Introducción a la seguridad perimetral

Gunnar Eyal Wolf Iszaevich
Instituto de Investigaciones Económicas — UNAM

Seminario Admin-UNAM
30 de junio, 2005

Resumen

Los administradores de sistemas nos enfrentamos a un panorama muy difícil: Convivmos con una red hostil, insegura. No podemos saber realmente con quién realmente estamos intercambiando nuestra información, ni quién tiene acceso a nuestros recursos.

Por medio de este texto, busco explicar primero que nada por qué esta situación es así, qué proceso de diseño llevó a la creación de una red como Internet, y cómo podemos dar los primeros pasos para reducir estos riesgos.

Este material fue preparado basándome en la presentación Principios para mantener la seguridad en redes TCP/IP `http://www.gwolf.org/seguridad/princ_red_segura/` — Ni la presentación contiene todo el material acá detallado, ni este texto cubre todos los aspectos de la presentación. Sugiero consultar ambos.

Índice

1. Introducción

¿Cómo podemos mantener segura nuestra red?

La pregunta es muy simple. La respuesta, sin embargo, viene en una sucesión interminable de niveles. Y, no, no es que quiera asustar a nadie. Todos nosotros podemos hacer nuestra parte por hacer de la red un mejor lugar. Muchos de nosotros, de hecho, vivimos de ello. ¿Por dónde podemos comenzar?

Sin dudarlo ni un momento, debemos comenzar comprendiendo la tecnología misma de nuestra red. Casi todas las redes hoy en día están construídas sobre el conjunto de protocolos TCP/IP, así que todo documento introductorio como este debe comenzar explicando sus características. Vamos, pues.

1.1. Particularidades de las redes TCP/IP

Para hablar acerca de TCP/IP, comenzaremos entendiendo por qué funciona como funciona a partir de los principios utilizados desde su diseño.

1.1.1. Historia: Principios del diseño de TCP/IP

La red que hoy conocemos como Internet tiene ya casi 40 años de historia. A fines de la década de los 60, DARPA (*Defense Advanced Research Projects Agency*, del Departamento de Defensa de los Estados Unidos) financió y encargó un proyecto a varias universidades: Crear una red de comunicaciones capaz de continuar operando aún en un escenario de guerra nuclear, en la cual parte de la infraestructura fuera destruída. Y, en efecto, para 1969 había ya una red

funcional utilizando enlaces dedicados de 56K entre cuatro universidades: La Universidad de California en Los Angeles (UCLA), la Universidad de California en Santa Barbara (UCSB), la Universidad de Utah y el Instituto de Investigación Stanford (SRI).

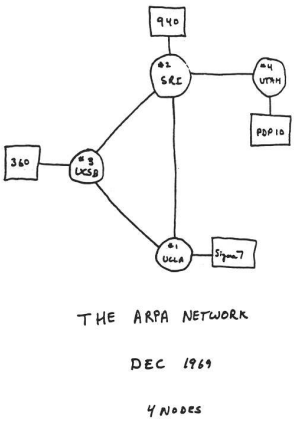


FIGURE 6.2 Drawing of 4 Node Network
(Courtesy of Alex McKenzie)

La tecnología elegida para esta tarea fue la de conmutación de paquetes[2], propuesta inicialmente por Donald Davies y Paul Baran a inicios de los 60, pues sólo con ella era posible asegurar que la interrupción de funcionamiento en un sector importante de la red no afectaría permanentemente a las conexiones previamente establecidas que utilizaban la infraestructura afecada.

1.1.2. Implicaciones actuales de los principios originales

Ahora, fuera de repetir datos ya por todos conocidos, ¿qué significa esto para nosotros?

Por un lado, el poder de cómputo y el ancho de banda han crecido de una manera increíble en casi cuatro décadas. Estos recursos eran extremadamente caros, por lo cual los protocolos de red debían hacer el uso más eficiente posible de ellos, y si bien la versión original de los protocolos básicos no es ya la que usamos hoy en día (TCP data de 1981 [3, 4]), esta visión sigue dominando el diseño de nuestras redes. Además, si bien el poder de cómputo ha crecido casi sin medida, del mismo modo ha crecido el tráfico de red — Los ruteadores siguen siendo computadoras que tienen carga constante, y cualquier requisito adicional que queramos darles debe ser muy bien planificado.

Más en lo específico a la seguridad: ¿qué provisiones de seguridad podríamos esperar que tuviera la red original?

Si las computadoras de la época eran bestias de toneladas de peso y con costos de cientos de miles de dólares, no nos cuesta entender dónde estaba enfocado el control de acceso: En el acceso físico a las instalaciones. ¿Para qué darle más trabajo al ruteador o al servidor si no existe el riesgo de que llegue un atacante con su portátil? ¿Qué mejor seguridad que la de haber sido autenticado

por un guardia al entrar al edificio?

Ha pasado, sí, mucho tiempo — Pero Internet es un ejemplo maravilloso de cómo un diseño errado o de ámbito insuficiente es prácticamente imposible de corregir más tarde. El último cambio fundamental de la infraestructura de red fue la migración de NCP a TCP/IP, entre fines de 1981 y principios de 1983 [5], de cuando había apenas algunos cientos de nodos.

1.2. Adecuando TCP/IP a la realidad actual

Si hoy conectáramos una computadora conectada al Internet de 1981, podría comunicarse con cualquiera de las computadoras de la red actual prácticamente sin problemas. Sin embargo, ha habido muy importantes adecuaciones a este juego de protocolos para permitirle crecer.

1.2.1. El tamaño de la red

Desde hace más de diez años, los oráculos nos vienen alertando acerca de que se nos está acabando el espacio en Internet. IP tiene un espacio de direccionamiento de 32 bits, lo cual nos brinda un máximo teórico de 4,294,967,296 (alrededor de una IP por cada dos seres humanos). Eso podría sonar suficiente, sin embargo, para permitir una correcta división entre diferentes tipos de organizaciones, el esquema original de Internet contemplaba el esquema de clases (direcciones de clase A, B, C, D y E). En buena parte, este esquema es el causante de que haya grandes huecos de baja densidad, así como áreas de muy alta densidad, y llevaban a un peligro real de que las direcciones se terminaran.

CIDR [6] resuelve en buena medida este problema sin romper la compatibilidad con el IP tradicional (requiere sólo un cambio menor en el software de ruteo), permitiendo asignar bloques de direcciones de longitud variable, en vez del rígido esquema de 256-65,536-16,777,216 de las clases.

