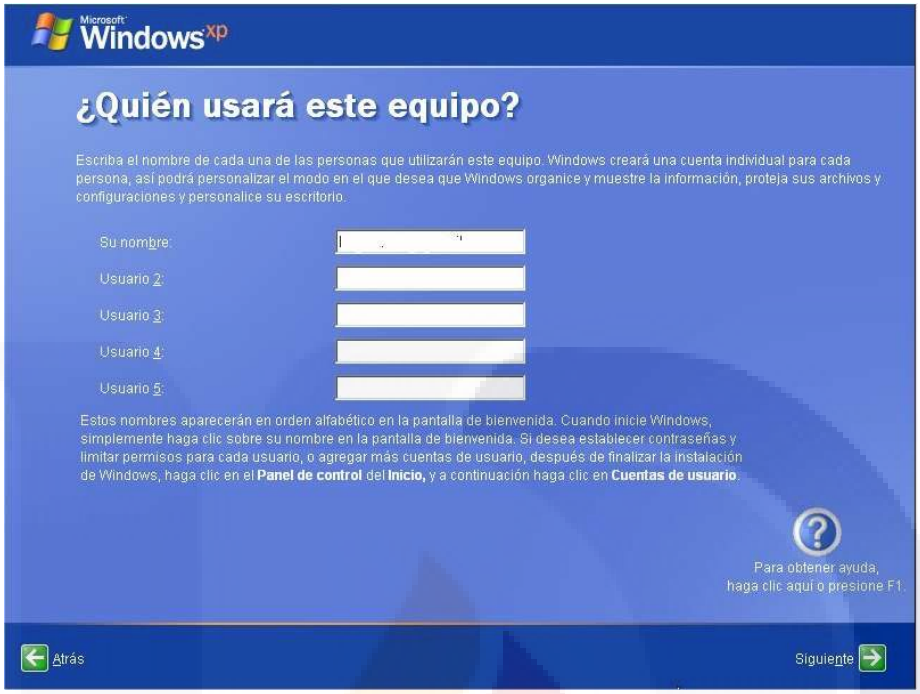
18. En la siguiente pantalla el equipo fue configurado con la primera opción: “Si, este equipo se conectará a través de una red de área local o una red doméstica” y se presionó el botón Siguiente:



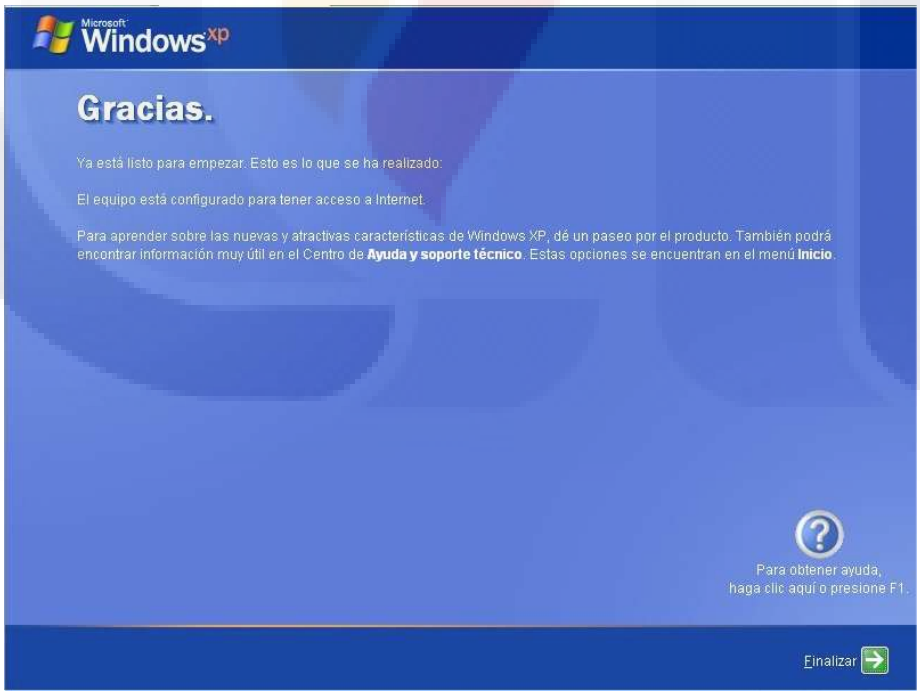
19. No fue necesario activar esta instalación de Windows XP por lo que fue seleccionada la segunda opción: “No, recordármelo dentro de unos días” y se presionó el botón Siguiente:



20. Se escribió el nombre del usuario principal y se presionó el botón Siguiente:



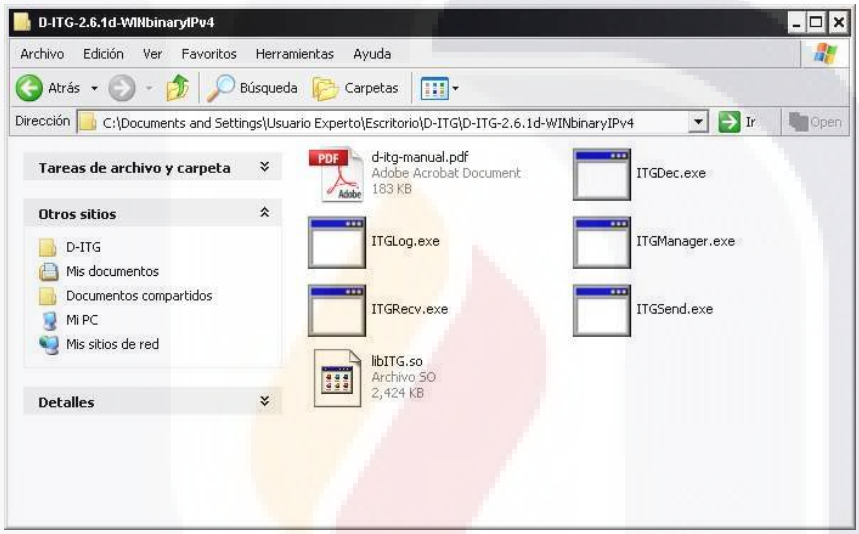
21. La instalación finalizó correctamente y se presionó el botón Finalizar:



2. Instalación de D-ITG.

1. La versión de D-ITG está compilada y lista para utilizarse solo fue necesario descargarla del siguiente enlace: http://www.grid.unina.it/software/ITG/codice/D-ITG-2.6.1d-WINbinaryIPv4.zip

2. El archivo D-ITG-2.6.1d-WINbinaryIPv4.zip fue descargado y se desempaquetado en la siguiente ruta: C:\Documents and Settings\Usuario Experto\Escritorio\D-ITG\D-ITG-2.6.1d-WINbinaryIPv4:



3. El contenido del archivo D-ITG-2.6.1d-WINbinaryIPv4.zip fue el siguiente:

- d-itg-manual.pdf
- ITGDec.exe
- ITGLog.exe
- ITGManager.exe
- ITGRecv.exe
- ITGSend.exe
- libITG.so

4. El Manual de D-ITG-2.6 fue descargado desde el siguiente enlace: http://www.grid.unina.it/software/ITG/codice/D-ITG2.6.1d-manual.pdf

Dado que en este equipo solo fue utilizado ITGSend.exe, en las siguientes 5 hojas se presenta la parte de ITGSend del manual de D-ITG.

1 ITGSend

1.1 Synopsis

In case of using a script file to generate multiple flows, type:

ITGSend ⟨script_file⟩ [-l [⟨logfile⟩]] [-L [⟨log_server_addr⟩] [⟨protocol_type⟩]] [-x [⟨receiver_logfile⟩]] [-X [⟨log_server_addr⟩] [⟨protocol_type⟩]]

If you want to remotely control the sender, launch it in daemon mode:

ITGSend -Q [-l [⟨logfile⟩]] [-L [⟨log_server_addr⟩] [⟨protocol_type⟩]] [-x [⟨receiver_logfile⟩]] [-X [⟨log_server_addr⟩] [⟨protocol_type⟩]]

Otherwise if you want to generate a single flow:

ITGSend [-m ⟨msr_type⟩] [-a ⟨destination_address⟩] [-rp ⟨destination_port⟩] [-sp ⟨source_port⟩] [-T ⟨protocol_type⟩] [-f ⟨TTL⟩] [-b ⟨DS byte⟩] [-rk ⟨receiver_serial_iface⟩] [-sk ⟨sender_serial_iface⟩] [-D] [-P] [-s ⟨seed⟩] [-t ⟨duration⟩] [-d ⟨gen_delay⟩] [-p ⟨payload_log_type⟩] [-j ⟨enable_idt_recovery⟩] [-l [⟨logfile⟩]] [-L [⟨log_server_addr⟩] [⟨protocol_type⟩]] [-x [⟨receiver_logfile⟩]] [-X [⟨log_server_addr⟩] [⟨protocol_type⟩]] [[-C ⟨pkts_per_s⟩ | -U ⟨min_pkts_per_s⟩ ⟨max_pkts_per_s⟩ | -E ⟨average_pkts_per_s⟩ | -V ⟨shape⟩ ⟨scale⟩ | -Y ⟨shape⟩ ⟨scale⟩ | -N ⟨mean⟩ ⟨std_dev⟩ | -O ⟨average_pkts_per_s⟩ | -G ⟨shape⟩ ⟨scale⟩] [-c ⟨pkt_size⟩ | -u ⟨min_pkt_size⟩ ⟨max_pkt_size⟩ | -e ⟨average_pkt_size⟩ | -v ⟨shape⟩ ⟨scale⟩ | -y ⟨shape⟩ ⟨scale⟩ | -n ⟨mean⟩ ⟨std_dev⟩ | -o ⟨average_pkt_size⟩ | -g ⟨shape⟩ ⟨scale⟩]] | [Telnet | DNS | CSa | CSi | Quake3 | VoIP [-x ⟨codec_type⟩] [-h ⟨protocol_type⟩] [-VAD]]

NOTE: launching ITGSend in background requires to redirect stdin to /dev/null

1.2 Description

Sender Component of the D-ITG Platform.
The script mode enables ITGSend to simultaneously generate several flows. Each flow is managed by a single thread, with a separate thread acting as a master and coordinating the other threads. To generate n flows, the script file has to contain n lines, each of which is used to specify the characteristics of one flow. Each line can contain all the options illustrated in Section 1.3, but those regarding the logging process (-l, -L, -X, -x). Such options can be specified at the command line and refer to all the flows.

