IPsec CÓMO

Abstract
Este documento trata sobre los tareas básicas y avanzadas necesarias para establecer una red privada virtual (VPN) sobre IPsec, basándose en los núcleos Linux 2.4 y 2.5/2.6. Como ya existe una gran cantidad de documentación disponible sobre el núcleo 2.4, este documento se centrará, en una primera etapa, en las nuevas características IPsec existentes en el núcleo de desarrollo. Versiones posteriores tratarán la versión 2.4 del núcleo Linux.

Table of Contents

Introducción

Puede encontrar la última versión de este documento en The Linux Documentation Project y en la página oficial http://www.ipsec-howto.org.

Razones para escribir este documento

Durante años he leído muchos documentos Cómo. La mayoría fueron muy valiosos para mi. Cuando se implementaron las nuevas características IPsec dentro del núcleo Linux, comencé a juguetear con ellas. Pronto me di cuenta de que no existía apenas documentación. Esto me animó a escribir este documento Cómo.

Formato de este documento

El documento se divide en 5 secciones.

Sección 1: Introducción

Esta sección

Sección 2: Teoría

Teoría sobre IPsec. Básicamente, los protocolos IPsec.

Section 3: Núcleos Linux 2.2 y 2.4

Esta sección describe cómo configurar FreeS/WAN sobre los núcleos 2.2 y 2.4.

Sección 4: Núcleo Linux 2.5/2.6

Esta sección describe cómo configurar una VPN IPsec empleando las herramientas KAME **setkey** y **racoon**, el servidor IKE isakmpd de OpenBSD y FreeS/WAN mediante el parche desarrollado por Herbert Xu.

Sección 5: Configuración avanzada

Esta sección trata configuraciones avanzadas tales como DHCP-sobre-IPsec, NAT-Transversal, etc.

Colaboradores a este documento

- Fridtjof Busse
- Uwe Beck
- Juanjo Ciarlante
- Ervin Hegedus
- Barabara Kane

Información legal

Copyright

Documentation Project) u otro proyecto similar, para su difusión en Internet. Si no lo envía al LDP, deberá hacer saber al LDP dónde está disponible.
- Licencie el trabajo derivado bajo esta misma licencia o emplee la GPL. Incluya un aviso de copyright y, al menos, un enlace a la licencia empleada.
- Dé el reconocimiento debido a los autores anteriores y a los colaboradores principales.

Si considera realizar un trabajo derivado que no sea una traducción, deberá plantear sus objetivos al mantenedor actual.

Renuncia de responsabilidad

El autor no asume ninguna responsabilidad sobre cualquier acción realizada a partir de este documento, ni ofrece ninguna garantía, implícita o explícita. Si su perro muere, ¡el autor no será responsable!

Documentos relacionados

- Networking Overview HOWTO
- Networking HOWTO
- VPN-Masquerade HOWTO
- VPN HOWTO
- Advanced Routing & Traffic Control HOWTO

Next >>>
Teoría

Teoría

¿Qué es IPsec?

IPsec es una extensión al protocolo IP que proporciona seguridad a IP y a los protocolos de capas superiores. Fue desarrollado para el nuevo estándar IPv6 y después fue portado a IPv4. La arquitectura IPsec se describe en el RFC2401. Los siguientes párrafos dan una pequeña introducción a IPsec.

IPsec emplea dos protocolos diferentes - AH y ESP - para asegurarla autenticación, integridad y confidencialidad de la comunicación. Puede proteger el datagrama IP completo o sólo los protocolos de capas superiores. Estos modos se denominan, respectivamente, módo túnel y modo transporte. En modo túnel el datagrama IP se encapsula completamente dentro de un nuevo datagrama IP que emplea el protocolo IPsec. En modo transporte IPsec sólo maneja la carga del datagrama IP, insertándose la cabecera IPsec entre la cabecera IP y la cabecera del protocolo de capas suepriores (vea Figure 1).

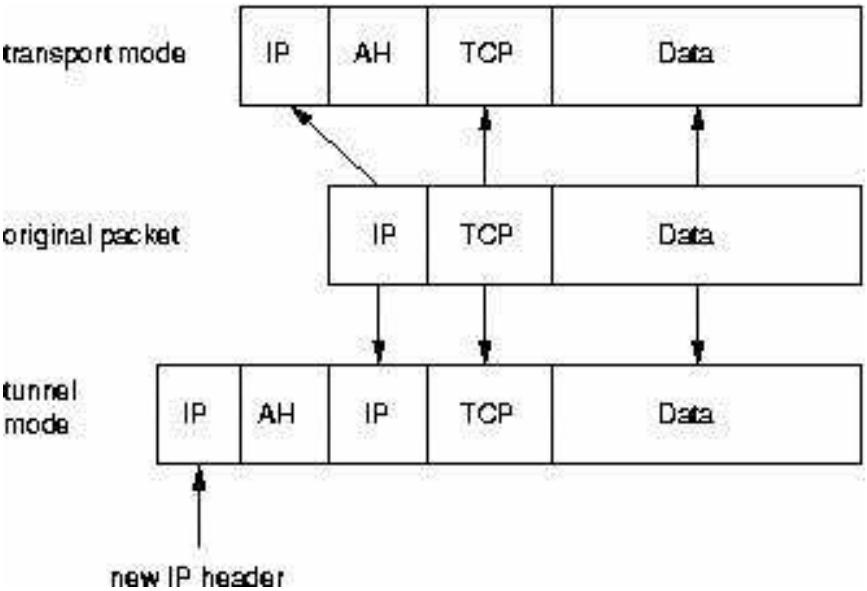

Figure 1. IPsec: modos túnel y transporte

Para proteger la integridad de los datagramas IP, los protocolos IPsec emplean códigos de autenticación de mensaje basados en resúmenes (HMAC - Hash Message Authentication Codes). Para el cálculo de estos HMAC los protocolos HMAC emplean algoritmos de resumen como MD5 y SHA para calcular un resumen basado en una clave secreta y en los contenidos del datagrama IP. El HMAC se incluye en la

cabecera del protocolo IPsec y el receptor del paquete puede comprobar el HMAC si tiene acceso a la clave secreta.

Para proteger la confidencialidad de lo datagramas IP, los protocolos IPsec emplean algoritmos estándar de cifrado simétrico. El estándar IPsec exige la implementación de NULL y DES. En la actualidad se suelen emplear algoritmos más fuertes: 3DES, AES y Blowfish.

Para protegerse contra ataques por denegación de servicio, los protocolos IPsec emplean ventanas deslizantes. Cada paquete recibe un número de secuencia y sólo se acepta su recepción si el número de paquete se encuentra dentro de la ventana o es posterior. Los paquetes anteriores son descartados inmediatamente. Esta es una medida de protección eficaz contra ataques por repetición de mensajes en los que el atacante almacena los paquetes originales y los reproduce posteriormente.