Internet, a partir de mediados de los 90, dejó de ser una red entre pares, para convertirse en una red con una gran cantidad usuarios consumidores de información, que se conectan con servidores proveedores de información. Estructuralmente nada define quién es cliente y quién servidor, pero los roles se han ido estableciendo con diferencias muy claras. Como otra exitosa respuesta a la sobrepoblación de la red apareció el NAT (*Network Address Translation*) [7], que, si bien compromete uno de los principios de diseño de la red al introducir puntos únicos de fallo y convertir el esquema de una red de conmutación de paquetes a una híbrida con circuitos virtuales, ha permitido salvar el crecimiento de Internet — Detrás de cualquier dirección IP podemos tener redes tan grandes y complejas como queramos, con el beneficio adicional (desde el punto de vista de seguridad) de una barrera insalvable ante conexiones entrantes.

1.2.2. Cifrado de datos

Como ya mencionamos, la red partió en su diseño de que el tiempo de procesamiento es caro y el equipo de cómputo difícil de conseguir y de mover —

La situación hoy es diametralmente opuesta. Casi cualquier persona tiene hoy una computadora o varias, muchas veces portátiles. Las redes ya no requieren siquiera acceso físico al edificio — cada vez más organizaciones tienen redes inalámbricas[8], y si bien ya contamos con esquemas de cifrado al parecer razonablemente buenos (como WPA [9]), no es imposible que vuelva a encontrarse una importante vulnerabilidad en su diseño, como ocurrió con su antecesor, WEP [10], tras un análisis respecto a la manera en que implementa RC4[11].

Se han presentado muchas propuestas respecto a cómo implementar, hoy que es posible, cifrado a nivel de red. Esto va desde los diversos esquemas de VPN hasta comunicaciones autenticadas punto a punto o red a red, como lo especifica IPSEC [12].Estas son, sin embargo, soluciones parciales que típicamente se pueden sólo llevar a cabo bajo ambientes controlados. Mi sugerencia es no confiar en el cifrado en capa de red, e implementar cifrado en capa de aplicación siempre que nos sea posible. Lo más seguro es asumir que todo el tráfico que pone nuestra computadora en la red puede ser interceptado - Entre más sensible sea el tráfico, más capas distintas de protección debe tener.

1.2.3. ¿IPv6?

Para atacar estos problemas de una manera más limpia, a partir de 1993 se inició el trabajo sobre el protocolo que reemeplazaría la actual implementación de TCP/IP: IPv6 [13]. Con este protocolo, el espacio de direcciones de red crece de 32 a 128 bits, se incorporan características de cifrado, agrega clases de direcciones *multicast* y *anycast* para racionalizar el uso de ancho de banda, se eliminan opciones poco utilizadas de IP (especialmente de TCP) para aligerar la carga a los ruteadores, y muchas cosas más.

Ahora, según las predicciones originales, IPv4 se mostraría insuficiente alrededor de 1997, y tendríamos una migración completa de Internet para el 2000. Hoy en día, el tráfico IPv6 es una fracción despreciable del tráfico IPv4, y cada vez se ve más remoto que haya una migración en gran escala a ese nuevo estándar. Vamos a no preocuparnos por ahora por lo que el futuro puede tener reservado para nosotros, y vamos a centrarnos en mantener tan seguras y eficientes como nos sea posible nuestras actuales redes IPv4.

2. Políticas

Las políticas de una red son un documento fundamental, sin el cual el rol de un administrador de sistemas no sólo se complica innecesariamente, sino que se vuelve imposible de realizar. Increíblemente, es también un documento que muy rara vez encontraremos (y, si lo encontramos, muy frecuentemente no estará correctamente elaborado) en ninguna organización, compañía o dependencia.

Si bien este texto y este seminario es referente a la seguridad perimetral, no podemos dejar de incluir mención a las políticas. Recuerden que la mayor parte de los ataques (tanto automáticos como dirigidos) vienen de la red interna — La seguridad perimetral es un paso fundamental para asegurar nuestra red, pero

sin duda hay mucho más.

2.1. Autoridad del documento

Como primer punto: Un documento de políticas de red debe ser firmado por la persona que tenga el cargo más alto de la organización — Sólamente de esa manera tendrá suficiente autoridad para ser obligatorio para cualquiera. Si en nuestro rol de técnico, de encargado de la red o de la seguridad, le bloqueamos alguna aplicación o censuramos el contenido de un usuario con cierto rango, lo único que lograremos será tener que aceptar la excepcíon y abrir un hoyo en nuestras políticas. Y una vez abierto el hoyo, seguramente habrá más concesiones. Algo importante, claro, es recordar que todo documento obligatorio para todo usuario de la red *también nos cubre a nosotros.* No por ser técnicamente capaces de brincarnos el monitoreo o las restricciones significa que tengamos ninguna autoridad para hacerlo: Las políticas de la red aplican a cada uno de sus usuarios.

En caso de cambios en la jerarquía (por ejemplo, ante la elección de un nuevo director), una de nuestras primeras tareas que debemos cumplir es solicitarle que revise el documento ya existente y lo ratifique, o nos indique los cambios a hacer.

Ahora, no basta con que elaboremos el documento y lo firme el director — Los usuarios de nuestra red tienen que manifestarse enterados (y, en su caso, manifestar estar de acuerdo) con esta normatividad. Debemos ir ante cada uno de los usuarios y solicitar que nos firmen el documento — Sólo así podremos, legalmente, aplicar las sanciones correspondientes ante quien no lo respete. Claro está, tenemos que asesorarnos respecto a qué hacer ante la negativa de un usuario a aceptarlo, así como de los alcances —tanto dentro de la organización como respecto a la normatividad exterior— de sus implicaciones legales.

2.2. Separación de políticas y procedimientos

Las *políticas de uso aceptable* se refieren a las acciones generales, de un modo bastante abstracto. Son documentos típicamente de un par de páginas, y bastante generales. En contraposición, los *procedimientos de seguridad* de una red (por poner un ejemplo) son documentos bastante más largos y detallados, que explican a detalle cómo se va a implementar cada uno de los puntos de las políticas. Los procedimientos deben presentar una estructura similar a la de las políticas, y cada uno de sus puntos debe sevir de explicación a su contraparte en las políticas. Al ser un documento derivado, el personal operativo puede adecuar el documento de procedimientos sin pasar por todo el proceso burocrático que significaría modificar las políticas.

¿Por qué hacemos esta separación?