1.3 Options

Flow options:

–m ⟨msr_type⟩	Set the type of meter. Two values are allowed: owdm (one way delay meter) and rttm (round trip time meter). Default is owdm. D-ITG does not provide any sort of synchronization among senders and receivers. In order to correctly measure packet One Way Delay (OWD), the clocks of sender and receiver must be synchronized by other means. Otherwise, we suggest to use the Round Trip Time (RTT) meter;
–a ⟨destination_address⟩	Set the destination address. Default is localhost;
–rp ⟨destination_port⟩	Set the destination port. Default is 8999;
–sp ⟨source_port⟩	Set the source port. If this option is not specified, the source port is set by the operating system;
–T ⟨protocol_type⟩	Set the protocol type. Valid values are UDP, TCP, ICMP, SCTP, DCCP. Default is UDP. If you choose ICMP you must specify the type of message. Root privileges are needed under Linux;
–f ⟨TTL⟩	Set the time to live (TTL). The value is interpreted as a decimal number, or as an hexadecimal number if the prefix "0x" is used. The range is [0, 255];
–b ⟨DS byte⟩	Set the DS byte for QoS tests. The value is interpreted as a decimal number, or as an hexadecimal number if the prefix "0x" is used. The range is [0, 255]. Default is 0 (Note: this option is disabled under Windows 2000 and XP, according to the "Microsoft Knowledge Base Article - 248611" *http://support.microsoft.com/Default.aspx?scid=kb;EN-US;q248611*; under Linux, root privileges are needed to set the DS byte to a value greater than 160);
–rk ⟨receiver_serial_iface⟩	Instructs the receiver to raise a signal on the specified serial interface every time a packet is received. Typical values are COM1, COM2, etc. under Windows and ttys0, ttys1, etc. under Linux;
–sk ⟨sender_serial_iface⟩	Raises a signal on the specified serial interface every time a packet is sent;
–s ⟨seed⟩	Set the seed for the random number generator. By default a random value is taken;
–D	Disable Nagle Algorithm;
–P	Enable high thread priority (only Windows platform);
–t ⟨duration⟩	Set the generation duration. It is expressed in milliseconds. Default is 10000 ms;
–d ⟨gen_delay⟩	Set the generation delay. It is expressed in milliseconds. Default is 0.

–p ⟨payload_log_type⟩	Set the type of information sent in the payload of each packet. Valid values are: 0 no information is sent in the payload packet; 1 only the sequence numbers are sent in the payload packet; 2 standard informations are sent in the payload packet. Default value is 2.
–j ⟨enable_idt_recovery⟩	Enable (1) or disable (0) the strategy used to guarantee the mean bitrate. Default value is 0.

Log options:

–l [⟨logfile⟩]	Generate the log file. If the meter type is OWDM and this option is omitted, ITGSend does not generate the log file. If the meter type is RTTM and this option is omitted, ITGSend generates a log file with the default log file name. The default log file name is /tmp/ITGSend.log under Linux and ITGSend.log under Windows;
–x [⟨receiver_logfile⟩]	Generate the log file at the receiver side. The default log file name is /tmp/ITGRecv.log under Linux and ITGRecv.log under Windows;
–L [⟨log_server_addr⟩] [⟨protocol_type⟩]	Remote log file. The first parameter is the log server IP address (default is localhost); the second parameter is the protocol used to rule the communication between the sender and the log server. Valid values are UDP and TCP (default is UDP). The log file name is specified by the –l option;
–X [⟨log_server_addr⟩] [⟨protocol_type⟩]	This option enables ITGRecv to remotely configure a log server. The first parameter is the log server IP address (default is localhost); the second parameter is the protocol used to rule the communication between the receiver and the log server. Valid values are UDP and TCP (default is UDP). The log file name is specified by the –x option;

Inter-departure time options:

–C ⟨pkts_per_s⟩	Constant inter-departure time (IDT)
–U ⟨min_pkts_per_s⟩ ⟨max_pkts_per_s⟩	Uniformly distributed IDT
–E ⟨average_pkts_per_s⟩	Exponentially distributed IDT
–V ⟨shape⟩ ⟨scale⟩	Pareto distributed IDT
–Y ⟨shape⟩ ⟨scale⟩	Cauchy distributed IDT
–N ⟨mean⟩ ⟨std_dev⟩	Normal distributed IDT
–O ⟨average_pkts_per_s⟩	Poisson distributed IDT
–G ⟨shape⟩ ⟨scale⟩	Gamma distributed IDT

NOTE: The IDT random variable provides the inter-departure time expressed in milliseconds. For the sake of simplicity, in case of Constant, Uniform, Exponential and Poisson variables, each parameter, say it x, is specified as a packet rate value. It is then converted to a time interval value ($x \rightarrow \frac{1000}{x}$). If no option

is specified, a constant IDT with 1000 packets per second is assumed.

Packet size options:

–c ⟨pkts_size⟩	Constant payload size.
–u ⟨min_pkts_size⟩ ⟨max_pkts_size⟩	Uniformly distributed payload size
–e ⟨average_pkts_size⟩	Exponentially distributed payload size
–v ⟨shape⟩ ⟨scale⟩	Pareto distributed payload size
–y ⟨shape⟩ ⟨scale⟩	Cauchy distributed payload size
–n ⟨mean⟩ ⟨std_dev⟩	Normal distributed payload size
–o ⟨average_pkts_size⟩	Poisson distributed payload size
–g ⟨shape⟩ ⟨scale⟩	Gamma distributed payload size

NOTE: If no option is specified, a constant payload size of 512 bytes is assumed.

Transport Layer protocols:

TCP Generates traffic using Transmission Control Protocol.

UDP Generates traffic using User Datagram Protocol.

DCCP Generates traffic using the Datagram Congestion Control Protocol, a message-oriented protocol like UDP with some new features. DCCP implements not only congestion control and congestion control negotiation, but also reliable connection setup, teardown, and feature negotiation. No options are required.

SCTP Generates traffic using the Stream Control Transmission Protocol. At the moment no special features of this protocol have been implemented yet. Soon multi-streaming can be managed. Two options are required: the first is an identificator of the session whom it belongs to and the second is the max number of streams of the session. NOTE: All streams belonging to the same session have to be specified with the same values for all the two options.

Application Layer protocols:

DNS Generate traffic with DNS traffic characteristics. No option is required. NOTE: DNS traffic generation works with both UDP and TCP transport layer protocols. Different settings will be ignored.

Telnet Generate traffic with Telnet traffic characteristics. No option is required. NOTE: Telnet traffic generation only works with TCP transport layer protocol. Different settings will be ignored.

VoIP Generate traffic with VoIP traffic characteristics. NOTE: VoIP traffic generation only works with UDP transport layer protocol. Different settings will

be ignored. VoIP options are:

–x ⟨codec_type⟩	Set the Codec type. VALUES:
	G.711.1 for G.711 codec with 1 sample per pkt (default)
	G.711.2 for G.711 codec with 2 samples per pkt
	G.723.1 for G.723.1 codec
	G.729.2 for G.729 codec with 2 samples per pkt
	G.729.3 for G.729 codec with 3 samples per pkt
–h ⟨protocolLtype⟩	Set the protocol type. VALUES:
	RTP for Real Time Protocol (default)
	CRTP for Real Time Protocol with header compression
–VAD	Set the Voice Activity Detection (it is off by default).

CSa Generate traffic with Counter Strike traffic characteristics related the active phase of the game. No option is required. NOTE: CSa traffic generation only works with UDP Transport Layer protocol. Different settings will be ignored [1].

CSi Generate traffic with Counter Strike traffic characteristics related the inactive phase of the game. No option is required. NOTE: CSi traffic generation only works with UDP Transport Layer protocol. Different settings will be ignored [1].

Quake3 Generate traffic with Quake III Arena traffic characteristics. No option is required. NOTE: Quake traffic generation only works with UDP Transport Layer protocol. Different settings will be ignored [2].

NOTE: If you specify an application layer protocol then you cannot specify any inter-departure time or packet size option. The other options illustrated above are allowed. If you want to specify an application layer protocol you must indicate it after every other option.

3. Instalación de Interfaz Gráfica Itggui-0911.

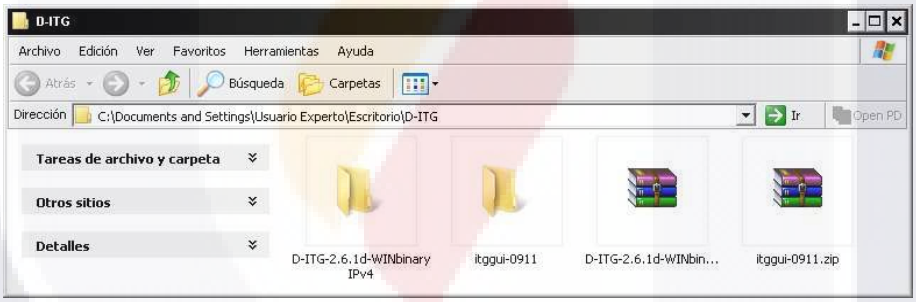
Para facilitar el uso de ITGSend fue utilizada la interfaz, Itggui-0911 esta interfaz está escrita en Java y nos permitió manejar todas las opciones de D-ITG 2.6

Los requerimientos de Itggui-0911 fueron:

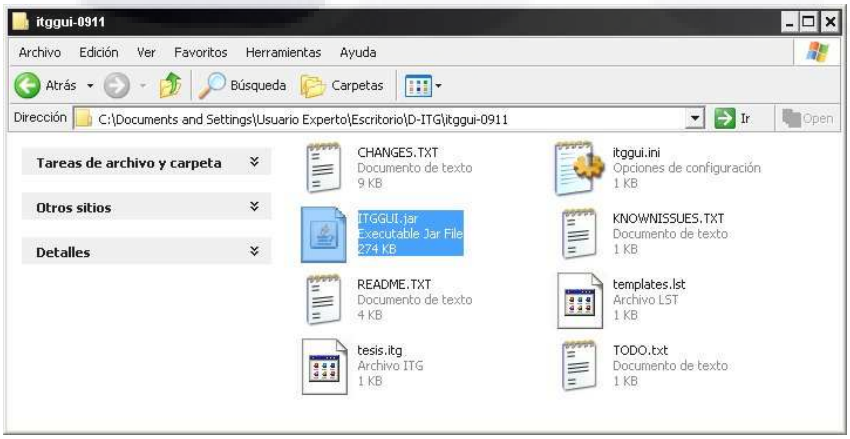
- D-ITG 2.6
- Java: la cual fue descargada desde el siguiente enlace: http://www.java.com/es/download/manual.jsp

1. Una vez instalado Java tggui-0911 fue descargada desde el siguiente enlace: http://www.semken.com/downloads/itggui-0911.zip

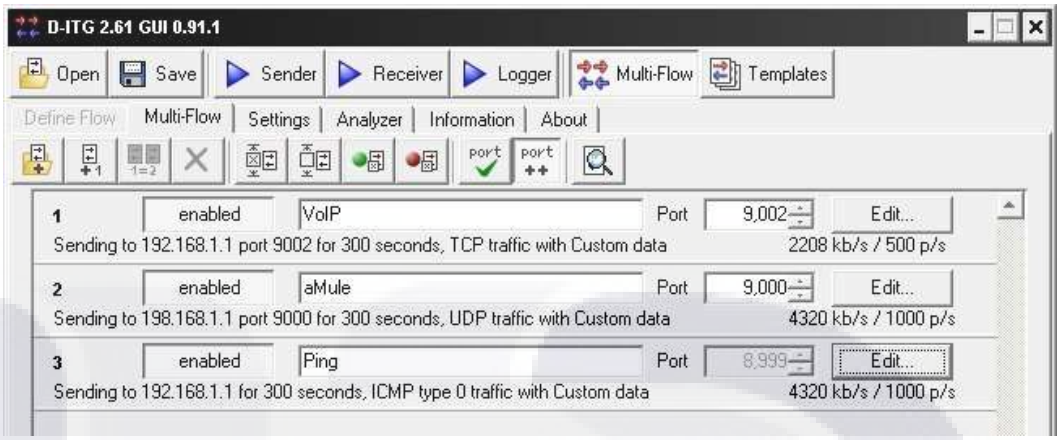
2. Fue necesario desempaquetar Itggui-0911 en el mismo directorio donde se dejó D-ITG:



3. Para iniciar la interfaz grafica solo fue necesario dar clic en el archivo ITGGUI.jar



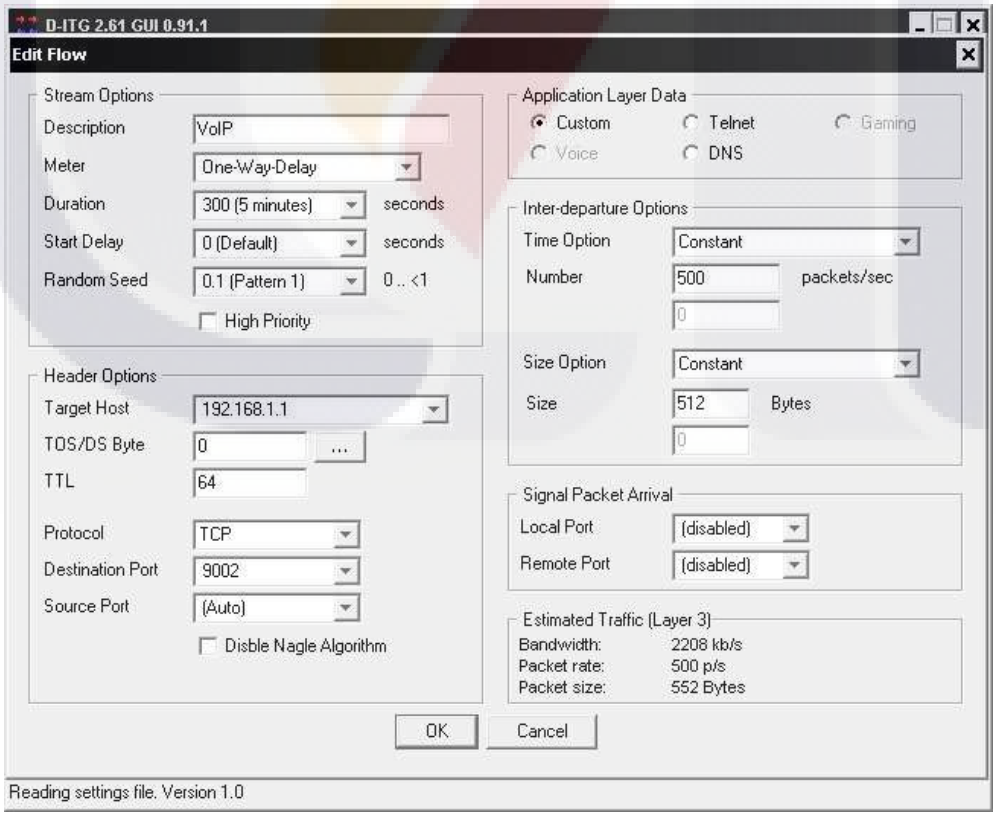
4. Se configuraron 2 Tráficos TCP; VoIP y aMule, así como un tráfico ICMP, esta configuración fue guardada para usar en los experimentos de los 3 escenarios:



Los flujos fueron configurados para ser enviados al Equipo A con IP 192.168.1.1

5. La configuración para cada uno de los tráficos mencionados se muestra a continuación:

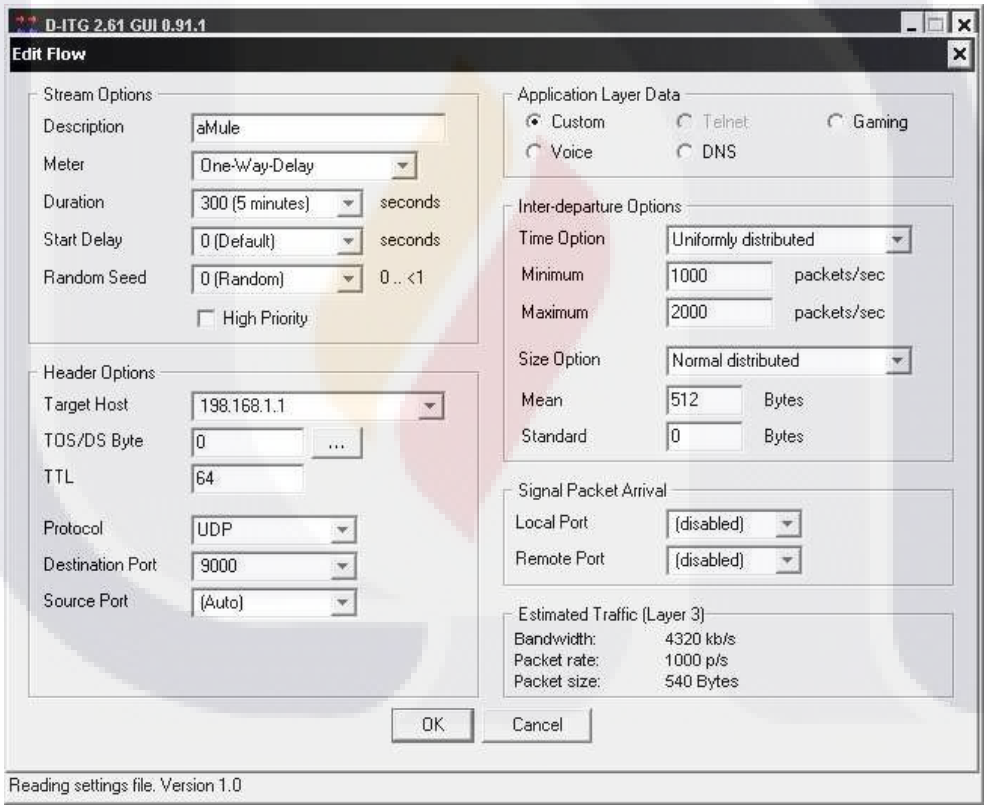
Configuración del tráfico VoIP:



Se puede observar los siguientes parámetros de configuración más importantes:

- Duración de 5 minutos.
- IP del equipo destino 192.168.1.1
- Paquetes TCP
- Tiempo de partida entre los paquetes controlado por una distribución Constante.
- Tamaño de los paquetes controlado por una distribución Constante.

Configuración del tráfico aMule:

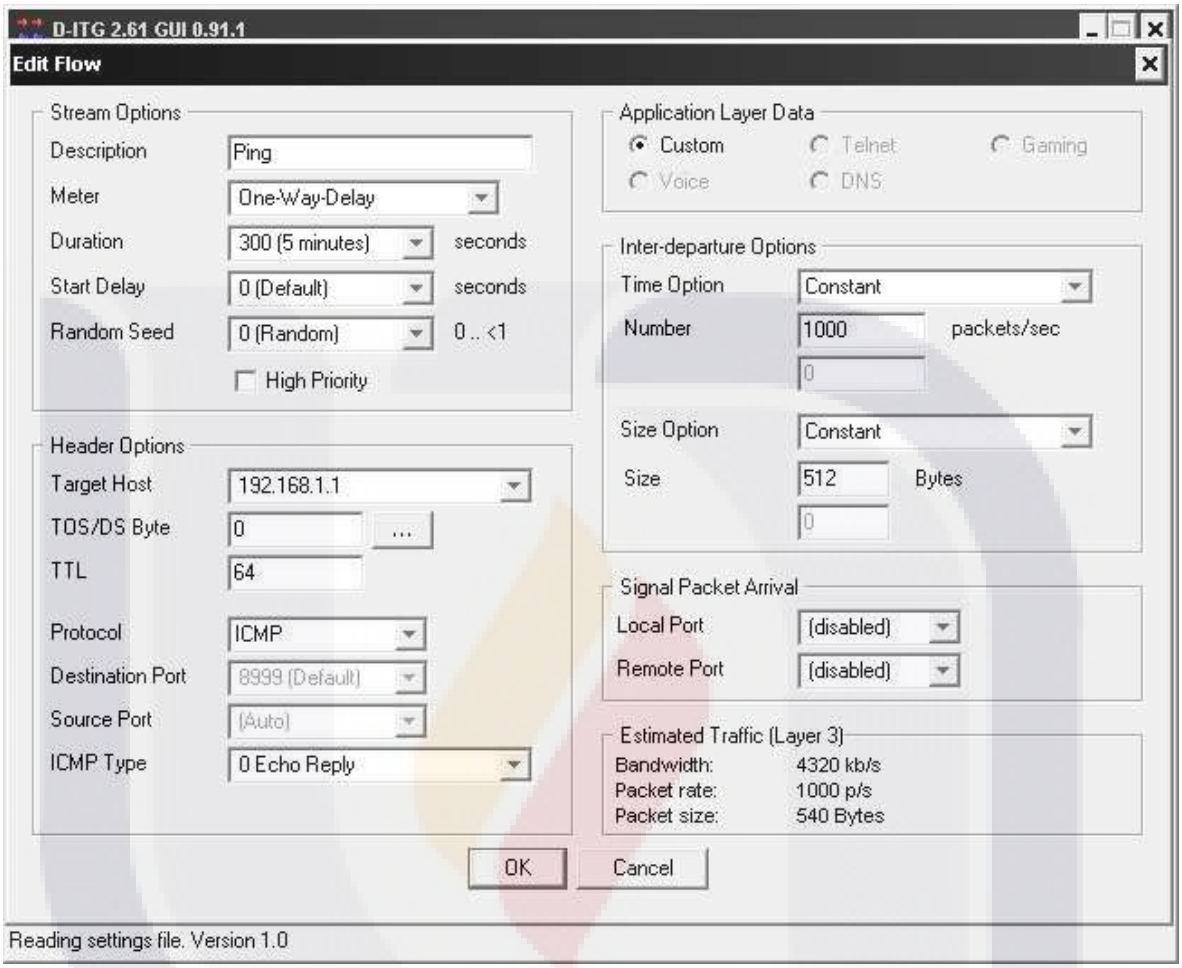


Se puede observar los siguientes parámetros de configuración más importantes:

- Duración de 5 minutos.
- IP del equipo destino 192.168.1.1
- Paquetes TCP
- Tiempo de partida entre los paquetes controlado por una distribución Uniforme.

- Tamaño de los paquetes controlado por una distribución Normal.