Para que los participantes de una comunicación puedan encapsular y desencapsular los paquetes IPsec, se necesitan mecanismos para almacenar las claves secretas, algoritmos y direcciones IP involucradas en la comunicación. Todos estos parámetros se almacenan en asociaciones de seguridad (SA - Security Associations). Las asociaciones de seguridad, a su vez, se almacenan en bases de datos de asociaciones de seguridad (SAD - Security Asocciation Databases).

Cada asociación de seguridad define los siguientes parámetros:

- Dirección IP origen y destino de la cabecera IPsec resultante. Estas son las direcciones IP de los participantes de la comunicación IPsec que protegen los paquetes.
- Protocolo IPsec (AH o ESP). A veces, se permite compresión (IPCOMP).
- El algoritmo y clave secreta empleados por el protocolo IPsec.
- Índice de parámetro de seguridad (SPI - Security Parameter Index). Es un número de 32 bits que identifica la asociación de seguridad.

Algunas implementaciones de la base de datos de asociaciones de seguridad permiten almacenar más parámetros:

- Modo IPsec (túnel o transporte)
- Tamaño de la ventana deslizante para protegerse de ataques por repetición.
- Tiempo de vida de una asociación de seguridad.

En una asociación de seguridad se definen las direcciones IP de origen y destino de la comunicación. Por ello, mediante una única SA sólo se puede proteger un sentido del tráfico en una comunicación IPsec full duplex. Para proteger ambos sentidos de la comunicación, IPsec necesita de dos asociaciones de seguridad unidireccionales.

Las asociaciones de seguridad sólo especifican cómo se supone que IPsec protegerá el tráfico. Para definir qué tráfico proteger, y cuándo hacerlo, se necesita información adicional. Esta información se

almacena en la política de seguridad (SP - Security Policy), que a su vez se almacena en la base de datos de políticas de seguridad (SPD - Security Policy Database).

Una política de seguridad suele especificar los siguientes parámetros:

- Direcciones de origen y destino de los paquetes por proteger. En modo transportes estas serán las mismas direcciones que en la SA. En modo túnel pueden ser distintas.
- Protocolos y puertos a proteger. Algunas implementaciones no permiten la definición de protocolos específicos a proteger. En este caso, se protege todo el tráfico entre las direcciones IP indicadas.
- La asociación de seguridad a emplear para proteger los paquetes.

La configuración manual de la asociación de seguridad es proclive a errores, y no es muy segura. Las claves secretas y algoritmos de cifrado deben compartirse entre todos los participantes de la VPN. Uno de los problemas críticos a los que se enfrenta el administrador de sistemas es el intercambio de claves: ¿cómo intercambiar claves simétricas cuando aún no se ha establecido ningún tipo de cifrado?

Para resolver este problema se desarrolló el protocolo de intercambio de claves por Internet (IKE - Internet Key Exchange Protocol). Este protocolo autentica a los participantes en una primera fase. En una segunda fase se negocian las asociaciones de seguridad y se escogen las claves secretas simétricas a través de un intercambio de claves Diffie Hellmann. El protocolo IKE se ocupa incluso de renovar periódicamente las claves para asegurar su confidencialidad.

Los protocolos IPsec

La familia de protocolos IPsec está formada por dos protocolos: el AH (Authentication Header - Cabecera de autenticación) y el ESP (Encapsulated Security Payload - Carga de seguridad encapsulada). Ambos son protocolos IP independientes. AH es el protocolo IP 51 y ESP el protocolo IP 50 (ver `/etc/protocols`). Las siguientes secciones tratarán brevemente sobre sus propiedades:

AH - Cabecera de autenticación

El protocolo AH protege la integridad del datagrama IP. Para conseguirlo, el protocolo AH calcula una HMAC basada en la clave secreta, el contenido del paquete y las partes inmutables de la cabecera IP (como son las direcciones IP). Tras esto, añade la cabecera AH al paquete. La cabecera AH se muestra en Figure 2.

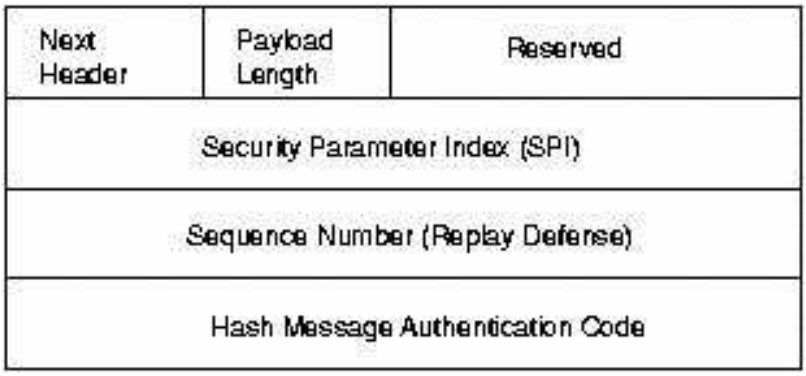

Figure 2. La cabecera AH proteje la integridad del paquete

La cabecera AH mide 24 bytes. El primer byte es el campo *Siguiente cabecera*. Este campo especifica el protocolo de la siguiente cabecera. En modo túnel se encapsula un datagrama IP completo, por lo que el valor de este campo es 4. Al encapsular un datagrama TCP en modo transporte, el valor correspondiente es 6. El siguiente byte especifica la longitud del contenido del paquete. Este campo está seguido de dos bytes reservados. Los siguientes 4 bytes especifican en *Índice de Parámetro de Seguridad* (SPI). El SPI especifica la asociación de seguridad (SA) a emplear para el desencapsulado del paquete. El *Número de Secuencia* de 32 bit protege frente a ataques por repeticón. Finalmente, los últimos 96 bit almacenan el *código de resumen para la autenticación de mensaje* (HMAC). Este HMAC protege la integridad de los paquetes ya que sólo los miembros de la comunicación que conozcan la clave secreta pueden crear y comprobar HMACs.

Como el protocolo AH protege la cabecera IP incluyendo las aprtes inmutables de la cabecera IP como las direcciones IP, el protocolo AH no permite NAT. NAT (Network address translation - Traducción de direcciones de red, también conocido como Enmascaramiento de direcciones) reemplaza una dirección IP de la cabecera IP (normalmente la IP de origen) por una dirección IP diferente. Tras el intercambio, la HMAC ya no es válida. La extensión a IPsec NAT-transversal implementa métodos que evitan esta restricción.

ESP - Carga de Seguridad Encapsulada

El protocolo ESP puede asegurar la integridad del paquete empleando una HMAC y la confidencialidad empleando cifrado. La cabecera ESP se genera y añade al paquete tras cifrarlo y calcular su HMAC. La cabecera ESP consta de dos partes y se muestra en Figure 3.

Figure 3. La cabecera ESP

Los primeros 32 bits de la cabecera ESP especifican el *Índice de Parámetros de Seguridad* (SPI). Este SPI especifica qué SA emplear para desencapsular el paquete ESP. Los siguientes 32 bits almacenan el

Número de Secuencia. Este número de secuencia se emplea para protegerse de ataques por repetición de mensajes. Los siguientes 32 bits especifican el *Vector de Inicialización* (IV - Initialization Vector) que se emplea para el proceso de cifrado. Los algoritmos de cifrado simétrico pueden ser vulnerables a ataques por análisis de frecuencias si no se emplean IVs. El IV asegura que dos cargas idénticas generan dos cargas cifradas diferentes.