El uso de la tecnología cambia constantemente. En muchos casos hemos visto ejemplos de políticas perfectamente adecuadas para el momento en que son elaboradas, pero con una altísima velocidad de obsolescencia — Por ejemplo, casi todos los laboratorios de uso público con políticas elaboradas hace más de cinco años dicen que *todo disco introducido a los equipos debe ser revisado contra virus*

en determinada máquina. ¿Qué hay de malo en esto? Que el principal vector de infección hoy en día ya casi no son los documentos que viajan en un disco — El principal vector de propagación es la red (sea el correo electrónico o las carpetas compartidas). En este caso, en el documento de políticas debería indicarse que *el usuario debe tomar las precauciones necesarias para evitar la propagación de virus a través de la información que él maneje*, y los procedimientos indicarían cómo debe tratar un usuario la información que introduzca por el método que sea al centro de cómputo.

Otro ejemplo: Si en nuestras políticas mencionamos que *ningún usuario debe hacer un uso excesivo del ancho de banda*, no sólo no caeremos presos de números absolutos (por ejemplo, hubiéramos podido explicitar que no deben usar más de 500Kbps, y al crecer nuestra conexión de 10Mbps a 1Gbps, estaríamos contraviniendo las políticas si le sacáramos jugo por completo), sino que evitamos las peligrosas situaciones de una política no aplicable (no tiene en realidad sentido limitar el ancho de banda a cada dirección de esta manera) y permitimos incluso interpretar esta norma para prohibir la instalación de sistemas P2P.

2.3. Ambigüedades, aplicabilidad

Si nuestro documento contiene secciones que no puedan ser aplicadas en la realidad a nuestra red (por razones técnicas o de trabajo)[1], no podemos simplemente decretar una excepción o invalidar una sección únicamente — Al perder autoridad nuestro documento por errores de redacción. Podríamos incluso verlo como que hay ya un incentivo para quien quiera brincarse nuestras reglas: Hay ya antecedente de incorrección.

Las ambigüedades nos presentan un panorama similar, y son aún más difíciles de localizar. Si dejamos puntos sin definir ni en las políticas ni en los procedimientos, o si hay algún punto que dos indicaciones sean contradictorias, nuestro documento pierde fuerza y puede servir de pie para ataque.

3. Firewalls

El término *firewall* viene de la industria de la construcción. Se refiere al implemento que impide la propagación de incendios. Esto es normalmente un muro de tabique sólido (en un ambiente donde la generalidad de los muros son de tablaroca o de madera). Un firewall es un equipo que filtra todo el tráfico de red, permitiendo como principal función separar los paquetes autorizados de los que no tienen por qué entrar a nuestra red.

[1] En alguna ocasión, en una facultad de la UNAM se prohibió la visita a todo sitio con contenido pornográfico. La facultad en cuestión imparte la carrera de psicología — y esta norma tuvo que ser revocada, pues había un profesor que estaba haciendo un estudio, precisamente, de sitios relativos a sexualidad, erotismo y pornografía. El documento tuvo que cancelarse. Esta situación podría haberse evitado si en vez de una política específica se hubiera planteado que todo uso de la red debe ser con fines académicos — eso permitiría fácilmente prohibir el acceso a sitios de este estilo a todo quien no los requiriera para un propósito válido.

Hay quien lo ha traducido al español como *muro de fuego*, pero —siguiendo la metáfora original, muy adecuada para lo que en realidad hace un firewall— la castellanización adecuada es *cortafuegos*.

Un firewall es hoy en día ya más que un aliado para obtener un nivel aceptable de seguridad, un elemento indispensable en toda organización. Si bien la descripción que daremos a continuación puede sonar escabrosa, es uno de los elementos que menos recursos de procesamiento requerirán, y —si bien debemos mantener nuestros ojos sobre de él, mantenerlo actualizado y monitoreado— más simples de configurar de nuestra red.

3.1. Sobre qué construir el firewall

Un firewall puede montarse utilizando un sistema operativo normal de servidor (sea Unix/Linux o incluso Windows), o por medio de un equipo integrado de varios vendedores (los llamados *firewalls por hardware* o *appliances*). Los sistemas Linux y *BSD cuentan con infraestructura muy completa para montar un firewall. Los sistemas Unix comerciales típicamente se pueden montar como firewalls empleando herramientas libres como *ipfilter* o propietarias. Windows cuenta con herramientas integradas al sistema, y hay herramientas adicionales disponibles.

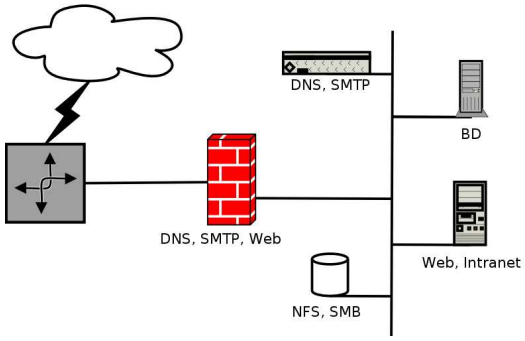
Mi recomendación va definitivamente sobre la primera opción: Linux (mediante iptables) o los sistemas *BSD (mediante pf o ipfw). El rendimiento es superior a prácticamente cualquier otra alternativa por estar la infraestructura construída dentro del mismo núcleo del sistema, al estar sus fuentes disponibles cuentan con una gran cantidad de extensiones, y tienen un alto nivel de robustez.

3.2. Configuraciones comunes

Un firewall puede utilizarse con muchas configuraciones diferentes, adecuándose a las necesidades de nuestra red. Muchas de las que aquí presentaremos pueden ser implementadas en conjunto. Algunas de las más comunes son:

3.2.1. Control de acceso a servicios sensibles

La configuración más simple que encontraremos en un firewall es básicamente un equipo con capacidad de ruteo, configurado para permitir el paso únicamente a determinados puertos, escondiendo nuestros servicios más sensibles (en el ejemplo, la Intranet, la base de datos y los sistemas de archivos compartidos por SMB o NFS) de quien esté fuera de nuestra red local

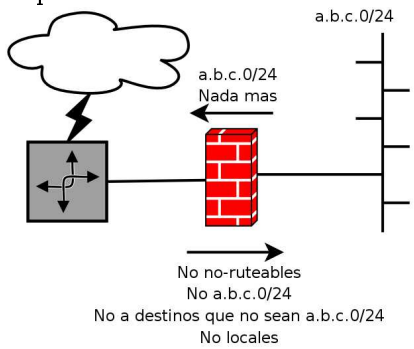


Es una configuración muy simple (cabe decir, demasiado simple), pero como sea, un primer paso — y un esquema aún muy común.