Configuración del tráfico ICMP - PING



Se puede observar los siguientes parámetros de configuración más importantes:

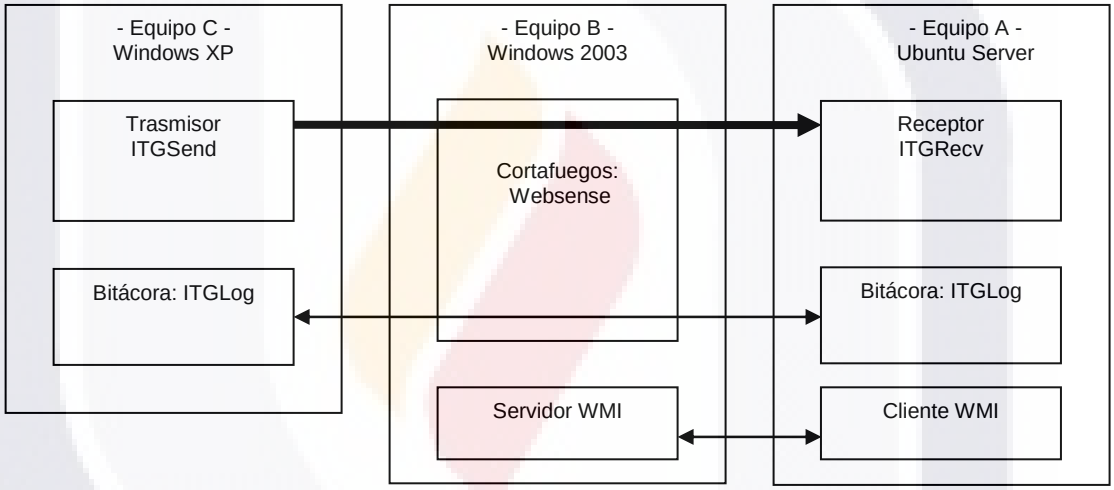
- Duración de 5 minutos.
- IP del equipo destino 192.168.1.1
- Paquetes ICMP
- Tiempo de partida entre los paquetes controlado por una distribución Constante.
- Tamaño de los paquetes controlado por una distribución Constante.

Procedimiento de obtención de datos en los tres escenarios.

En esta sección se describe el procedimiento realizado para obtener los datos de CPU, memoria RAM y jitter de los paquetes multimedia. Como se mencionó el experimento tuvo una duración de 5 minutos y se tomaron 150 muestras es decir una muestra cada 2 segundos. Los datos que se necesitaron fueron:

1. Porcentaje de CPU utilizado.
2. Bits de memoria RAM utilizada.
3. jitter de Paquetes en el flujo multimedia.

Escenario uno: Cortafuegos en Windows.



1. Obtención de Datos de CPU y memoria con el Cliente WMI:

1. Se programó un script llamado Websense-Colector.sh en el equipo A para automatizar la recolección el uso de CPU y memoria RAM usando **WMI** con el siguiente contenido:

```
#!/bin/bash

date > $3
for ((i=0; i< $2; i++))
do
date +%T >> $3
wmic -U tesis/Administrator%nimda //172.16.1.15 "select CommittedBytes,AvailableBytes,CommitLimit from
Win32_PerfRawData_PerfOS_Memory" >> $3
wmic -U tesis/Administrator%nimda //172.16.1.15 "select PercentProcessorTime from
Win32_PerfFormattedData_PerfOS_Processor" >> $3
sleep $1
done
```

2. La forma de usar el script fue la siguiente:

Websense-Colector.sh (retardo en segundos) (numero de muestras) (nombre del archivo a generar). El script fue utilizado de la siguiente manera:

```
root@router:/opt/D-ITG-2.6.1d# ./Websense-Colector.sh 1 150 LaboratorioWindows-5-Minutos
```

3. El scritp anterior cada dos segundos se conectó al Cortafuegos Websense y recolectó información de memoria disponible y CPU utilizados. La información obtenida fue necesaria procesarla ya que contenía una gran cantidad de información basura.

A continuación se muestra las primeras 50 líneas del archivo ***LaboratorioWindows-5-Minutos*** que correspondes a 6 de 150 muestras que se obtuvieron. Los datos en negritas son los datos que se necesitaron:

```
root@router:/opt/D-ITG-2.6.1d# tail -50 lab_windows.log
19:53:02
CLASS: Win32_PerfRawData_PerfOS_Memory
AvailableBytes|CommitLimit|CommittedBytes
54362112 |1192697856|638017536
CLASS: Win32_PerfFormattedData_PerfOS_Processor
Name|PercentProcessorTime
0|1
_Total|1
19:53:04
CLASS: Win32_PerfRawData_PerfOS_Memory
AvailableBytes|CommitLimit|CommittedBytes
54317056 |1192697856|638042112
CLASS: Win32_PerfFormattedData_PerfOS_Processor
Name|PercentProcessorTime
0|7
_Total|7
19:53:06
CLASS: Win32_PerfRawData_PerfOS_Memory
AvailableBytes|CommitLimit|CommittedBytes
54308864 |1192697856|638107648
CLASS: Win32_PerfFormattedData_PerfOS_Processor
Name|PercentProcessorTime
0|7
_Total|7
19:53:08
CLASS: Win32_PerfRawData_PerfOS_Memory
AvailableBytes|CommitLimit|CommittedBytes
54304768 |1192697856|638136320
CLASS: Win32_PerfFormattedData_PerfOS_Processor
Name|PercentProcessorTime
0|16
_Total|16
19:53:10
CLASS: Win32_PerfRawData_PerfOS_Memory
AvailableBytes|CommitLimit|CommittedBytes
54292480 |1192697856|638173184
CLASS: Win32_PerfFormattedData_PerfOS_Processor
Name|PercentProcessorTime
0|1
_Total|1
```

4. Fue necesario crear una macro en Microsoft Office 2003 para sacar los datos del % de uso de CPU y Bits de memoria libre mostrados en negritas, al archivo con la información de CPU se le puso el nombre ***LaboratorioWindows-CPU-5-Minutos.txt*** y al archivo con la información de memoria RAM se le puso el nombre de ***LaboratorioWindows-MEM-5-Minutos.txt***. A continuación se muestra los primeros 5 datos de ambos archivos:

% de CPU Utilizado
1
7
7
16

Bits de memoria Disponible:
54362112
54317056
54308864
54304768
54292480

5. Los datos de CPU ya se encontraban listos para utilizarse, por el contrario los datos de memoria fueron procesados para cambiarlos de Bits de memoria disponible a Bits de memoria utilizada, para lograr lo anterior fue necesario restarle a 494514176 que es el total de la memoria RAM a cada uno de los datos de Bits de memoria disponible, para automatizar este proceso se creó un script llamado ***fix_mem_windows.sh*** en el equipo A con el siguiente contenido:

```
#! /bin/bash
while read linea
do
FIX=$((494514176 - $linea))
echo $FIX >> Fix_$1
done < $1
```

El script anterior fue utilizado de la siguiente forma:

```
root@router:/opt/D-ITG-2.6.1d#  ./fix_mem_windows.sh  LaboratorioWindows-MEM-5-Minutos.txt
```

La instrucción anterior nos creo el archivo llamado: ***Fix_LaboratorioWindows-MEM-5-Minutos.txt*** a continuación se muestran los primeros 5 renglones de dicho archivo:

Bits de memoria Utilizada:
440152064
440197120
440205312
440209408
440221696

2. Obtención del jitter para el flujo Multimedia VoIP con ITGDec:

En esta pagina asi como las siguientes 2 se muestra el manual de ITGDec

4 ITGDec

4.1 Synopsis

ITGDec [⟨logfile⟩ [–v | –i] [–t] [–s] [–l ⟨text_log_file⟩] [–o ⟨octave_log_file⟩] [–d ⟨delay_interval_size⟩] [–j ⟨jitter_interval_size⟩] [–b ⟨bitrate_interval_size⟩] [–p ⟨packetloss_interval_size⟩] [–f ⟨max_flow_num⟩] [–P] [–I]] | [–h | –help]

4.2 Description

The ITGDec decoder is the utility to analyze the results of the experiments conducted by using the D-ITG generation platform. ITGDec parses the log files generated by ITGSend and ITGRecv and calculates the average values of bitrate, delay and jitter either on the whole duration of the experiment or on variable-sized time intervals. You can analyze the binary log file only on the operating system used to create that file. You can use another operating system if the log file is in text format. The *Total time* of the experiment is calculated as the difference between receiving time of last and first packet.
The displayed *Jitter* is an average value. It is calculated according to Figure 4.2.

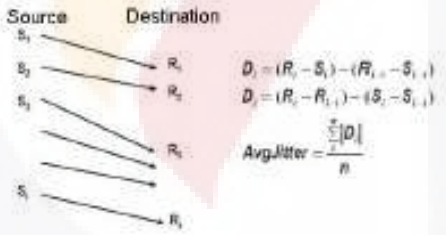


Figure 1: Jitter formula

The displayed *Delay* is calculated as the average of differences between receiving and sending times of packets. D-ITG does not provide any sort of synchronization among senders and receivers. In order to correctly measure packet One Way Delay (OWD) the clocks of sender and receiver must be synchronized by other means. Depending on the requested resolution, NTP or GPS can be used. In the case of synchronization issues we suggest to use the Round Trip Time (RTT) meter.
The displayed *Delay standard deviation* (σ) is calculated according to equation 1:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(d_i - \hat{d}\right)^2} \qquad (1)$$

where N is the number of packets considered, d_i is the delay of packet i, and $\hat{d}$ is the average delay of packets.

4.3 Options

–v	Print average and total results on screen in standard visualization format.
–i	Print average and total results on screen in a different format in which each field is separated by "*" charachter.
–t	The log file in input is considered to be in text format. If not specified the log file in input is considered to be in binary format.
–s ⟨personal_string⟩	The log file in input is split in N files where N is the number of flows detected in the log file. The names of the split files are in following format: "sender ip address"-"receiver ip address"-"flow number".⟨personal_string⟩.dat . If not provided, the Default value for ⟨personal_string⟩ is "log".
–l ⟨text_log_file⟩	Produce an output log file in text format with name ⟨text_log_file⟩ .
–o ⟨octave_log_file⟩	Generate a log file named "octave_log_file" that can be imported in Octave/Matlab.
–f ⟨flow_info⟩	If ⟨flow_info⟩ is a number, only the flows with flow number less or equal to ⟨flow_info⟩ will be considered. If ⟨flow_info⟩ = t all packets will be considered as belonging to the same flow.
–d ⟨delay_interval_size⟩	Every ⟨delay_interval_size⟩ milliseconds the average of transmission time of packets is calculated and printed in a file called "delay.dat". The structure of this file is as follows: a first column with time reference is present and is followed by different columns containing results calculated for each flow. A final column with aggregate result is also printed. In the first row "sender ip address"-"receiver ip address"-"flow number" is also printed in the columns related to single flows results. This is the reference for single flows information.