IPsec emplea cifradores de bloque para el proceso de cifrado. Por ello, puede ser necesario rellenar la carga del paquete si la longitud de la carga no es un múltiplo de la longitud del paquete. En ese caso se añade la longitud del relleno (pad length). Tras la longitud del relleno se coloca el campo de 2 bytes *Siguiente cabecera* que especifica la siguiente cabecera. Por último, se añaden los 96 bit de HMAC para asegurar la integridad del paquete. Esta HMAC sólo tiene en cuenta la carga del paquete: la cabecera IP no se incluye dentro de su proceso de cálculo.

El uso de NAT, por lo tanto, no rompe el protocolo ESP. Sin enmbargo, en la mayoría de los casos, NAT aún no es compatible en combinación con IPsec. NAT-Transversal ofrece una solución para este problema encapsulando los paquetes ESP dentro de paquetes UDP.

El protocolo IKE

El protocolo IKE resuelve el problema más importante del establecimiento de comunicaciones seguras: la autenticación de los participantes y el intercambio de claves simétricas. Tras ello, crea las asociaciones de seguridad y rellena la SAD. El protocolo IKE suele implementarse a través de servidores de espacio de usuario, y no suele implementarse en el sistema operativo. El protocolo IKE emplea el puerto 500 UDP para su comunicación.

El protocolo IKE funciona en dos fases. La primera fase establece un ISAKMP SA (*Internet Security Association Key Management Security Association - Asociación de seguridad del protocolo de gestión de claves de asociaciones de seguridad en Internet*). En la segunda fase, el ISAKMP SA se emplea para negociar y establecer las SAs de IPsec.

La autenticación de los participantes en la primera fase suele basarse en claves compartidas con anterioridad (PSK - Pre-shared keys), claves RSA y certificados X.509 (**racoon** puede realizar esta autenticación incluso mediante Kerberos).

La primera fase suele soportar dos modos distintos: modo principal y modo agresivo. Ambos modos autentican al participante en la comunicación y establecen un ISAKMP SA, pero el modo agresivo sólo usa la mitad de mensajes para alcanzar su objetivo. Esto, sin embargo, tiene sus desventajas, ya que el modo agresivo no soporta la protección de identidades y, por lo tanto, es susceptible a un ataque *man-in-the-middle* (por escucha y repetición de mensajes en un nodo intermedio) si se emplea junto a claves compartidas con anterioridad (PSK). Pero sin embargo este es el único objetivo del modo agresivo, ya que los mecanismos internos del modo principal no permiten el uso de distintas claves compartidas con anterioridad con participantes desconocidos. El modo agresivo no permite la protección de identidades y

transmite la identidad del cliente en claro. Por lo tanto, los participantes de la comunicación se conocen antes de que la autenticación se lleve a cabo, y se pueden emplear distintas claves pre-compartidas con distintos comunicantes.

En la segunda fase, el protocolo IKE intercambia propuestas de asociaciones de seguridad y negocia asociaciones de seguridad basándose en la ISAKMP SA. La ISAKMP SA proporciona autenticación para protegerse de ataques *man-in-the-middle*. Esta segunda fase emplea el modo rápido.

Normalmente, dos participants de la comunicación sólo negocian una ISAKMP SA, que se emplea para negociar varias (al menos dos) IPsec SAs unidireccionales.

[<<< Previous](#) [Home](#) [Next >>>](#)
IPsec CÓMO Núcleos Linux 2.2 y 2.4 -- FreeS/WAN

Núcleos Linux 2.2 y 2.4 -- FreeS/WAN

Por hacer

Núcleo Linux 2.5/2.6 empleando herramientas KAME

Este capítulo explica el uso de la pila IPsec propia del núcleo Linux ≥2.5.47 y 2.6.*. La instalación y configuración de esta pila IPsec difiere completamente de FreeS/WAN y es similar a las variantes *BSD como FreeBSD, NetBSD y OpenBSD.

Primero se tratará la configuración e instalación del núcleo Linux y las herramientas de espacio de usuario. Después, se explicará el establecimiento de conexiones con negociación manual de claves en modo transporte y túnel. Finalmente, se tratará la configuración de conexiones con negociación automática de claves empleando claves pre-compartidas y certificados X.509. Finalmente, se explicará el soporte a *roadwarriors* (conexiones puntuales a la VPN a través de IPs cambiantes, habitualmente a través de proveedores de acceso telefónico a Internet).

Instalación

La instalación necesita de un núcleo Linux de versión 2.5.47 o superior, o 2.6.*. El código fuente del núcleo puede descargarse de http://www.kernel.org. Tras descargar el código fuente, se deberá extraer el código del paquete, configurar el núcleo y compilarlo.

```
cd /usr/local/src
tar xvjf /ruta-al-codigo-fuente/linux-<version>.tar.bz2
cd linux-<version>
make xconfig
make bzImage
make modules
make modules_install
make install
```

Estos son los comando utilizados más frecuentemente para configurar y compilar el núcleo Linux. Si necesita alguna configuración especial, consulte el Kernel-Cómo.

Es importante que, al configurar el núcleo, active las siguientes opciones:

```
Networking support (NET) [Y/n/?] y
  *
  * Networking options
  *
  PF_KEY sockets (NET_KEY) [Y/n/m/?] y
  IP: AH transformation (INET_AH) [Y/n/m/?] y
  IP: ESP transformation (INET_ESP) [Y/n/m/?] y
  IP: IPsec user configuration interface (XFRM_USER) [Y/n/m/?] y

Cryptographic API (CRYPTO) [Y/n/?] y
  HMAC support (CRYPTO_HMAC) [Y/n/?] y
  Null algorithms (CRYPTO_NULL) [Y/n/m/?] y
  MD5 digest algorithm (CRYPTO_MD5) [Y/n/m/?] y
  SHA1 digest algorithm (CRYPTO_SHA1) [Y/n/m/?] y
  DES and Triple DES EDE cipher algorithms (CRYPTO_DES) [Y/n/m/?] y
  AES cipher algorithms (CRYPTO_AES) [Y/n/m/?] y
```

Dependiendo de la versión del núcleo utilizada puede que necesite también activar el soporte de IPv6.

Una vez que el núcleo se ha compilado e instalado se deben instalar las herramientas de espacio de usuario. En la actualidad, las herramientas se mantienen en http://ipsec-tools.sourceforge.net/. Al compilar el paquete a mano, puede que necesite especificar la localización de las cabeceras del núcleo. Este paquete necesita las cabeceras del núcleo Linux versión 2.5.47 o superior.

```
./configure --with-kernel-headers=/lib/modules/2.5.47/build/include
make
make install
```

Ahora todo debería estar listo para empezar.