3.2.2. Ayuda para evitar *IP spoofing*

Una de las debilidades que conlleva el diseño original de TCP/IP, por el requisito de permitir a la red sobrevivir fallas masivas de parte de sus nodos, es que no podemos asegurar que un paquete proviene de la fuente que dice provenir — Un atacante puede *inyectar* paquetes en una conexión que aparenten venir de uno de los puntos válidos. Hoy en día no es ya posible exigir que la respuesta pase de vuelta por su sistema[2].

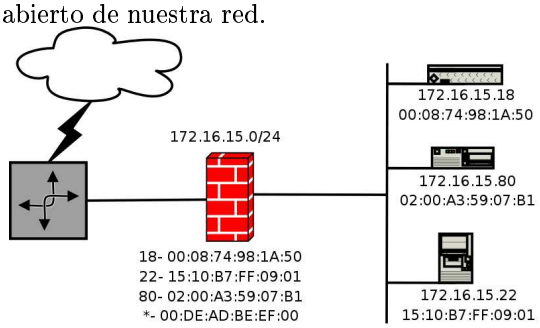
Podemos exigir a nuestro firewall que todo paquete proveniente de la interfaz de red externa que aparenten venir de una dirección interna, de una dirección no ruteable [14] (de los rangos 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12) o del rango local (127.0.0.0/8) sea rechazado. Lo que es más, si queremos brindar nuestro grano de arena de seguridad al mundo, prohibiremos la salida de cualquier paquete que no provenga de nuestra red, evitando que alguien realice ataques de este tipo desde nuestra red.



[2] Aunque esta funcionalidad sí está presente en la implementación básica de todo *stack* de TCP/IP, en toda implementación bien hecha la tendremos desactivada. Para verificar que en sus sistemas esta funcionalidad no esté activado, busquen la configuración relativa al mensaje ICMP tipo *source-route*

3.2.3. Autenticación de capa física

Todas las tarjetas de red Ethernet tienen una dirección física (la dirección MAC - Media Access Control), que es ligada a la dirección lógica que le asignemos (la dirección IP) a través del protocolo ARP (Address Resolution Protocol), que opera a nivel de enlace (capa 2). Podemos usar nuestro firewall para asegurarnos que todas las computadoras que se conecten a nuestra red hayan sido previamente identificadas por nosotros, para evitar que un atacante use un punto abierto de nuestra red.

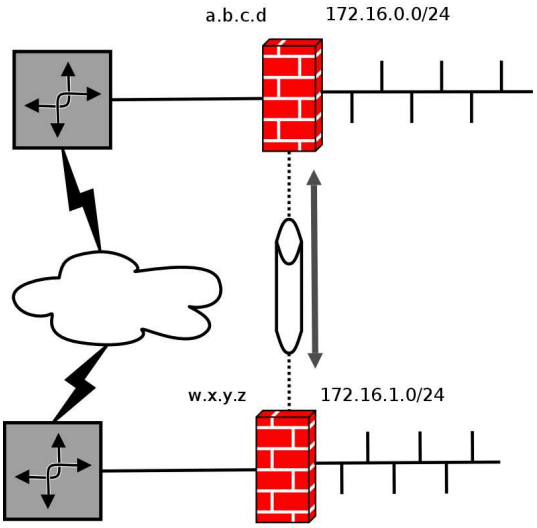


Es importante recalcar que casi todas las tarjetas Ethernet permiten la modificación de su dirección MAC. Sería un error sentirnos seguros únicamente por tener configurado nuestro firewall para validar la correspondencia de direcciones MAC a IP, pero —nuevamente— es un paso importante que dificultará la tarea de un atacante potencial.

3.2.4. Redes Privadas Virtuales (VPNs)

Hace años, cuando teníamos en nuestra organización redes separadas geográficamente, lo que hacíamos era contratar un enlace dedicado entre ambas sedes, y sólo en una de ellas manejábamos la conexión a Internet. Esto era más barato, pues el ancho de banda hacia Internet costaba mucho, y nos permitía un control absoluto del uso de nuestros recursos. Esta situación ha cambiado — Hoy en día los enlaces dedicados son mucho más caros que los enlaces a Internet.

Claro está, no es recomendable presentar todos nuestros recursos a cualquier computadora de Internet, o manejar documentos sensibles sobre una red pública. Para ello, podemos configurar una red privada virtual, para que sobre la conexión vía Internet instalemos un túnel a través del cual cifremos toda la comunicación:

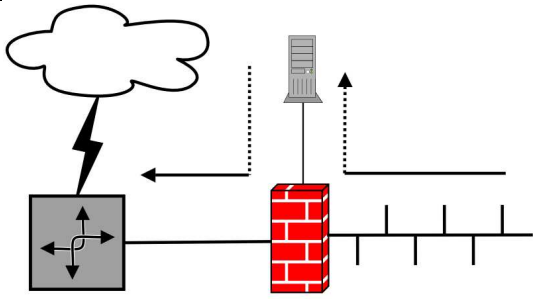


De este modo, toda la comunicación entre las dos (o más) puntas irá cifrada, y las diferentes sedes pueden incluso aparecer como dentro de la misma red local.

No olvidemos, nuevamente, que la mayor parte de los ataques viene de dentro — El tráfico dentro de nuestras redes sigue yendo en claro. Si esto no satisface nuestras necesidades de privacidad, podemos montar seguridad punto a punto, con esquemas como el de IPsec.

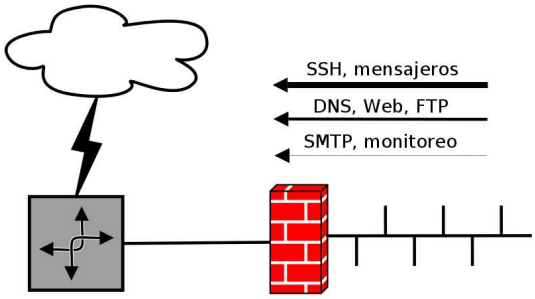
3.2.5. Proxies transparentes / autenticación

Un firewall puede *secuestrar* los paquetes que lo atraviesan, enviándolos a una computadora distinta del destino declarado. De este modo, podemos configurar desde proxies transparentes (que nos ahorrarán ancho de banda), hasta arquitecturas de autenticación al estilo Kerberos.