-b ⟨bitrate_interval_size⟩	Every ⟨bitrate_interval_size⟩ milliseconds the average of instantaneous bit rate of packets is calculated and printed in a file called "bitrate.dat". The structure of this file is as follows: a first column with time reference is present and is followed by different columns containing results calculated for each flow. A final column with aggregate result is also printed. In the first row "sender ip address"-"receiver ip address"-"flow number" is also printed in the columns related to single flows results. This is the reference for single flows information.
-j ⟨jitter_interval_size⟩	Every ⟨jitter_interval_size⟩ milliseconds the average of instantaneous jitter of packets is calculated and printed in a file called "jitter.dat". The structure of this file is as follows: a first column with time reference is present and is followed by different columns containing results calculated for each flow. A final column with aggregate result is also printed. In the first row "sender ip address"-"receiver ip address"-"flow number" is also printed in the columns related to single flows results. This is the reference for single flows information.
-p ⟨packet_loss_interval_size⟩	Every ⟨packet_loss_interval_size⟩ milliseconds the average packet loss is calculated and printed in a file called "packetloss.dat". The structure of this file is as follows: a first column with time reference is present and is followed by different columns containing results calculated for each flow. A final column with aggregate result is also printed. In the first row "sender ip address"-"receiver ip address"-"flow number" is also printed in the columns related to single flows results. This is the reference for single flows information.
-P	Print on screen the size of each packet.
-I	Print on screen the inter departure time between each packet.

4.4 Compatibility with previous versions

Version 2.4 of ITGDec is still capable of processing log files of D-ITG previous versions. To have a version of ITGDec that is compatible with 2.3 version (or previous) log files it is necessary to compile the source file (ITGDecod.cpp) by using a pre-compiler option. This option is `V23` and it has to be passed to the compiler with the directive `-D`.
Therefore, to compile ITGDec to support this feature it is necessary to modify the `Makefile` present in the `src` directory and to add the `-DV23` option. Alternatively it is possible to compile ITGDec alone, using the following syntax on the command line: "**g++ ITGDecod.cpp -DV23 -lm -o ITGDec**".

Con la información mostrada fue posible utilizar ITDec.

1. El programa ITGLog generó un archivo llamado ***test-cada2-segundos.log***. El cual fue necesario procesarlo con el programa ITGDec para obtener la información del jitter. De acuerdo al manual la sintaxis para obtener el jitter de manera general es la siguiente:

ITGDec [<archivo de bitácora>] [-j <intervalo en milisegundos>][-f <número de flujo>]

En nuestro caso para obtener el jitter del flujo numero 1 con periodos de 2 segundos la sintaxis utilizada fue: ***ITGDec ./test-cada2-segundos.log -f 1 -j 2000***
A continuación se muestra el comando ejecutado en la consola del equipo A:

```
root@router:/opt/D-ITG-2.6.1d#  ITGDec ./test-cada2-segundos.log -f 1 -j 2000
root@router:/opt/D-ITG-2.6.1d#  1-172.16.1.17-172.16.1.254.dat.jit
root@router:/opt/D-ITG-2.6.1d#  mv 1-172.16.1.17-172.16.1.254.dat.jit test-cada2-segundos-
```

2. El archivo generado fue ***test-cada2-segundos-JITTER.log*** , los primeros 10 datos se muestran a continuación:

```
0.000000 0.001859
2.000000 0.001408
4.000000 0.001474
6.000000 0.000823
8.000000 0.000669
10.000000 0.000714
12.000000 0.000719
14.000000 0.000670
16.000000 0.000835
18.000000 0.000666
```

3. Se puede observar que la primera columna hace referencia al tiempo en segundos de la muestra, esta columna de información no fue necesaria así que fue eliminada, el resultado de los datos anteriores se muestra a continuación:

```
0.001859
0.001408
0.001474
0.000823
0.000669
0.000714
0.000719
0.000670
0.000835
0.000666
```

Los datos utilizados para la prueba de hipótesis correspondientes al primer escenario se muestran el la siguiente tabla.

Número de Muestra	% CPU Utilizado	Bits MEM Utilizada	Multimedia jitter en Segundos
1	1	440152064	0.001859
2	7	440197120	0.001408
3	7	440205312	0.001474
4	16	440209408	0.000823
5	1	440221696	0.000669
6	0	440242176	0.000714
7	7	440254464	0.000719
8	9	440262656	0.00067
9	8	440057856	0.000835
10	7	440066048	0.000666
11	3	440078336	0.000671
12	0	440778752	0.000668
13	0	440758272	0.000744
14	3	440782848	0.000748
15	2	440815616	0.000728
16	5	440844288	0.00143
17	2	440832000	0.000861
18	1	440868864	0.00067
19	6	440913920	0.001768
20	6	440926208	0.000988
21	4	440811520	0.001004
22	3	440852480	0.001059
23	8	440864768	0.001006
24	0	440913920	0.001059
25	3	440897536	0.001004
26	7	440934400	0.000999
27	9	441028608	0.000995
28	13	441085952	0.000995
29	10	441106432	0.000986
30	8	441163776	0.000993
31	2	441208832	0.001049
32	13	441204736	0.000998
33	5	441196544	0.001006
34	8	441221120	0.001002
35	8	441311232	0.001065
36	0	441344000	0.001074
37	2	441307136	0.001
38	3	441352192	0.001061
39	8	441401344	0.00107
40	6	441434112	0.001067
41	12	441417728	0.001076
42	7	441458688	0.000991
43	0	441487360	0.000986

44	4	441552896	0.001083
45	0	441552896	0.00098
46	10	441577472	0.001001
47	6	441589760	0.000992
48	10	441626624	0.001036
49	6	441614336	0.001028
50	5	441769984	0.001072
51	1	441790464	0.001075
52	3	441806848	0.000998
53	1	441782272	0.000988
54	4	441286656	0.000979
55	5	441327616	0.001008
56	7	441380864	0.001012
57	0	441442304	0.000996
58	7	441507840	0.001008
59	1	441561088	0.001059
60	13	441602048	0.001077
61	10	441589760	0.001062
62	7	441651200	0.001014
63	1	441683968	0.001009
64	7	441712640	0.001006
65	3	441745408	0.001077
66	5	441675776	0.001002
67	5	441696256	0.000989
68	8	441733120	0.000979
69	7	441733120	0.001
70	1	441741312	0.000987
71	7	441798656	0.001081
72	1	441839616	0.001076
73	3	441827328	0.000998
74	3	441851904	0.000999
75	3	441888768	0.000997
76	4	441409536	0.000999
77	5	441380864	0.001141
78	7	441425920	0.000998
79	1	441450496	0.000999
80	0	441483264	0.000996
81	7	441470976	0.000989
82	5	441475072	0.001079
83	3	441524224	0.001006
84	2	441565184	0.000999
85	2	441577472	0.001074
86	1	441618432	0.000985
87	11	441720832	0.001144
88	7	441745408	0.000998
89	6	441815040	0.00101
90	6	441856000	0.001002

91	3	441917440	0.000988
92	7	441954304	0.00109
93	0	441958400	0.001133
94	2	442028032	0.001016
95	3	442093568	0.000993
96	11	442146816	0.001011
97	10	442126336	0.001054
98	5	442179584	0.001013
99	3	442245120	0.001076
100	0	442265600	0.000984
101	9	442277888	0.001026
102	7	442331136	0.001142
103	9	442355712	0.001138
104	9	442421248	0.001066
105	0	442433536	0.001008
106	2	442486784	0.00099
107	4	442511360	0.000993
108	12	442499072	0.001083
109	4	442548224	0.001074
110	4	442687488	0.001008
111	6	442658816	0.000994
112	1	442654720	0.001015
113	29	442691584	0.000986
114	7	442753024	0.001132
115	5	442810368	0.001087
116	13	442867712	0.000972
117	7	442933248	0.001018
118	57	443027456	0.000992
119	5	443068416	0.00108
120	7	443088896	0.001068
121	4	443133952	0.000999
122	7	443174912	0.000993
123	57	443232256	0.001143
124	2	443269120	0.001014
125	0	443342848	0.000984
126	5	443379712	0.001015
127	7	443424768	0.000989
128	57	443432960	0.001074
129	6	443478016	0.001003
130	3	443531264	0.000987
131	4	443600896	0.000991
132	1	443588608	0.000992
133	55	443637760	0.000997
134	6	443723776	0.001011
135	8	443777024	0.001068
136	3	443768832	0.00099
137	8	443846656	0.000996

138	58	443879424	0.000982
139	1	443916288	0.001138
140	10	443928576	0.001067
141	10	443936768	0.00106
142	9	444002304	0.000987
143	53	444030976	0.000996
144	5	444035072	0.001079
145	4	444080128	0.001065
146	9	444203008	0.000998
147	6	444297216	0.000996
148	57	444329984	0.001052
149	1	444424192	0.001075
150	3	444449192	0.000982

Tabla 4: Datos utilizados en las prueba de hipótesis para el escenario en Windows.

Escenario dos: Cortafuegos en GNU/Linux.

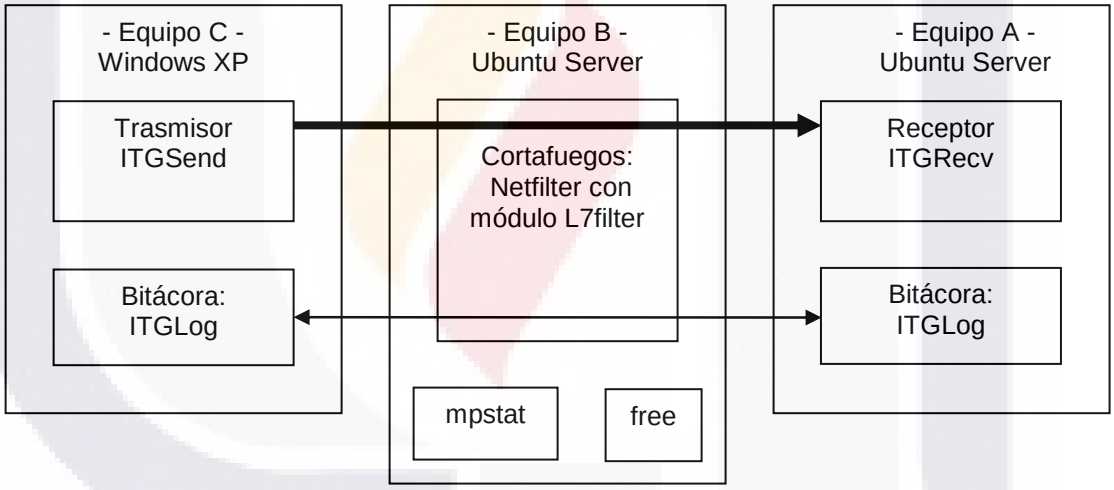


Diagrama lógico del segundo escenario: Cortafuegos en Windows: Netfilter + L7filter

1. Obtención de Datos de CPU y memoria con los comandos:

1. Se programó un script llamado ***Linux-Colector.sh*** el cual fue almacenado y ejecutado en el mismo cortafuegos para automatizar la recolección el uso de CPU y memoria RAM:

```
#!/bin/bash

date > $3_memoria
date > $3_cpu
for ((i=0; i< $2; i++))
do
free -bt | grep Mem | awk '{print $4}' >> $3_memoria
mpstat | grep all | awk '{print $11}' | tail -1 >> $3_cpu
sleep $1
done
```

Se usó el comando **mpstat** para obtener el porcentaje de CPU disponible y el comando **free** para obtener la memoria disponible.