Conexiones con difusión manual de claves empleando setkey

Una conexión con difusión manual de claves significa que todos los parámetros necesarios para el establecimiento de la conexión son proporcionados por el administrador. El protocolo IKE no se emplea para autenticar automáticamente a los comunicantes y negociar estos parámetros. El administrador decide qué protocolo, algoritmo y clave emplear para la creación de las asociaciones de seguridad y rellena la base de datos de asociaciones de seguridad (SAD) de la manera adecuada.

Modo transporte

Esta sección tratará el establecimiento de una conexión en modo transporte con difusión manual de claves. Esta es, probablemente, la mejor manera de empezar, ya que es la conexión más simple que se puede establecer. Esta sección asume que dos máquinas con direcciones IP 192.168.1.100 y 192.168.2.100 se comunican empleando IPsec.

Todos los parámetros almacenados en las SAD y SPD pueden modificarse empleando el mandato **setkey**. Este mandato tiene una página de manual muy completa. Sólo indicaremos aquí las opciones necesarias para el establecimiento de una conexión

en modo transporte. **setkey** lee órdenes de un fichero cuando se invoca con **setkey -f /etc/ipsec.conf**. A continuación se muestra un fichero `/etc/ipsec.conf` adecuado:

```
#!/usr/sbin/setkey -f

# Configuración for 192.168.1.100

# Vaciar las SAD y SPD
flush;
spdflush;

# Atención: Emplee estas claves sólo para pruebas
# ¡Debería generar sus propias claves!

# SAs para AH empleando claves largas de 128 bits
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5 \
0xc0291ff014dccdd03874d9e8e4cdf3e6;
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5 \
0x96358c90783bbfa3d7b196ceabe0536b;

# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Políticas de seguridad
spdadd 192.168.1.100 192.168.2.100 any -P out ipsec
           esp/transport//require
           ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P in ipsec
           esp/transport//require
           ah/transport//require;
```

Si desea emplear conexiones con difusión manual de claves fuera del entorno de pruebas, necesitará claves para reemplazar las proporcionadas en este script. Emplee las siguientes órdenes para generar sus claves:

```
$ # Claves largas de 128 bits
$ dd if=/dev/random count=16 bs=1| xxd -ps
16+0 Records ein
16+0 Records aus
cd0456eff95c5529ea9e918043e19cbe

$ # Claves largas de 192 bits
$ dd if=/dev/random count=24 bs=1| xxd -ps
24+0 Records ein
24+0 Records aus
9d6c4a8275ab12fbfdcaf01f0ba9dcfb5f424c878e97f888
```

Emplee el dispositivo **/dev/random** al generar las claves ya que asegura claves aleatorias.

El script primero limpia la base de datos de asociaciones de seguridad (SAD) y la base de datos de políticas de seguridad (SPD). Después crea las SAs AH y ESP. El mandato **add** añade una asociación de seguridad a la SAD y requiere las direcciones IP de origen y destino, el protocolo IPsec (**ah**), el SPI (**0x200**) y el algoritmo. El algoritmo de autenticación se especifica con **-A** (el de cifrado con **-E**, el de compresión con **-C**; la compresión IP aún no está soportada). Tras el algoritmo se especifica la clave. Puede estar formateada en ASCII encerrado entre comillas dobles, o en hexadecimal con el prefijo **0x**.

Linux da soporte a los siguientes algortmos para AH y ESP: hmac-md5 y hmac-sha, des-cbc y 3des-cbc. En un plazo breve de tiempo, probablemente los siguientes algoritmos serán soportados: simple (sin cifrado), blowfish-cbc, aes-cbc, hmac-sha2-256 y hmac-sha2-512.

spdadd añade políticas de seguridad a la SPD. Estas políticas definen qué paquetes se protegerán con IPsec y qué protocolos y claves emplear. El mandato requiere las direcciones IP origen y destino de los paquetes a proteger, el protocolo (y puerto) a proteger (any) y la política a emplear (-P). La política especifica la dirección (in/out), la acción a aplicar (ipsec/discard/none), el protocolo (ah/esp/ipcomp), el modo (transport) y el nivel (use/require).

Este fichero de configuración debe crearse en los dos extremos que formarán parte de la comunicación IPsec. Mientras que el listado mostrado funciona sin ningún cambio en el sistema 192.168.1.100, deberá modificarse ligeramente en 192.168.2.100 para reflejar el cambio de dirección de los paquetes. La manera más sencilla de hacer esto es intercambiar las diercciones en las políticas de seguridad: reemplazar **-P in** por **-P out** y viceversa. El resultado se muestra a continuación:

```
#!/usr/sbin/setkey -f

# Configuración para 192.168.2.100

# Vaciar las SAD y SPD
flush;
spdflush;

# Atención: Emplee estas claves sólo para pruebas
# ¡Debería generar sus propias claves!

# SAs para AH empleando claves largas de 128 bits
add 192.168.1.100 192.168.2.100 ah 0x200 -A hmac-md5 \
0xc0291ff014dccdd03874d9e8e4cdf3e6;
```

```
add 192.168.2.100 192.168.1.100 ah 0x300 -A hmac-md5 \
0x96358c90783bbfa3d7b196ceabe0536b;

# SAs para ESP empleando claves largas de 192 bits (168 + 24 paridad)
add 192.168.1.100 192.168.2.100 esp 0x201 -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831;
add 192.168.2.100 192.168.1.100 esp 0x301 -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df;

# Políticas de seguridad
spdadd 192.168.1.100 192.168.2.100 any -P in ipsec
           esp/transport//require
           ah/transport//require;

spdadd 192.168.2.100 192.168.1.100 any -P out ipsec
           esp/transport//require
           ah/transport//require;
```

Una vez que el fichero de configuración esté preparado en cada uno de los extremos de la comunicación, puede cargarse empleando **setkey -f /etc/ipsec.conf**. Para comprobar el funcionamiento, puede mostrar la SAD y la SPD:

```
# setkey -D
# setkey -DP
```

La configuración ahora será similar a la de Figure 4.

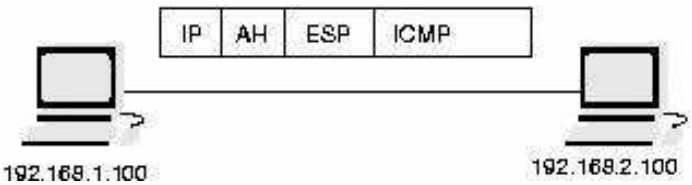

Figure 4. Dos máquinas en modo transporte empleando AH y ESP

Si hace ping de una máquina a la otra el tráfico se cifrará y tcpdump mostrará los siguientes paquetes:

```
12:45:39.373005 192.168.1.100 > 192.168.2.100: AH(spi=0x00000200,seq=0x1):
ESP(spi=0x00000201,seq=0x1) (DF)
12:45:39.448636 192.168.2.100 > 192.168.1.100: AH(spi=0x00000300,seq=0x1):
ESP(spi=0x00000301,seq=0x1)
12:45:40.542430 192.168.1.100 > 192.168.2.100: AH(spi=0x00000200,seq=0x2):
ESP(spi=0x00000201,seq=0x2) (DF)
12:45:40.569414 192.168.2.100 > 192.168.1.100: AH(spi=0x00000300,seq=0x2):
ESP(spi=0x00000301,seq=0x2)
```

Modo túnel

El modo túnel se emplea cuando los dos pares que utilizan IPsec funcionan como un gateway y protegen el tráfico entre dos redes (Figure 5). Los paquetes IP originales se cifran y encapsulan en un gateway y se transmiten al otro extremo del tunel. Allí se desencapsulan y se tratan los paquetes originales sin protección.