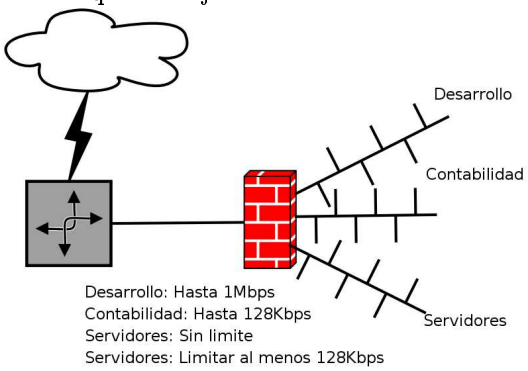
3.2.6. Calidad de servicio

Podemos indicar al firewall que, de acuerdo a diferentes criterios, coloque a los paquetes en diferentes colas, para darle prioridad a una sobre la otra. Esto nos permite, sin limitar ancho de banda, dar preferencia a los paquetes de conexiones interactivas (en las que los usuarios se dan cuenta más fácil de cualquier degradación) sobre las automáticas. Si no hay congestión, todos los paquetes pasan sin retrasos. Si hay congestión, se le da prioridad a los más importantes.



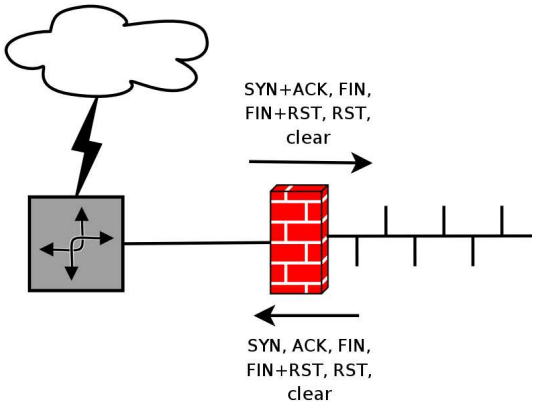
3.2.7. Control de ancho de banda

Ligado al punto anterior, podemos asignar topes de ancho de banda para las actividades de nuestros diferentes grupos de usuarios, o incluso para los distintos protocolos que manejemos.



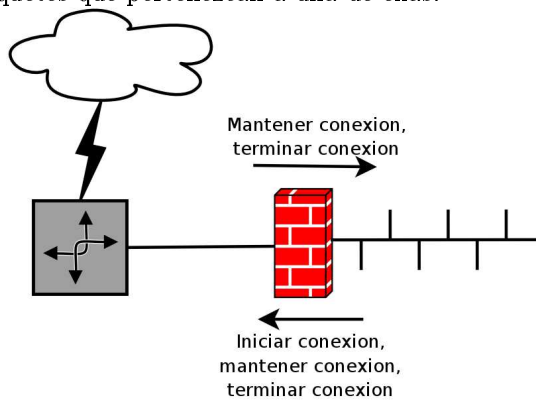
3.2.8. Banderas extrañas

Los encabezados de TCP/IP proporcionan la funcionalidad necesaria para establecer, mantener y finalizar sesiones, pero esta funcionalidad puede ser víctima de abuso — Por ejemplo, hace varios años encontraron que enviar a determinados sistemas operativos un paquete con las banderas SYN, URG y RST a la vez provocaba que se congelara. Podemos controlar las banderas de los paquetes entrantes y salientes.



3.2.9. Manejo de estado

Con el anterior esquema podemos controlar que sólo entren los paquetes legítimos que respondan a una conexión establecida - Pero un atacante podría inyectar paquetes que aparenten responder a una conexión establecida. Si le indicamos al firewall que mantenga el estado (esto es, que realice *stateful firewalling*, técnica inventada por Checkpoint que permite dar seguimiento a las conexiones como un todo, en vez de a cada paquete por separado), mantendrá una lista en memoria de las conexiones activas, y sólo permitirá el paso a paquetes que pertenezcan a una de ellas.

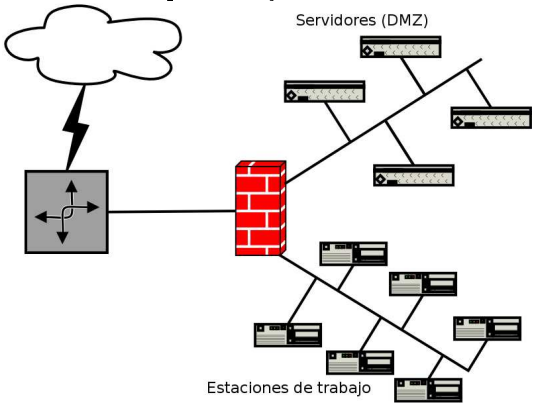


3.2.10. Zonas múltiples

Un esquema de zonas múltiples (también conocido como de zona desmilitarizada) nos permite separar nuestro segmento de servidores de nuestro segmento de trabajo. Claro está, podemos definir cuantos segmentos requiramos.

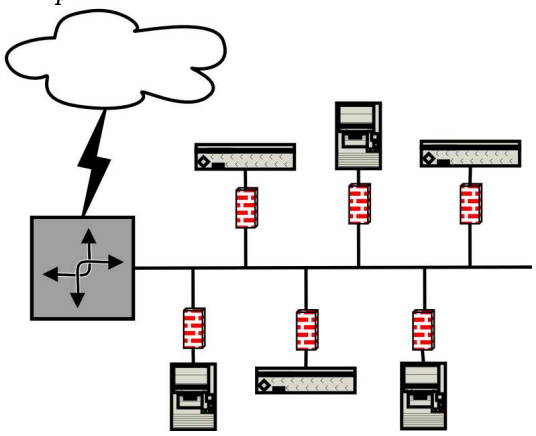
Este esquema es muy común en instituciones que mantienen sus servidores *en casa*, aminorando el impacto que un ataque externo (o un intento de abuso de privilegios de un usuario autorizado) pueda tener. Si vamos a usar este esquema,

debemos crear reglas filtrando el tráfico entre nuestras redes internas con tanto cuidado como el que manejamos con el exterior.



3.2.11. Firewalls en host

Muchas veces no tenemos control completo de la infraestructura de nuestra red. Muchas de las necesidades que nos cubre un firewall en nuestra red las podemos obtener configurando las reglas correspondientes en nuestros servidores. Esta configuración es conocida, especialmente en el mundo Windows, como *firewall personal*.