2. La forma de usar el script fue la siguiente:
Linux-Colector.sh (retardo en segundos) (numero de muestras) (nombre del archivo a generar). El script fue utilizado de la siguiente manera:

```
root@firewall:/opt# ./Linux-Colector.sh 2 150 Multimedia_ICMP_Amule
```

3. El scritp anterior cada dos segundos recolectó información de memoria disponible dicha información fue almacenada en el archivo ***Multimedia_ICMP_Amule_memoria***, el CPU disponible fue almacenado en el archivo ***Multimedia_ICMP_Amule_cpu***. La información obtenida fue necesaria procesarla ya que se necesitaba obtener el CPU y memoria utilizados

4. Para la obtención de la memoria utilizada fue necesario cambiar los Bits de memoria disponible a Bits de memoria utilizada, para lograr lo anterior fue necesario restarle a 494514176 que es el total de la memoria RAM a cada uno de los datos de Bits de memoria disponible, para automatizar este proceso se creó un script llamado ***fix_mem_linux.sh*** en el equipo A con el siguiente contenido:

```
#! /bin/bash
while read linea
do
FIX=$((494514176 - $linea))
echo $FIX >> Fix_$1
done < $1
```

El script anterior fue utilizado de la siguiente forma:

```
root@router:/opt/D-ITG-2.6.1d#  ./fix_mem_linux.sh  Multimedia_ICMP_Amule_memoria
```

La instrucción anterior nos creo el archivo llamado: ***Fix_Multimedia_ICMP_Amule_memoria*** a continuación se muestran los primeros 5 renglones de dicho archivo:

```
72495104
72503296
72495104
72622080
72613888
```

5. Para la obtención del CPU utilizado fue necesario cambiar el porcentaje de CPU disponible a porcentaje de CPU utilizado, para lograr lo anterior fue necesario restarle 100 a cada uno de los datos de CPU disponible, para automatizar este proceso se creó un script llamado ***fix_cpu_linux.sh*** en el equipo A con el siguiente contenido:

```
#! /bin/bash
while read linea
do
FIX=$((100 - $linea))
echo $FIX >> Fix_$1
done < $1
```

El script anterior fue utilizado de la siguiente forma:

```
root@router:/opt/D-ITG-2.6.1d#  ./fix_cpu_linux.sh  Multimedia_ICMP_Amule_cpu
```

La instrucción anterior nos creo el archivo llamado: ***Fix_Multimedia_ICMP_Amule_cpu*** a continuación se muestran los primeros 5 renglones de dicho archivo:

```
5
5
5
5
5
```

2. Obtención del jitter para el flujo Multimedia VoIP con ITGDec:

1. El programa ITGLog generó un archivo llamado ***test-cada2-segundos.log***. fue necesario utilizar el programa ITGDec para obtener la información del jitter. De acuerdo al manual la sintaxis para obtener el jitter de manera general es la siguiente:

ITGDec [<archivo de bitácora>] [-j <intervalo en milisegundos>][-f <número de flujo>]

En nuestro caso para obtener el jitter del flujo numero 1 con periodos de 2 segundos la sintaxis utilizada fue: ***ITGDec ./test-cada2-segundos.log -f 1 -j 2000***
A continuación se muestra el comando ejecutado en la consola del equipo A:

```
root@router:/opt/D-ITG-2.6.1d#  ITGDec ./test-cada2-segundos.log -f 1 -j 2000
root@router:/opt/D-ITG-2.6.1d#  1-172.16.1.17-172.16.1.254.dat.jit
root@router:/opt/D-ITG-2.6.1d#  mv 1-172.16.1.17-172.16.1.254.dat.jit test-cada2-segundos-
```

2. El archivo generado fue ***test-cada2-segundos-JITTER.log*** , los primeros 10 datos se muestran a continuación:

```
0.000000 0.000225
2.000000 0.000258
4.000000 0.000147
6.000000 0.000228
8.000000 0.000141
10.000000 0.000141
12.000000 0.000145
14.000000 0.000150
16.000000 0.000713
18.000000 0.000149
```

3. Se puede observar que la primera columna hace referencia al tiempo en segundos de la muestra, esta columna de información no fue necesaria así que fue eliminada, el resultado de los datos anteriores se muestra a continuación:

```
0.000225
0.000258
0.000147
0.000228
0.000141
0.000141
0.000145
0.000150
0.000713
0.000149
```

Los datos utilizados para la prueba de hipótesis correspondientes al segundo escenario se muestran el la siguiente tabla.

Numero de Muestra	% CPU Utilizado	Bits MEM Utilizada	Multimedia jitter en Segundos
1	5	72495104	0.000225
2	5	72503296	0.000258
3	5	72495104	0.000147
4	5	72622080	0.000228
5	5	72613888	0.000141
6	5	72589312	0.000141
7	5	72622080	0.000145
8	5	72622080	0.00015
9	5	72622080	0.000713
10	5	72613888	0.000149
11	5	72749056	0.000142
12	5	72749056	0.000835
13	5	72749056	0.000235
14	5	72740864	0.000234
15	5	72740864	0.000142
16	5	72740864	0.000139
17	5	72749056	0.000147
18	5	72749056	0.000227
19	5	72749056	0.000222
20	5	72749056	0.000137
21	5	72757248	0.000134
22	5	72757248	0.000227
23	5	72757248	0.000224
24	5	72757248	0.000234
25	5	72757248	0.000677
26	5	72822784	0.000309
27	5	72757248	0.001434
28	5	72749056	0.000213
29	5	72757248	0.000567
30	5	72757248	0.000302
31	5	72749056	0.000209
32	5	72757248	0.000151
33	5	72790016	0.00015
34	5	72749056	0.000145
35	5	72749056	0.000223
36	5	72876032	0.000222
37	5	72843264	0.000144
38	5	72908800	0.000143
39	5	72876032	0.000141
40	5	72876032	0.000818
41	5	72876032	0.000151
42	5	72876032	0.00029
43	5	72843264	0.000749

44	5	72867840	0.000235
45	5	72933376	0.000248
46	5	72867840	0.001454
47	5	72867840	0.000141
48	5	72859648	0.000308
49	6	72876032	0.000147
50	6	72876032	0.000148
51	6	72867840	0.000152
52	6	72876032	0.000211
53	6	72876032	0.00055
54	6	72867840	0.000137
55	6	72900608	0.000224
56	6	72859648	0.000216
57	6	72867840	0.000142
58	6	72876032	0.000138
59	6	72876032	0.000211
60	6	73003008	0.00068
61	6	73003008	0.000855
62	6	72994816	0.000147
63	6	73011200	0.000172
64	6	73003008	0.000144
65	6	73011200	0.000231
66	6	73011200	0.000225
67	6	73003008	0.000154
68	6	72970240	0.000778
69	6	72994816	0.000138
70	6	73003008	0.000222
71	6	73003008	0.000928
72	6	73003008	0.000142
73	6	72994816	0.000227
74	6	72994816	0.000131
75	6	72994816	0.000757
76	6	72986624	0.000153
77	6	73003008	0.000141
78	6	73003008	0.000294
79	6	72994816	0.000142
80	6	73027584	0.000143
81	6	72986624	0.000137
82	6	73003008	0.000148
83	6	72994816	0.000299
84	6	73003008	0.000147
85	6	73121792	0.000827
86	6	73121792	0.000151
87	6	73113600	0.000151
88	6	73105408	0.000217
89	6	73138176	0.00022
90	6	73138176	0.000151

91	6	73154560	0.000149
92	6	73121792	0.000144
93	6	73121792	0.000227
94	6	73129984	0.000233
95	6	73129984	0.000152
96	6	73195520	0.000224
97	6	73121792	0.000137
98	6	73129984	0.000142
99	6	73113600	0.00014
100	6	73105408	0.000144
101	6	73170944	0.000307
102	6	73121792	0.00014
103	6	73129984	0.000144
104	6	73129984	0.000156
105	6	73129984	0.000225
106	6	73121792	0.000234
107	6	73129984	0.000714
108	6	73138176	0.000929
109	6	73265152	0.000139
110	6	73265152	0.00016
111	6	73265152	0.000723
112	6	73265152	0.00079
113	6	73281536	0.000154
114	6	73256960	0.000228
115	6	73256960	0.000221
116	6	73248768	0.000139
117	6	73240576	0.000134
118	6	73486336	0.000145
119	6	73576448	0.000158
120	6	73576448	0.000149
121	6	73560064	0.000829
122	6	73519104	0.000171
123	6	73510912	0.000146
124	6	73510912	0.000761
125	6	73519104	0.00014
126	6	73510912	0.000217
127	6	73519104	0.000719
128	6	73519104	0.000138
129	6	73519104	0.000147
130	6	73486336	0.000142
131	6	73519104	0.000232
132	6	73494528	0.000233
133	6	73486336	0.000146
134	6	73502720	0.000149
135	6	73502720	0.000161
136	6	73502720	0.000235
137	6	73502720	0.000227

138	6	73510912	0.000149
139	6	73502720	0.000154
140	6	73478144	0.000137
141	7	73502720	0.000137
142	7	73519104	0.00014
143	7	73510912	0.000132
144	7	73510912	0.000301
145	7	73519104	0.000142
146	7	73519104	0.000151
147	7	73486336	0.000212
148	7	73502720	0.000807
149	7	73502720	0.000134
150	7	73502720	0.000135

Tabla 5: Datos utilizados en la prueba de tesis para el escenario en GNU/Linux.

Escenario tres: Cortafuegos en Sistema Integrado.

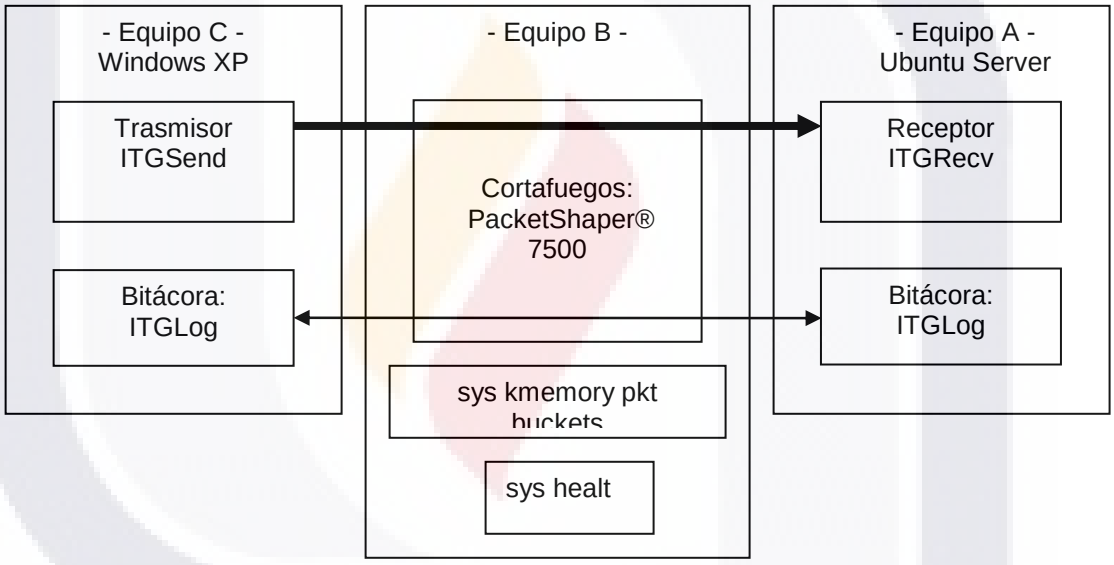


Diagrama lógico del tercer escenario: Cortafuegos en Sistema Integrado PacketShaper® 7500

1. Obtención de Datos de CPU y memoria utilizando comandos SYS:

Fue necesario iniciar dos sesiones **ssh** (*secure shell)* , una sesión se utilizó para obtener datos del CPU y la otra sesión para obtener datos de la memoria.