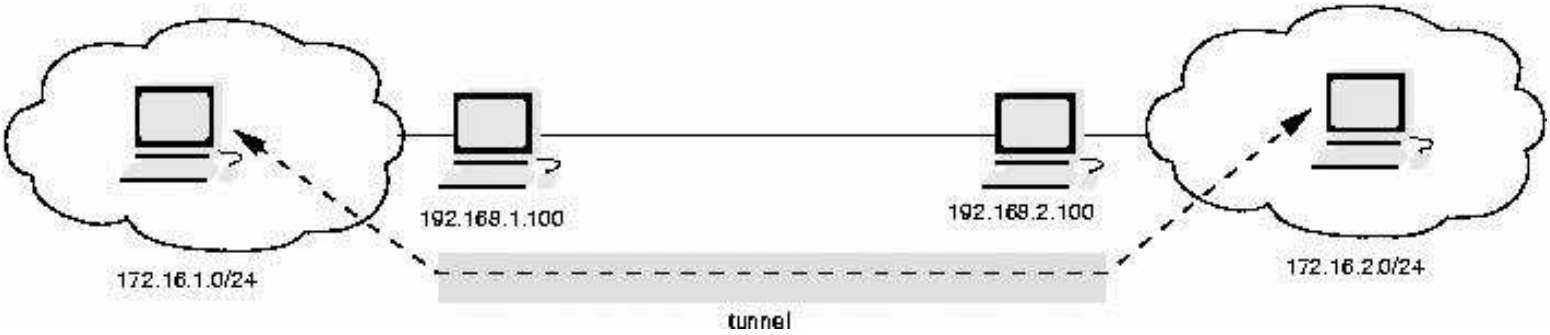

Figure 5. Los dos extremos protegen el tráfico entre dos redes

La configuración de las asociaciones de seguridad y políticas para el modo túnel es similar a la del modo transporte y se muestra en el siguiente listado.

```
#!/usr/sbin/setkey -f

# Vaciar las SAD y SPD
flush;
spdflush;

# SAs para ESP realizando cifrado con claves largas de 192 bit (168 + 24 paridad)
# y autenticación empleando claves largas de 128 bits
add 192.168.1.100 192.168.2.100 esp 0x201 -m tunnel -E 3des-cbc \
0x7aeaca3f87d060a12f4a4487d5a5c3355920fae69a96c831 \
-A hmac-md5 0xc0291ff014dccdd03874d9e8e4cdf3e6;

add 192.168.2.100 192.168.1.100 esp 0x301 -m tunnel -E 3des-cbc \
0xf6ddb555acfd9d77b03ea3843f2653255afe8eb5573965df \
-A hmac-md5 0x96358c90783bbfa3d7b196ceabe0536b;

# Políticas de seguridad
spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
           esp/tunnel/192.168.1.100-192.168.2.100/require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
           esp/tunnel/192.168.2.100-192.168.1.100/require;
```

Este ejemplo sólo emplea el protocolo ESP. El protocolo ESP asegura integridad y confidencialidad. En este caso el orden de los algoritmos ESP es importante. Primero se necesita definir el algoritmo de cifrado y su clave, y después el algoritmo de autenticación y su clave.

Al contrario que en la implementación de IPsec de BSD, una asociación de seguridad en Linux sólo puede usarse para modo transporte o túnel. El modo transporte es el modo predeterminado, por lo que cuando se desee modo túnel, la asociación de seguridad deberá definirse mediante **-m tunnel**.

Las políticas de seguridad ahora especifican las direcciones IP de las redes protegidas. Los paquetes que empleen estas direcciones IP de origen y destino se cifrarán mediante IPsec. Cuando el modo túnel se usa, la política de seguridad debe especificar *tunnel* y las direcciones IP de los pares que implementan la protección. Esta información es necesaria para localizar las IPsec SA adecuadas.

Difusión automática de claves mediante racoon

El servidor KAME IKE **racoon** también ha sido portado a Linux. Este servidor puede establecer conexiones IPsec con difusión automática de claves. Racoon permite emplear autenticación basada en claves compartidas con anterioridad, certificados X.509 y Kerberos. El servidor puede usarse en modo principal, modo agresivo y modo base en fase uno de IKE. Este capítulo describe la configuración de **racoon** en modo principal, empleando claves compartidas con anterioridad y certificados X.509 (por hacer: Kerberos). Por último lugar se expondrá el caso típico del *roadwarrior*, el comercial que se conecta desde sitios distintos empleando habitualmente conexiones a través de acceso telefónico.

Claves compartidas con anterioridad

La manera más sencilla para realizar la autenticación mediante **racoon** es emplear claves precompartidas. Estas claves deben definirse en un fichero `/etc/psk.txt`. Este fichero no debería ser leído por usuarios sin privilegios (**chmod 400 /etc/psk.txt**) y tiene la siguiente sintaxis:

```
# Direcciones IPv4
192.168.2.100          clave precompartida simple
5.0.0.1                0xe10bd52b0529b54aac97db63462850f3
# USER_FQDN
ralf@spenneberg.net    Esta es una clave precompartida para una dirección de correo
# FQDN
www.spenneberg.net     Esta es una clave precompartida
```

El fichero se organiza en columnas. La primera columna almacena la identidad del contrario autenticado por la clave precompartida. Cualquier cosa que comience en la segunda columna es la clave precompartida.

La configuración de **racoon** es muy sencilla. El siguiente listado muestra un fichero de configuración `/etc/racoon.conf` típico:

```
path pre_shared_key "/etc/psk.txt";

remote 192.168.2.100 {
        exchange_mode main;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method pre_shared_key;
                dh_group modp1024;
        }
}

sainfo address 172.16.1.0/24 any address 172.16.2.0/24 any {
        pfs_group modp768;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

Este fichero de configuración define primero el lugar donde **racoon** puede encontrar las claves precompartidas. Después define un miembro de la comunicación 192.168.2.100 y los parámetros a usar en la fase uno de la negociación IKE. El segundo párrafo especifica los parámetros que pueden emplearse para el establecimiento de asociaciones de seguridad. Esta definición puede ser específica para ciertas direcciones IP, o puede ser general si se emplea **anonymous** en lugar de las direcciones IP. En este punto se definen los algoritmos de cifrado, autenticación y compresión para la SA. Se deben definir los tres para evitar un error durante el lanzamiento de **racoon**.

El servidor IKE **racoon** no inicia la negociación del túnel de manera inmediata al iniciarse. **racoon** espera a que se necesite emplear el túnel. Para que esta notificación se lleve a cabo, el núcleo necesita saber cuándo debe realizarse. Para ello, el administrador necesita definir políticas de seguridad sin las asociaciones de seguridad apropiadas. En el momento en que el núcleo Linux necesite proteger un paquete según las políticas de seguridad definidas, y no exista ninguna asociación de seguridad disponible, el núcleo Linux llamará a **racoon** y solicitará las asociaciones de seguridad adecuadas. **Raccon** iniciará en ese momento las negociaciones IKE que concluirán con la creación de las SAs. Tras esto, el núcleo Linux podrá enviar los paquetes.

Así, para la configuración indicada se necesitarán definir las siguientes políticas en 192.168.1.100:

```
#!/usr/sbin/setkey -f
#
# Vaciar SAD y SPD
flush;
spdflush;

# Crear políticas para racoon
spdadd 172.16.1.0/24 172.16.2.0/24 any -P out ipsec
          esp/tunnel/192.168.1.100-192.168.2.100/require;

spdadd 172.16.2.0/24 172.16.1.0/24 any -P in ipsec
          esp/tunnel/192.168.2.100-192.168.1.100/require;
```

Una vez que las políticas se carguen empleando la orden **setkey-f /etc/ipsec.conf**, se podrá lanzar **racoon**. Mientras se realicen pruebas, **racoon** debería lanzarse mediante **racoon -F -c /etc/racoon.conf**. La configuración del otro extremo de la comunicación deberá modificarse para reflejar el sentido inverso. Las direcciones IP de los ficheros `/etc/psk.txt`, `/etc/ipsec.conf` y `/etc/racoon.conf` deberán intercambiarse.

Se puede seguir el proceso de inicialización del tunel a través de los ficheros de registro (*logs*) del sistema:

```
2003-02-21 18:11:17: INFO: main.c:170:main(): @(#)racoon 20001216 20001216
 sakane@kame.net
2003-02-21 18:11:17: INFO: main.c:171:main(): @(#)This product linked Open
SSL 0.9.6b [engine] 9 Jul 2001 (http://www.openssl.org/)
2003-02-21 18:11:17: INFO: isakmp.c:1365:isakmp_open(): 127.0.0.1[500] use
d as isakmp port (fd=7)
2003-02-21 18:11:17: INFO: isakmp.c:1365:isakmp_open(): 192.168.1.100[500]
 used as isakmp port (fd=9)
2003-02-21 18:11:37: INFO: isakmp.c:1689:isakmp_post_acquire(): IPsec-SA r
equest for 192.168.2.100 queued due to no phase1 found.
2003-02-21 18:11:37: INFO: isakmp.c:794:isakmp_ph1begin_i(): initiate new
phase 1 negotiation: 192.168.1.100[500]<=>192.168.2.100[500]
2003-02-21 18:11:37: INFO: isakmp.c:799:isakmp_ph1begin_i(): begin Identit
y Protection mode.
2003-02-21 18:11:37: INFO: vendorid.c:128:check_vendorid(): received Vendor
 ID: KAME/racoon
2003-02-21 18:11:37: INFO: vendorid.c:128:check_vendorid(): received Vendor
 ID: KAME/racoon
2003-02-21 18:11:38: INFO: isakmp.c:2417:log_ph1established(): ISAKMP-SA es
tablished 192.168.1.100[500]-192.168.2.100[500] spi:6a01ea039be7bac2:bd288f
f60eed54d0
2003-02-21 18:11:39: INFO: isakmp.c:938:isakmp_ph2begin_i(): initiate new p
hase 2 negotiation: 192.168.1.100[0]<=>192.168.2.100[0]
2003-02-21 18:11:39: INFO: pfkey.c:1106:pk_recvupdate(): IPsec-SA establish
ed: ESP/Tunnel 192.168.2.100->192.168.1.100 spi=68291959(0x4120d77)
2003-02-21 18:11:39: INFO: pfkey.c:1318:pk_recvadd(): IPsec-SA established:
 ESP/Tunnel 192.168.1.100->192.168.2.100 spi=223693870(0xd554c2e)
```

Certificados X.509

Racoon permite emplear certificados X.509 en el proceso de autenticación. Estos certificados pueden comprobarse contra una autoridad de certificación. La configuración es similar a la empleada con claves precompartidas y se diferencia sólo en la parte de autenticación:

```
path certificate "/etc/certs";

remote 192.168.2.100 {
        exchange_mode main;
        certificate_type x509 "mi_certificado.pem" "mi_clave_privada.pem";
        verify_cert on
        my_identifier asn1dn;
        peers_identifier asn1dn;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method rsasig;
                dh_group modp1024;
        }
}

sainfo address 172.16.1.0/24 any address 172.16.2.0/24 any {
        pfs_group modp768;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

El certificado y la clave privada se almacenan en la ruta de ciertificados `/etc/certs`. Esta ruta se establece mediante la opción **`path certificate`** dentro del fichero de configuración. Los certificados y listas de revocación de certificados se almacenan en formato PEM tal y como se generan con **`openssl`**. Para la generación de certificados lea el capítulo acerca de certificados X.509. Si el certificado del otro extremo va a comprobarse contra una autoridad de certificación (**`verify_cert on`** es la opción predeterminada), el certificado de la autoridad de certificación también deberá almacenarse en este directorio. Para que OpenSSL encuentre el certificado, éste deberá ser renombrado o enlazado empleando un nombre específico calculado a partir del propio certificado:

```
ln -s CAfile.pem `openssl x509 -noout -hash < CAfile.pem`.0
```

Si el certificado debe ser comprobado frente a una fichero de revocación de certificados (CRL), éste deberá almacenarse en el mismo directorio empleando un nombre calculado de manera similar:

```
ln -s CRLfile.pem `openssl crl -noout -hash < CAfile.pem`.r0
```

Al almacenar los certificados y la clave privada, es importante darse cuenta de que **racoon** no puede descifrar una clave privada. Por lo tanto, la clave privada deberá almacenarse en texto claro sin cifrar. Si creó una clave privada cifrada, deberá descrifrarla:

```
# openssl rsa -in my_private_key.pem -out my_private_key.pem
read RSA key
Enter PEM pass phrase: password
writing RSA key
```

Roadwarrior

Los *roadwarriors* son clientes que emplean direcciones IP dinámicas desconocidas para conectarse con la pasarela de la VPN. Estos clientes suponen dos problemas para **racoon**:

- La dirección IP no se conoce, y por lo tanto no puede especificarse en el fichero de configuración de **racoon** ni en el fichero `/etc/psk.txt`. Por este motivo se deberá emplear una manera distinta de determinar la identidad del cliente. ¡Emplear claves precompartidas requiere el uso del modo agresivo! La mejor solución es emplear certificados X.509.
- No puede crearse ninguna política de seguridad sobre la que actúe **racoon**, ya que la dirección IP de destino no se conoce. **racoon** debe crear la política de seguridad y la asociación de seguridad mientras se inicia la conexión.