Hay muchas funciones que no nos cubre un firewall por host, pero si es nuestra única opción, es mejor que nada. El proceso de análisis de paquetes es muy simple, y no representa una gran carga para el sistema operativo de cada uno de los servidores. Además, es bueno considerar mantener siempre esta configuración en cada uno de nuestros hosts aún teniendo un firewall en el perímetro de nuestra red — Esto nos ayuda a asegurarnos que, incluso con la ausencia de parte de nuestra infraestructura de seguridad, cada uno de los servidores expone al público únicamente los componentes indispensables.

4. Sistemas de Detección de Intrusos (IDS)

Un firewall, sin embargo, dista mucho de resolver toda nuestra problemática en redes TCP/IP. Sirve únicamente como protección perimetral, pero no brinda protección alguna contra ataques internos. Puede permitir o rechazar protocolos/puertos específicos, pero no inspeccionar el contenido de los paquetes individuales — Si una conexión puede establecerse exitosamente (esto es, si va sobre un puerto autorizado), no hay mucho más que pueda hacer un firewall — El cliente/atacante está libre para intentar penetrar nuestro sistema..

El problema principal con la seguridad perimetral es que, una vez dentro, toda la red está a disposición del atacante. No estamos a salvo ni siquiera si tenemos una red con adecuado control de acceso físico — el tener una conexión explotable hacia un punto, cualquiera este sea, de la red interna nos brinda acceso completo a nuestros servicios internos.

Esta situación puede sonar desalentadora — Ahí es donde entran al juego los sistemas de detección de intrusos. Un IDS a la medida de cada uno de nuestros sistemas es, en realidad, la respuesta a todos nuestros problemas de seguridad. Sin embargo, tristemente, no existe algo tan específico — Aunque sí nos ayuda a acercarnos.

Definamos pues a un IDS de una manera menos ambiciosa: Es un conjunto de programas que monitorean el ambiente, y toman las acciones que hayamos definido ante el comportamiento que detecten como hostil.

Hay que tomar en cuenta que esta definición sigue siendo demasiado abierta — ¿Qué ambiente monitorean? ¿Cómo lo hacen? ¿Qué acciones podemos definir? Y tal vez lo que es más de cuidado, ¿qué es comportamiento hostil?

Claro está, el definir estas respuestas en términos más realistas hacen que el IDS pierda parte importante de la magia e invulnerabilidad de la primer definición.

Vamos sobre del primer punto: ¿Qué es monitorear el ambiente?

La respuesta a esta pregunta nos lleva a los dos (tres, en mi opinión) principales tipos de IDS. Yendo de lo general a lo específico:

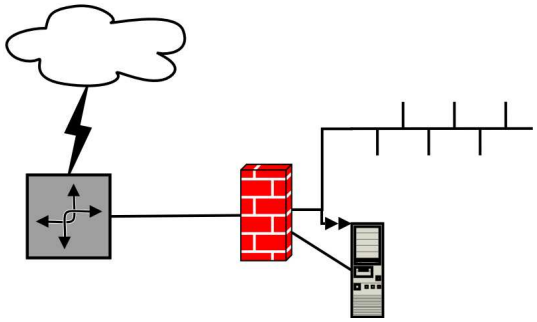
- Orientado a red (NIDS)
- Orientado a servidor (HIDS)
- Orientado a aplicación (¿AIDS?)

4.1. Orientado a red (NIDS)

Casi todo mundo, al escuchar acerca de un IDS, imagina un NIDS. Si queremos con un sólo equipo revisar los paquetes que van a toda una red, esto es lo que requeriremos:

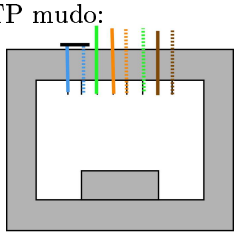
4.1.1. Levantando un NIDS

Un NIDS puede colgar simplemente de un firewall de esta manera:



Tenemos dos conexiones entre el firewall y el IDS — Una de ellas es una de sólo escucha, y la otra está completamente aislada de la red protegida. Esto significa que tenemos un canal por medio del cual el IDS está escuchando el tráfico entrante tal como lo recibe la red protegida, y para comunicarse con ella tiene que hacerlo a través del firewall.

Para crear una conexión de este tipo, un muy buen aliado casero es el cable UTP mudo:



Basta con que cortemos el par 1-2 de un cable UTP estándar para que lo convirtamos en un cable de sólo escucha. Es mejor incluso si unimos ambos polos del par cortado de ambos lados, para que el dispositivo que conectemos sienta que hay algo conectado al otro lado — Esto evitará que desactive la interfaz por no sentir un link activo.

La operación general de un NIDS es simple. Un NIDS escucha constantemente la red en modo promiscuo, en sistemas Unix casi siempre a través de la interfaz libpcap, y analiza el flujo de datos que va encontrando, en búsqueda patrones sospechosos.

4.1.2. Tipos de NIDS

Basándonos en los mecanismos que usan para clasificar el tráfico que manejan, podemos separarlos en:

Basado en firmas Compara el tráfico entrante contra patrones conocidos. Es ágil, extensible, confiable, y nos dará pocos falsos postivos (esto es, no nos responderá en falso a tráfico legítimo que no sea un ataque). ¿Su principal desventaja? Como requiere conocer la firma (una cadena característica)

de cada ataque, no reportará ataques nuevos y aún no catalogados, o variaciones triviales a ataques ya registrados.

Basado en análisis de anomalías Inicia su ciclo de actividad generando un patrón de tráfico no hostil, y una vez en producción, utiliza algún mecanismo para buscar tráfico que no concuerde con este patrón. Su funcionamiento es tan impredecible como puede ser hoy en día el campo de la inteligencia artificial — Una regla errada puede dar al traste con toda nuestra demás configuración. Además, si en la etapa de entrenamiento aparece un patrón hostil y es catalogado como tráfico normal, o no todos los tipos de tráfico normal fueron analizados, el reporte estará viciado de orígen. Como sea, esta es la única tecnología que promete protegernos aún de ataques desconocidos