1. Para obtener los datos de CPU se utilizó el comando **sys healt** , este comando fue ejecutado cada dos segundos y almacenado su resultado en el artchivo CPU.log:

```
PacketShaper# sys healt >> CPU.log
```

2. Para obtener datos de la memoria se utilizó el comando **sys kmemory pkt buckets** este comando fue ejecutado cada dos segundos y almacenado su resultado en el artchivo CPU.log:

```
PacketShaper# sys kmemory pkt buckets >> MEM.log
```

3. La información obtenida fue necesaria procesarla ya que contenía una gran cantidad de información basura.

El promedio de uso de CPU siempre fue 99 % y debido que no existe documentación de *sys health*, fue necesario tomar el Vector1 ya que ese valor tuvo variación a lo largo de la prueba.

A continuación se muestra las primeras 46 líneas del archivo ***CPU.log*** que correspondes a las primeras 2 de las 150 muestras que se obtuvieron. Los datos en negritas son los datos que se necesitaron:

```
Current % Idle =  100
Average % Idle =  99 (last 25 secs)
Minimum % Idle =  0

Vector 1 (  1332,   1684,   1684)
Vector 2 (     0,      0,      0)
Vector 3 (  2190,   2208,   1314)
Load  %    =  0 (1000 ticks)
Random Ppt   =  0   0
Hard Ppt     =  0   0
Shedding Ppt =  0   0
No Buf Ppt   =  0   0
TxDone Queue =  0   0
TxPend Queue =  0   0
Rx Queue     =  0   0
TxDone Bkup  =  0   0 ticks
Rx Queue Dly =  0   0 ticks
Overhead     =  0   0 ticks

              TCP    UDP  Legacy   Total
Flows (Current):     4      2      0      6
Flows (Maximum):     4      2      3      6

Current % Idle =  100
Average % Idle =  99 (last 25 secs)
Minimum % Idle =  0

Vector 1 (  1366,   1684,   1684)
Vector 2 (    18,     12,     42)
Vector 3 (  2190,   2208,   1314)
Load  %    =  0 (1000 ticks)
Random Ppt   =  0   0
Hard Ppt     =  0   0
Shedding Ppt =  0   0
No Buf Ppt   =  0   0
TxDone Queue =  0   0
TxPend Queue =  0   0
Rx Queue     =  0   0
TxDone Bkup  =  0   0 ticks
Rx Queue Dly =  0   0 ticks
Overhead     =  0   0 ticks

              TCP    UDP  Legacy   Total
Flows (Current):     4      1      0      5
Flows (Maximum):     4      2      3      6
```

A continuación se muestra las primeras 50 líneas del archivo ***MEM.log*** que correspondes a las primeras 2 de las 150 muestras que se obtuvieron. Los datos en negritas son los datos que se necesitaron:

Packet kmalloc buffer pool bucket information:

Size	Used	Total	Limit	Allocs	Fails	Bytes	Percent
8	0	6	1048576	380	0	168	0.00
16	1	244668	524288	245860	0	8808048	65.18
32	6	14	262144	638	0	728	0.01
64	26	27	131072	161	0	2268	0.02
128	10	17487	65536	17494	0	2588076	19.15
256	0	1	32768	1	0	276	0.00
512	0	1	24576	1	0	532	0.00
1024	0	1	8192	1	0	1044	0.01
2048	0	1	4096	1	0	2068	0.02
4096	0	1	2048	1	0	4116	0.03
8192	0	1	1024	1	0	8212	0.06
16384	1	2	512	2	0	32808	0.24
32768	0	1	256	1	0	32788	0.24
65536	0	1	128	1	0	65556	0.49
131072	0	1	64	1	0	131092	0.97
262144	0	1	32	1	0	262164	1.94
524288	0	1	16	1	0	524308	3.88
1048576	0	1	8	1	0	1048596	7.76
2097144	44	262216	2105336	**264547**	0	13512848	100.00

Packet kmalloc buffer pool bucket information:

Size	Used	Total	Limit	Allocs	Fails	Bytes	Percent
8	0	6	1048576	381	0	168	0.00
16	2	244668	524288	245863	0	8808048	65.18
32	8	14	262144	640	0	728	0.01
64	26	27	131072	161	0	2268	0.02
128	10	17487	65536	17494	0	2588076	19.15
256	0	1	32768	1	0	276	0.00
512	0	1	24576	1	0	532	0.00
1024	0	1	8192	1	0	1044	0.01
2048	0	1	4096	1	0	2068	0.02
4096	0	1	2048	1	0	4116	0.03
8192	0	1	1024	1	0	8212	0.06
16384	1	2	512	2	0	32808	0.24
32768	0	1	256	1	0	32788	0.24
65536	0	1	128	1	0	65556	0.49
131072	0	1	64	1	0	131092	0.97
262144	0	1	32	1	0	262164	1.94
524288	0	1	16	1	0	524308	3.88
1048576	0	1	8	1	0	1048596	7.76
2097144	47	262216	2105336	**264553**	0	13512848	100.00

4. Fue necesario crear macros en Microsoft Office 2003 para procesar los archivos anteriores y obtener los datos del % de uso de CPU y Bits de memoria libre mostrados en negritas. A continuación se muestra el resultado de las macros para los primeros 5 muestras:

% de CPU Utilizado
1.332
1.366
1.322
1.339
1.346

KBits de memoria Utilizada:
264547
264553
264553
264553
264557

5. Los datos de CPU ya se encontraban listos para utilizarse, por el contrario los datos de memoria fue necesario procesarlos para poder cambiar los de Bits de memoria disponible a Bits de memoria utilizada, para lograr lo anterior fue necesario multiplicar por 1024 a cada uno de los datos de Kbits de memoria disponible, para automatizar este proceso se creó un script llamado ***fix_mem_integrado.sh*** en el equipo A con el siguiente contenido:

```
#! /bin/bash
while read linea
do
FIX=`echo $linea \* 1024 |bc`;
echo $FIX >> bits_$1
done < $1
```

El script anterior fue utilizado de la siguiente forma:

```
root@router:/opt/D-ITG-2.6.1d#  ./fix_mem_integrado.sh  bits_mem_integrado.txt
```

La instrucción anterior nos creo el archivo llamado: ***bits_mem_integrado.txt*** a continuación se muestran los primeros 5 renglones de dicho archivo:

```
270896128
270902272
270902272
270902272
270906368
```

2. Obtención del jitter para el flujo Multimedia VoIP con ITGDec:

1. El programa ITGLog generó un archivo llamado ***test-cada2-segundos.log***. Este archivo fue procesado con el programa ITGDec para obtener la información del jitter. De acuerdo al manual la sintaxis para obtener el jitter de manera general es la siguiente:

ITGDec [<archivo de bitácora>] [-j <intervalo en milisegundos>][-f <número de flujo>]

En nuestro caso para obtener el jitter del flujo numero 1 con periodos de 2 segundos la sintaxis utilizada fue: ***ITGDec ./test-cada2-segundos.log -f 1 -j 2000***
A continuación se muestra el comando ejecutado en la consola del equipo A:

```
root@router:/opt/D-ITG-2.6.1d#  ITGDec ./test-cada2-segundos.log -f 1 -j 2000
root@router:/opt/D-ITG-2.6.1d#  1-172.16.1.17-172.16.1.254.dat.jit
root@router:/opt/D-ITG-2.6.1d#  mv 1-172.16.1.17-172.16.1.254.dat.jit test-cada2-segundos-
```

2. El archivo generado fue ***test-cada2-segundos-JITTER.log*** , los primeros 10 datos se muestran a continuación:

```
0.000000 0.000124
2.000000 0.000201
4.000000 0.000111
6.000000 0.000129
8.000000 0.000269
10.000000 0.000113
12.000000 0.000113
14.000000 0.000115
16.000000 0.000271
18.000000 0.00083
```

3. Se puede observar que la primera columna hace referencia al tiempo en segundos de la muestra, esta columna de información no es necesaria así que fue eliminada, el resultado de los datos anteriores se muestra a continuación:

```
0.000124
0.000201
0.000111
0.000129
0.000269
0.000113
0.000113
0.000115
0.000271
0.00083
```

Los datos utilizados para la prueba de hipótesis correspondientes al tercer escenario se muestran el la siguiente tabla.

Numero de Muestra	% CPU Utilizado	Bits MEM Utilizada	Multimedia jitter en Segundos
1	1.332	270896128	0.000124
2	1.366	270902272	0.000201
3	1.322	270902272	0.000111
4	1.339	270902272	0.000129
5	1.346	270906368	0.000269
6	1.342	270906368	0.000113
7	1.347	270906368	0.000113
8	1.355	270910464	0.000115
9	1.36	270910464	0.000271
10	1.328	270911488	0.00083
11	1.353	270912512	0.000234
12	1.337	270913536	0.000277
13	1.347	270913536	0.000118
14	1.339	270919680	0.000121
15	1.342	270923776	0.000747
16	1.338	270923776	0.000193
17	1.343	270923776	0.000113
18	1.339	270923776	0.000194
19	1.346	270923776	0.000184
20	1.337	270929920	0.000116
21	1.34	270929920	0.00011
22	1.329	270930944	0.000115
23	1.336	270935040	0.000264
24	1.324	270937088	0.000687
25	1.336	270941184	0.000198
26	1.343	270941184	0.000113
27	1.337	270945280	0.000117
28	1.34	270945280	0.000111
29	1.332	270949376	0.000808
30	1.332	270950400	0.000211
31	1.342	270951424	0.000199
32	1.339	270952448	0.000111
33	1.335	270952448	0.000186
34	1.331	270958592	0.000723
35	1.35	270958592	0.000116
36	1.349	270958592	0.000191
37	1.345	270958592	0.00092
38	1.328	270962688	0.000114
39	1.337	270962688	0.000271
40	1.34	270962688	0.001291
41	1.339	270962688	0.000118
42	1.352	270966784	0.000117
43	1.339	270966784	0.000739