Para alcanzar estos objetivos, el fichero de configuración `/etc/racoon.conf` necesita ser objeto de varias modificaciones:

```
path certificate "/etc/certs";

remote anonymous {
        exchange_mode main;
        generate_policy on;
        passive on;
        certificate_type x509 "mi_certificado.pem" "mi_clave_privada.pem";
        my_identifier asn1dn;
        peers_identifier asn1dn;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm md5;
                authentication_method rsasig;
                dh_group modp1024;
        }
}


sainfo anonymous {
        pfs_group modp1024;
        encryption_algorithm 3des;
        authentication_algorithm hmac_md5;
        compression_algorithm deflate;
}
```

La opción **generate_policy on** ordena a **racoon** crear la política apropiada cuando se inicie una nueva conexión. La opción

passive on indica a **racoon** que debe permanecer pasivo y esperar a que una nueva conexión se inicie desde fuera. **racoon** no puede iniciar una conexión.

El cambio más importante, sin embargo, es la definición de **anonymous** en las líneas **remote** y **sainfo**. Esto indica a **racoon** que acepte conexiones desde cualquier lugar.

[<<< Previous](#) [Home](#) [Next >>>](#)
Núcleos Linux 2.2 y 2.4 -- FreeS/WAN
Núcleo Linux 2.5/2.6 empleando isakmpd de OpenBSD

Núcleo Linux 2.5/2.6 empleando isakmpd de OpenBSD

Thomas Walpuski ha portado el servidor IKE del sistema operativo OpenBSED a Linux (http://bender.thinknerd.de/~thomas/IPsec/isakmpd-linux.html). **isakmpd** puede ahora emplearse con el núcleo Linux 2.5.47+ y 2.6.x para establecer VPNs con IPsec. Este capítulo describe la instalación y configuración de isakmpd.

Instalación

Si emplea una distribución basada en RPM o Debian, la instalación puede realizarse empleando los paquetes correspondientes. El autor de este documento ha realizado un paquete RPM de isakmpd para el núcleo Linux 2.6.0 (http://www.spenneberg.org/VPN/Kernel-2_6_IPsec). Este paquete puede no funcionar en las versiones anteriores del núcleo, ya que la interfaz binaria de aplicación (ABI) del núcleo ha cambiado varias veces. El proyecto Debian incluye un paquete que puede instalarse empleando **apt-get install isakmpd**.

Al instalar desde código fuente necesitará el paquete **keynote** (http://www1.cs.columbia.edu/~angelos/keynote.html) si desea emplear certificados X.509. Además, necesitará un núcleo Linux 2.5.47+ o 2.6.x.

Para conseguir el código fuente de isakmpd siga los pasos mencionados en la página web de Thomas Walpuski. Tras ello, edite el GNUmakefile de manera adecuada y active la línea **OS=linux**. Si no almacena el núcleo Linux en `/usr/src/linux`, necesitará modificar además el fichero `sysdep/linux/GNUmakefile.sysdep`.

La compilación puede realizarse empleando el mandato **make**.

isakmpd viene con dos mandatos adicionales: **keyconv** y **certpatch**. Estas herramientas están en el directorio `apps` y pueden compilarse a mano (son parte de mi paquete RPM). **Certpatch** puede añadir un SubjectAltName a un certificado existente, mientras que **keyconv** convierte de DNSSEC a openssl y viceversa.

Yo logré compilar estas herramientas mediante (su experiencia puede variar):

```
gcc -DMP_FLAVOUR=MP_FLAVOUR_GMP -I../.. -I../../sysdep/linux   -I /usr/src/linux-
2.6.0 -lcrypto -lgmp certpatch.c -o certpatch
gcc  -I../.. -I../../sysdep/linux   -I /usr/src/linux-2.6.0 -lcrypto -lgmp base64.c
keyconv.c -o keyconv
```

Un último apunte: Todas las páginas de manual están en formato Latin1. Red Hat 9 no puede mostrar estas páginas de manual. Deberá convertirlas para poder leerlas (hecho en el paquete RPM): **iconv --from-code LATIN1 --to-code UTF-8 --output isakmpd2.8 isakmpd.8**

Una vez que isakmpd se haya compilado, deberá generar la siguiente estructura de directorios obligatoria:

```
mkdir /etc/isakmpd
mkdir /etc/isakmpd/ca
mkdir /etc/isakmpd/certs
mkdir /etc/isakmpd/keynote
mkdir /etc/isakmpd/crls
mkdir /etc/isakmpd/private
mkdir /etc/isakmpd/pubkeys
```

Uso de claves precompartidas (PSK)

Isakmpd emplea un fichero de configuración y un fichero de políticas. Estos son `/etc/isakmpd/isakmpd.conf` y `/etc/isakmpd/isakmpd.policy`. La configuración emplea el conocido formato .INI. Cada sección comienza con una línea del modo:

```
[seccion]
```

Y dentro de cada sección puede asignar un valor a una etiqueta:

```
etiqueta=valor
```

Si el valor es mayor que una línea puede emplear la técnica de la barra inversa para emplear varias líneas. Los comentarios pueden ponerse en cualquier lugar empleando el símbolo de sostenido #.

Para comenzar examinaremos una configuración simple que emplea un secreto precompartido para realizar la autenticación. Eche un vistazo a Figure 5 in the Section called *Modo túnel* para la configuración.

```
[General]
Listen-on=              192.168.1.100

[Phase 1]
192.168.2.100=                ISAKMP-par-oeste

[Phase 2]
Connections=            IPsec-este-oeste

[ISAKMP-par-oeste]
Phase=                  1
Local-address=          192.168.1.100
Address=                192.168.2.100
Authentication=         EstaEsLaContraseña

[IPsec-este-oeste]
Phase=                  2
ISAKMP-peer=            ISAKMP-par-oeste
```

```
Configuration=          Modo-rapido-predeterminado
Local-ID=               Red-este
Remote-ID=              Red-oeste

[Red-oeste]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.2.0
Netmask=                255.255.255.0

[Red-este]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.1.0
Netmask=                255.255.255.0

[Modo-rapido-predeterminado]
DOI=                    IPSEC
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-MD5-PFS-SUITE
```

Este fichero de configuración describe un túnel entre las dos pasarelas 192.168.1.100 y 192.168.2.100. Este túnel puede ser usado por 172.16.1.0/24 y 172.16.2.0/24. Este fichero de configuración es el de la pasarela 192.168.1.100.

Veamos las secciones de una en una. La primera sección [General] describe la configuración general, indicando si isakmpd debería asociarse a direcciones IP específicas durante el arranque. Se recomienda si tiene varias direcciones IP en su gateway VPN.

La sección [Phase 1] describe qué configuración debe usarse para el par 192.168.2.100. Si la dirección IP del par no es conocida (roadwarrior) puede emplear **default** en su lugar.

La sección [Phase 2] describe los túneles a crear una vez que la autenticación de primera fase se establece. Si fuera posible que isakmpd no iniciara las conexiones, puede emplear **Passive-connections** en su lugar.

Ahora debe definir los nombers a los que se refiere en las secciones **Phase 1** y **Phase 2**. Primero definimos ISAKMP-par-oeste. Esta definición se emplea en **Phase 1** y sabemos la dirección local (**Local-address**) y la dirección remota (**Address**). Si no conociéramos la dirección remota, elimine esta última etiqueta. **Authentication** indica que la autenticación deberá realizarse a través de una contraseña precompartida dada en texto en claro.