4.1.3. Ventajas y debilidades de los NIDS

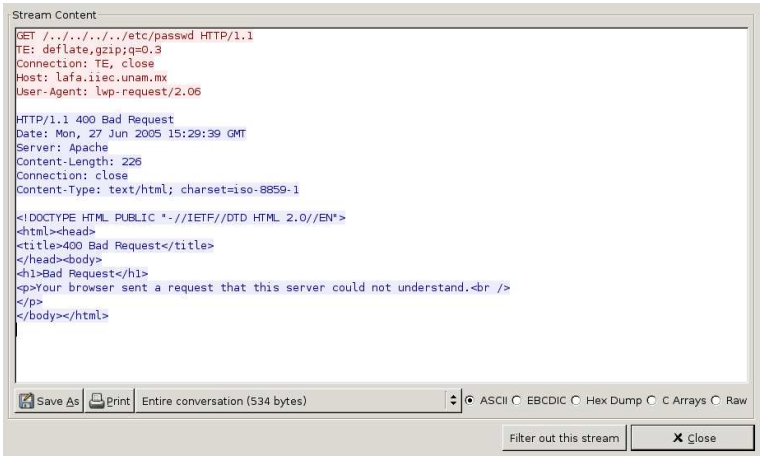
Hoy casi todas las redes usan más switches que hubs, y esto representa un gran reto para los NIDS. Por un lado, ya no tenemos un sólo punto desde donde podamos monitorear a la red entera, por lo que tenemos que planear más dónde colocamos nuestros equipos. Hay switches que presentan *puertos de cascadeo*, hacia los cuales intentan hacer llegar el tráfico completo agregado de todos sus puertos — Obviamente, si tenemos niveles normales de uso en la red, probablemente terminemos perdiendo paquetes en este puerto. En una red switcheada muchas veces tenemos que renunciar a monitorear ciertos sistemas —digamos, comunicación entre escritorios— para concentrarnos en los servidores. Recordemos sólamente que en el campo de la seguridad no existe una estrategia infalible para todo escenario. Cada red es única y nos presenta diferentes realidades.

Una gran ventaja de los NIDS es la transparencia con la cual pueden ser instalados: Los usuarios de la red —internos y externos— no van siquiera a notar su presencia. En buena parte de los casos, ni siquiera tendremos que dar de baja la red para instalarlo (o, en todo caso, bastará con un par de minutos para instalar la tarjeta de red adicional en el firewall). Más aún, esto nos da una gran ventaja sobre un posible atacante: Si no sabe de la existencia de nuestro NIDS, no intentará atacarlo para borrar las huellas de su ataque. Más aún, aún si ya se hubiera adueñado de nuestra red interna, en la configuración propuesta tendría que controlar nuestro firewall para poder atacar al NIDS.

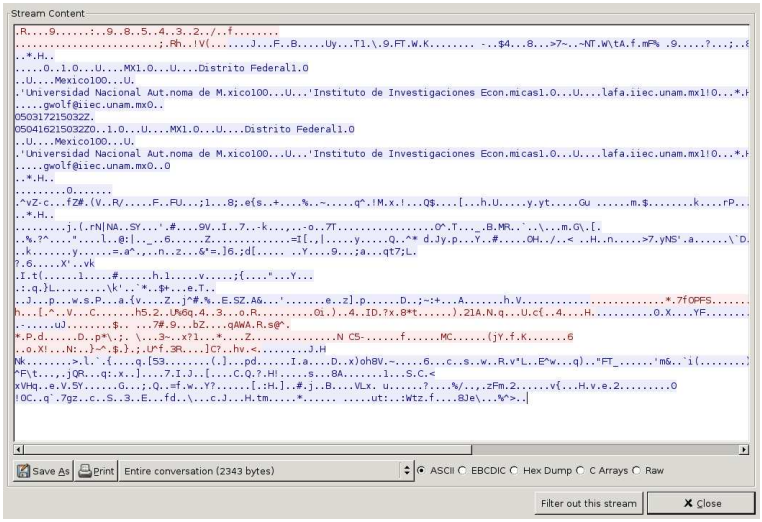
Como el análisis que realiza es en tiempo real sobre los paquetes de la red, es posible que *reaccione* en el mismo momento en que se está llevando a cabo el ataque.[3]

Ahora, un NIDS resulta inútil cuando el tráfico de red va sobre un canal cifrado. Por ejemplo, es muy fácil determinar que este patrón de tráfico hacia un servidor Web es señal de un ataque:

[3]Esto nos lleva a preguntarnos qué tan bueno es que nuestros equipos respondan de manera automática. Recordemos que existen los falsos positivos, y hay pocas cosas que molesten más a un usuario que la impredictibilidad de la red ante un patrón que casualmente fue mal detectado. Incluso si no hay lugar a dudas respecto a la naturaleza maligna del ataque, rara vez conviene responder en automático a un atacante determinado.



Sin embargo, si ese mismo ataque se lleva a cabo sobre HTTPS (HTTP sobre SSL), el tráfico que observamos es este:



El texto que aparece legible en la segunda captura no forma parte de la solicitud, es la información del certificado SSL. En ambos casos, la consulta es exactamente la misma, pero en el segundo caso un NIDS no va a reportar ninguna alarma.

Por último, un NIDS puede detectar la existencia de comportamiento hostil, pero no puede indicar si un ataque fue exitoso o no — sólamente nos reporta el hecho de que el ataque ocurrió. En el ejemplo anterior, nuestro servidor (en la primer captura) nos respondió con un mensaje de error 400 — El NIDS no sabe si esto fue por haber encontrado el intento de obtener información privilegiada y haberlo evitado, o simplemente porque el documento solicitado no existe.

4.2. Orientado a servidor (HIDS)

Como contraparte, tenemos a los sistemas de detección de intrusos orientados a servidor. Estos hacen caso omiso del contenido de los paquetes de red, enfocan su análisis a los datos del sistema: Bitácoras, registros de auditoría, estado del sistema, monitoreo del uso de recursos, etc. Veamos un par de casos.

4.2.1. Análisis de bitácoras y registros de auditoría

Los diferentes procesos del sistema van registrando los eventos que consideran relevantes a nuestras bitácoras. Como administradores de sistemas, todos conocemos bien el *mantra*: Lee tus bitácoras, revisa tus bitácoras, consulta tus bitácoras — Sin embargo, tenemos que saber qué estamos buscando. Ademas, si ponemos un humano a leer cientos y cientos de líneas buscando patrones sospechosos, terminará brincándose la información importante por considerarla como más paja.

Un HIDS que analiza bitácoras corre periódicamente y, de las diferentes bitácoras que está configurado para manejar, descarta las líneas meramente informativas, y nos presenta el material importante. Posiblemente incluso pueda tomar alguna acción basado en esta información, como avisarnos al celular requiriendo nuestra atención inmediata.