44	1.343	270966784	0.000121
45	1.341	270970880	0.000112
46	1.35	270970880	0.000192
47	1.332	270970880	0.000202
48	1.345	270970880	0.000769
49	1.342	270974976	0.000115
50	1.329	270976000	0.000119
51	1.33	270976000	0.000745
52	1.35	270976000	0.000192
53	1.345	270976000	0.000199
54	1.342	270976000	0.000114
55	1.338	270976000	0.000119
56	1.334	270977024	0.000115
57	1.338	270978048	0.000281
58	1.351	270979072	0.001397
59	1.347	270979072	0.000115
60	1.32	270985216	0.000123
61	1.341	270985216	0.000117
62	1.337	270989312	0.000115
63	1.339	270989312	0.000194
64	1.327	270993408	0.000191
65	1.343	270993408	0.000119
66	1.341	270993408	0.000115
67	1.34	270997504	0.000125
68	1.339	270997504	0.000194
69	1.331	270999552	0.000206
70	1.334	270999552	0.000679
71	1.337	271000576	0.000112
72	1.335	271004672	0.000116
73	1.337	271006720	0.000116
74	1.34	271010816	0.000118
75	1.337	271010816	0.000276
76	1.356	271013888	0.000782
77	1.347	271014912	0.000187
78	1.334	271016960	0.00011
79	1.341	271019008	0.000115
80	1.328	271021056	0.000115
81	1.346	271021056	0.000188
82	1.34	271022080	0.000189
83	1.348	271022080	0.000119
84	1.334	271028224	0.000202
85	1.335	271028224	0.00028
86	1.334	271029248	0.000116
87	1.342	271029248	0.000117
88	1.337	271029248	0.000117
89	1.343	271030272	0.000171
90	1.332	271034368	0.001254

91	1.334	271036416	0.00078
92	1.342	271036416	0.000198
93	1.342	271037440	0.000114
94	1.341	271037440	0.000111
95	1.322	271039488	0.000121
96	1.336	271043584	0.000273
97	1.338	271043584	0.00011
98	1.32	271043584	0.000115
99	1.34	271047680	0.000114
100	1.339	271047680	0.000134
101	1.345	271051776	0.000279
102	1.328	271055872	0.000114
103	1.333	271055872	0.000114
104	1.335	271057920	0.000114
105	1.348	271057920	0.000189
106	1.347	271058944	0.000112
107	1.341	271060992	0.000113
108	1.34	271065088	0.000274
109	1.345	271065088	0.00012
110	1.345	271065088	0.00011
111	1.334	271065088	0.000112
112	1.344	271069184	0.000197
113	1.339	271069184	0.000187
114	1.331	271073280	0.000113
115	1.341	271075328	0.00012
116	1.33	271075328	0.000116
117	1.35	271076352	0.000192
118	1.343	271076352	0.000207
119	1.338	271082496	0.0009
120	1.347	271082496	0.000745
121	1.354	271082496	0.00071
122	1.332	271082496	0.00012
123	1.329	271086592	0.000118
124	1.332	271086592	0.000833
125	1.346	271090688	0.000772
126	1.347	271091712	0.000117
127	1.343	271092736	0.000186
128	1.331	271093760	0.000116
129	1.335	271093760	0.000186
130	1.33	271099904	0.000886
131	1.337	271099904	0.000205
132	1.349	271099904	0.00012
133	1.344	271099904	0.000756
134	1.341	271104000	0.000193
135	1.34	271104000	0.000116
136	1.347	271108096	0.00011
137	1.338	271108096	0.000112

138	1.342	271108096	0.000191
139	1.335	271110144	0.000197
140	1.332	271110144	0.000114
141	1.352	271112192	0.000737
142	1.334	271112192	0.00096
143	1.342	271118336	0.000819
144	1.336	271118336	0.000115
145	1.342	271118336	0.000117
146	1.326	271118336	0.001405
147	1.341	271122432	0.000202
148	1.33	271122432	0.000761
149	1.345	271122432	0.000114
150	1.327	271123456	0.000129

Tabla 6: Datos utilizados en la prueba de tesis para el escenario en sistema integrado.

APENDICE

I. GLOSARIO DE TERMINOS.

Ancho de banda

Cantidad de datos que se pueden transmitir en una unidad de tiempo.

Anfitrión

Un sistema o dispositivo conectado a la red.

Anfitrión Dual-homed

Computadora de propósito general que tiene al menos dos interfaces de red.

Anfitrión Bastión

Es una computadora en una red que ofrece un único punto de entrada y salida a internet desde la la red interna y viceversa. Usados para mitigar los riesgos de seguridad de una red, ofreciendo un barrera entre el área pública y privada.

jitter

Variación en Latencia, es la diferencia entre el tiempo en que llega un paquete y el tiempo que se cree llegará el paquete. Si el valor es menor a 100 ms el jitter puede ser compensado de manera apropiada.

Latencia

Suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red.

Paquete

La unidad fundamental de la comunicación en Internet.

Proxy

Programa o dispositivo que realiza una acción en representación de otro. es un ordenador que intercepta las conexiones de red que un cliente hace a un servidor de destino.

Filtrado de paquetes

La acción de un dispositivo para controlar selectivamente el flujo de datos desde y hacia una red. El filtrado de paquetes permite o bloquea los paquetes, generalmente al ser ruteados de una red a otra (la mayoría de las veces de la Internet a una red interna, y viceversa). Para realizar el filtrado de paquetes, debe configurar una serie de reglas que especifican qué tipo de paquetes (por ejemplo, aquellos con origen o destino un puerto o dirección IP) se han permitido y qué tipos se van a bloquear. Filtrado de paquetes puede ocurrir en un encaminador, en un puente, de forma individual en un anfitrión. Es a veces conocido como selectivo. [1]

[1] Algunas literaturas de redes (en particular, la versión de UNIX BSD de Berkeley) utiliza el término "filtrado de paquetes" para referirse a otra cosa (A la selección de ciertos paquetes de una red para el análisis, como se hace por la *etherfind tcpdump* o programas).

Sistema Integrado

A veces traducido del inglés como *embebido*, *empotrado* o *incrustado* es un sistema informático de uso específico construido dentro de un dispositivo mayor. Los sistemas integrados se utilizan para usos muy diferentes de los usos generales para los que se emplea un ordenador personal. En un sistema integrado la mayoría de los componentes se encuentran incluidos en la placa base (*motherboard*) (la tarjeta de red, módem, etc.)

BIBLIOGRAFIA

- Andrew and Tanenbaum (2003). Redes de computadoras.
- Avolio (1999). "Firewalls and Internet security, the second hundred (Internet) years." The Internet Protocol Journal **2,2**: 24-32.
- BusinessWeek (March 2003.). "The Big Guys Latch Onto Linux". BusinessWeek.
- CERT, C. E. R. T.-. (2000b). "Exploitation of unprotected windows networking shares." CERT incident note IN-2000-02.
- CERT, C. E. R. T.-. (2000c). "911 worm." CERT incident note IN-2000-03.
- CERT, C. E. R. T.-. (2001a). "Widespread compromises via \ramen" toolkit." **CERT incident note IN-2001-01**.
- Denning (1989.). "The Internet worm." American Scientist **77**(2): 126-128.
- Free Software Foundation, I. (2009). "GNU Operating System." Retrieved 19 de Mayo, 2009, from http://www.gnu.org/home.es.html.
- Hambridge, S. a. S., J. C. (1993). "Horses and barn doors: Evolution of corporate guidelines for Internet usage." Proceedings of the USENIX Seventh System Administration Conference (LISA '93).
- Ingham, K. and S. Forrest (2002). "A History and Survey of Network Firewalls." University of New Mexico.
- Iris, R. (2008). "Sistemas de Detección de Instrusos." Retrieved 10 de Septiembre, 2008, from http://www.rediris.es/cert/doc/unixsec/node26.html#sensores-ids.
- James and Yu (2004). "Performance Evaluation of Linux Bridge." School of Computer Science, Telecommunications, and Information System (CTI)
- DePaul University.
- Kahn, A., Al-Darwish, N., Guizani, M., Menten, M., and Youssef, H. (1997). "Design and implementation of a software bridge with packet filtering and statistics collection functions." International Journal of Network Management 7, 5 (September-October): 251-263.
- Keromytis, A. D. a. W., J. L. (2000). "Transparent network security policy enforcement." FREENIX Track. 2000 USENIX Annual Technical Conference,.
- Kirch, B. O. and T. Dawson (1993). Linux Network Administrators Guide. www.faqs.org.
- Limoncelli, T. (1999). "Tricks you can do if your Firewall is a bridge." 1st Conference on Network Administration, 7-10 April 1999, Santa Clara, CA, USA.
- Liu, J. a. M., Y. (1999). "Packet filtering in bridge." Internet Workshop (WS'99),: 18-20.
- Microsoft. (2009). "Introducción a WMI." Retrieved 29 de Marzo de 2009, 2009, from http://technet.microsoft.com/es-es/library/cc787057.aspx.
- Muffett, A. (1995). "hacking with AutoHack - auditing security behind the firewall." The Fifth USENIX UNIX Security Symposium: 21-34.
- Northcutt., S. (1999). Network Intrusion Detection: An Analyst´s Handbook. .
- Ranum, M. J. (1992). "A network Firewall." Proceedings of the First World Conference on System Administration and Security.

- Rochlis, E. a. (1989). "With microscope and tweezers: An analysis of the Internet virus of November 1988." IEEE Computer Society Symposium on Security and Privacy: 326-343.
- Schneier (2000). "Secrets and Lies: Digital Security in a Networked World." ACM: 188 - 193.
- Spafford (1988). "The Internet worm program: An analysis. Tech. Rep. Purdue Technical
- Report CSD-TR-823."
- Spafford (1991). "The Internet worm incident. Tech. Rep. Purdue Technical Report
- CSD-TR-933."
- Stoll, C. (1988). "Stalking the wily hacker." ACM 31: 484-497.
- Treese, G. W. a. W., A. (1993). "X through the Firewall and other application relays." Proceedings of the USENIX Summer Conference.
- Ubuntu. (2009). "Ubuntu." Retrieved 01 de abril de 2009, 2009, from http://www.ubuntulinux.org/community/ubuntustory.
- UNAM, L. J. (2007). "Robo de identidad: la vida en manos ajenas." Retrieved 31 de marzo de 2009, 2008, from http://www.jornada.unam.mx/2007/04/17/index.php?section=economist&article=020n1eiu.
- Wiki.Ubuntu. (2008). "DraftBreezyAnnouncement." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/DraftBreezyAnnouncement.
- Wiki.Ubuntu. (2008). "DraftHoaryReleaseAnnouncement." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/DraftHoaryReleaseAnnouncement.
- Wiki.Ubuntu. (2008). "EdgyEft." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/EdgyEft.
- Wiki.Ubuntu. (2009). "DapperDrake." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/DapperDrake.
- Wiki.Ubuntu. (2009). "FeistyFawn." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/FeistyFawn.
- Wiki.Ubuntu. (2009). "GutsyGibbon." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/GutsyGibbon.
- Wiki.Ubuntu. (2009). "HardyHeron." Retrieved 1 de abril de 2009, 2009, from https://wiki.ubuntu.com/HardyHeron.
- Wikipedia.Espionaje. (2008). "Espionaje." Retrieved Miercoles 10 de Septiembre, 2008, from http://es.wikipedia.org/wiki/Espionaje.
- Zwicky, E. D., S. Cooper, et al. (2000). Building Internet Firewalls.