En el ejemplo que vimos en la sección de NIDS, obtenemos en la bitácora de Apache la siguiente información:

```
   [Mon Jun 20 10:29:39 2005] [error] [client 132.248.72.73] Invalid
URI in request GET /../../../../etc/passwd HTTP/1.1
   [Mon Jun 20 10:30:11 2005] [error] [client 132.248.72.73] Invalid
URI in request GET /../../../../etc/passwd HTTP/1.1
```

Esto es, lo que recibimos en una bitácora es el *efecto* de cada una de las solicitudes. No importa, en este caso, si la primera solicitud viajó en claro y la segunda cifrada — En la bitácora queda el resultado de ambas, y un HIDS puede analizarlas fácilmente.

4.2.2. Estado del sistema

Hay muchas cosas que pueden estar fuera de lugar en un sistema operativo moderno. ¿Qué puede revisar un HIDS? Entre muchas otras cosas:

- Existencia de archivos con nombres raros ('. ', '...', ' ', etc. - Nombres hechos para ser ocultados, clara evidencia de un atacante)
- Existencia de archivos nuevos con SETUID (con permisos para asumir la identidad de otro usuario — Esto es necesario para varias funciones legítimas, como abrir puertos bajos o modificar el archivo de contraseñas, pero puede también llevar a que un usuario normal tome control de nuestro sistema)
- Servicios de red inesperados

- Modificaciones en los archivos de configuración del sistema
- Entradas nuevas de cron/at
- Binarios cuyo *checksum* no corresponde al del paquete
- Nuevas cuentas
- Cuentas sin contraseña
- Cuentas que permitan la entrada por intercambio de llaves ssh
- Relaciones de confianza con otros sistemas
- Estado de interfaces de red
- Cambio en el patrón común de consumo de memoria, procesador, red y disco, incluso de horario
- Últimas entradas de los usuarios, especialmente desde hosts poco comunes

4.2.3. Ventajas y desventajas de los HIDS

El primer punto con el que nos encontramos es la gran cantidad de información a la que tenemos acceso — A veces es incluso demasiada. Un ataque relativamente simple puede causarnos una negación de servicio llenando nuestro espacio de bitácoras, especialmente si usamos registros de auditoría.

Un HIDS es casi siempre invocado a intervalos regulares, lo que hace imposible utilizarlo en tiempo real cuando ocurre un ataque. Aún así, la información que genera es de un tremendo valor para entender qué fue lo que permitió al atacante penetrar nuestro sistema, o cómo nuestro sistema logró evitarlo.

Un HIDS es siempre intrusivo — Requiere ser instalado en cada uno de los sistemas a monitorear. Puede generar mucho tráfico adicional de red, si lo configuramos para que envíe sus reportes a un punto centralizado (lo cual es una buena práctica — Si un atacante penetra uno de nuestros sistemas, muy probablemente buscará borrar sus huellas).

Un HIDS es además muy ruidoso para un atacante, con picos en el procesamiento y uso de red. Esto puede delatar su existencia a un atacante alerta.

4.3. Orientado a aplicación (¿AIDS?)

Hay un punto que no tocan ni los NIDS ni los HIDS tradicionales. Puedo asegurarles que no van a encontrar el término AIDS dentro de la literatura, en parte por ser un acrónimo nefasto, y en buena parte porque lo acabo de inventar... Pero eso no le quita su importancia: El enfatizar en que nosotros, como administradores y programadores de nuestras propias aplicaciones, debemos hacernos responsables por la detección de intrusos en las mismas. Muchas veces, la puerta de entrada a un atacante dirigido (no automático, no en masa, sino que un atacante que busca acceso privilegiado a *nuestra* información) no es una

vulnerabilidad bien conocida en la infraestructura que utilizamos, sino que en una de las aplicaciones que tenemos hechas en casa.

Estas aplicaciones típicamente son desarrolladas por nosotros mismos y tienen una mucho más estrecha gama de usuarios que la nuestra infraestructura. Además, si alguien busca dañar a nuestra organización, probablemente sea motivado por los datos que manejan nuestros sistemas. Sólamente nosotros podemos conocer el funcionamiento interno de nuestros sistemas — y por lo tanto, sólamente nosotros podemos saber qué comportamiento es anómalo o peligroso.

Sólamente nosotros podemos cuidar de nuestros sistemas.

No existe un AIDS genérico. Por nuestra protección, tenemos la *obligación* de proveer o, muy por lo menos, facilitar la implementación de la funcionalidad de un AIDS. ¿Cómo podemos hacerlo?

- Verificar siempre el resultado de nuestras operaciones
- Validar toda la entrada proveniente del agente del usuario
- Proveer un mecanismo de registro en bitácora, posiblemente independiente del que provee el marco en el cual desarrollamos (Por ejemplo, los datos críticos de una aplicación Web deben ir tanto a la bitácora del servidor como a una bitácora específica, para evitar que datos críticos se ahoguen entre paja)
- Manejar mecanismos de notificación

Y, obviamente, todo lo que —conociendo nuestra aplicación— sintamos necesario.

Referencias

[1] An Atlas of Cyberspaces `http://www.cybergeography.org/atlas/historical.html`

[2] Redes de conmutación de paquetes `http://en.wikipedia.org/wiki/Packet_switching`

[3] RFC 791, definición de IP `http://www.faqs.org/rfcs/rfc791.html`

[4] RFC 793, definición de TCP `http://www.ietf.org/rfc/rfc793.txt`

[5] RFC 801, plan de transición de NCP a TCP `http://www.faqs.org/rfcs/rfc801.html`

[6] CIDR - Classless Inter-Domain Routing `http://en.wikipedia.org/wiki/CIDR`

[7] NAT `http://en.wikipedia.org/wiki/NAT`

[8] http://en.wikipedia.org/wiki/Wifi WiFi-Redesinalámbricas

[9] WPA2: Wi-Fi Protected Access http://en.wikipedia.org/wiki/Wi-Fi_Protected_Access

[10] WEP: Wired Equivalent Privacy http://es.wikipedia.org/wiki/WEP

[11] Weaknesses in the Key Scheduling Algorithm of RC4 www.crypto.com/papers/others/rc4_ksaproc.ps, Scott Fluhrer, Itsik Mantin, Adi Shamir

[12] IPsec http://en.wikipedia.org/wiki/IPSEC

[13] IPv6 http://es.wikipedia.org/wiki/IPv6

[14] RFC 1918: Asignación de direcciones para redes privadas http://www.faqs.org/rfcs/rfc1918.html