Después se define el túnel IPsec-este-oeste. Se usa en la **Phase 2** y se establecerá con el par ISAKMP (**ISAKMP-peer** ISAKMP-par-oeste. Deseamos definir la configuración (**Configuration**) de la conexión y los IDs adicionales para el túnel (**Local-ID** y **Remote-ID**.

Al no estar definidos estos IDs, deberemos hacerlo. El tipo de identificador (**ID-type**) puede ser IPV4_ADDR, IPV6_ADDR, IPV4_ADDR_SUBNET y IPV6_ADDR_SUBNET.

Finalmente debemos definir la configuración del modo rápido, al que nos referimos en la descripción del túnel. Definimos el **DOI** (de manera predeterminada: IPSEC), el tipo de intercambio (**EXCHANGE_TYPE**) (de manera predeterminada: QUICK_MODE) y las **Suites** (protocolos y estándares que se emplearán). El modo del ejemplo (QM-ESP-3DES-MD5-PFS-SUITE) es el QuickMode-Encapsulated-Security-Payload-3DES-Encryption-MD5-HMAC-Perfect-Forward-Secrecy. Puede

especificar varias *suites* separadas por comas. Consulte la página de manual para obtener un listado de todos las suites y transformaciones posibles.

El fichero `isakmpd.policy` es mucho más corto. El siguiente listado muestra un ejemplo:

```
KeyNote-Version: 2
Authorizer: "POLICY"
Licensees: "passphrase:EstaEsLaContraseña"
Conditions: app_domain == "IPsec policy" &&
            esp_present == "yes" &&
            esp_enc_alg == "3des" &&
            esp_auth_alg == "hmac-md5" -> "true";
```

Para probar la conexión, inicie isakmpd empleando la siguiente línea:

```
isakmpd -d -4 -DA=90
```

Esto iniciará isakmpd en primer plano (-d) empleando IPv4 (-4) y un nivel de depuración de 90.

Una vez que la conexión se haya iniciado se podrá realizar un ping de una subred a la otra. Si también ha instalado las ipsec-tools podrá emplear el mandato **setkey** para ver las políticas y asociaciones de seguridad añadidas por isakmpd. Si detiene el isakmpd en ejecución con ctrl-c, no vaciará las SAD y SPD. Deberá hacer esto manualmente con el mandato **setkey**. Si detiene isakmpd empleando **kill -TERM** vaciará las SAD y SPD.

Empleando certificados X.509

isakmpd también puede emplear certificados X.509 para el proceso de autenticación. Puede crear sus certificados empleando las herramientas habituales. Necesitará para cada máquina que vaya a formar parte de la VPN los siguientes ficheros:

- `/etc/isakmpd/private/local.key` La clave privada de la máquina en formato .pem. Sus permisos deben ser 600.
- `/etc/isakmpd/ca/ca.crt` El certificado de la autoridad de certificación en que confíe.
- `/etc/isakmpd/certs/ip-address.crt` El certificado de la máquina local.

Para que isakmpd encuentre y utilice el certificado, éste debe incluir un SubjectAltName. Esta extensión X.509v3 puede definirse durante la generación del certificado o con posterioridad empleando la orden **certpatch**. Esta orden necesita la clave privada de la CA, extrae el certificado, añade la extensión, y vuelve a firmar el certificado.

```
certpatch -i ip-address -k ca.key  originalcert.crt newcert.crt
```

Certpatch puede añadir una dirección IP, un FQDN o un UFQDN al certificado.

Una vez que estos ficheros se almacenen en las carpetas adecuadas y tengan asignados los permisos adecuados, puede crear el fichero de configuración y el fichero de políticas. En el fichero de configuración deberá eliminar la línea Authentication y

añadir una línea **ID=Este** a la sección ISAKMP-par-oeste. Después deberá definir Este. Además, deberá especificar los directorios X.509. A continuación se muestra el fichero de configuración completo:

```
[General]
Listen-on=              192.168.1.100

[Phase 1]
192.168.2.100=                ISAKMP-par-oeste

[Phase 2]
Connections=            IPsec-este-oeste

[ISAKMP-par-oeste]
Phase=                  1
Local-address=          192.168.1.100
Address=                192.168.2.100
ID=                     Este

[Este]
ID-type=                IPV4_ADDR
Address=                192.168.1.100

[IPsec-este-oeste]
Phase=                  2
ISAKMP-peer=            ISAKMP-par-oeste
Configuration=          Modo-rapido-predeterminado
Local-ID=               Red-este
Remote-ID=              Red-oeste

[Red-oeste]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16.1.0
Netmask=                255.255.255.0

[Red-este]
ID-type=                IPV4_ADDR_SUBNET
Network=                172.16..2.0
Netmask=                255.255.255.0

[Modo-rapido-predeterminado]
DOI=                    IPSEC
EXCHANGE_TYPE=          QUICK_MODE
Suites=                 QM-ESP-3DES-MD5-PFS-
SUITE
[X509-certificates]
CA-directory=           /etc/isakmpd/ca/
Cert-directory=         /etc/isakmpd/certs/
Private-key=            /etc/isakmpd/private/local.key
```

El fichero de políticas también necesita modificarse. Como sólo se desea permitir la conexión de pares que utilicen certificados firmados por la autoridad de certificación en que se confía, debe añadir una la línea tras la línea Authorizer. A continuación se muestra el fichero completo:

```
KeyNote-Version: 2
Authorizer: "POLICY"
Licensees: "DN:/C=DE/ST=NRW/L=Steinfurt/O=Spenneberg.Com/OU=VPN/CN=RootCA"
Conditions: app_domain == "IPsec policy" &&
            esp_present == "yes" &&
            esp_enc_alg == "3des" &&
            esp_auth_alg == "hmac-md5" -> "true";
```

El texto que sigue a **DN:** debe coincidir con la línea **subject** del certificado de la CA:

```
openssl x509 -in ca/ca.crt -noout -subject
```

Ahora puede iniciar isakmpd de la manera normal para probar la configuración.

Uso de FreeS/WAN en el núcleo Linux 2.6

Por hacer (la semana que viene ;-)

<<< Previous
Núcleo Linux 2.5/2.6 empleando herramientas KAME

Home

Next >>>
Generación de certificados X.509

Generación de certificados X.509

Hoy en día, casi todas las implementaciones de VPN permiten el uso de certificados X.509 para la autenticación de los participantes. Estos son los mismos certificados que se emplean para la implementación de SSL en el protocolo HTTP.

Este capítulo mostrará brevemente la creación de estos certificados.

Uso de OpenSSL

La manera más sencilla de crear certificados X.509 en Linux es mediante el mandato **openssl** y sus herramientas auxiliares. Al instalar el paquete OpenSSL, también suele instalarse el mandato auxiliar **CA** y/o **CA.pl**. Emplearemos este mandato para crear los certificados.

Primero, deberá comprobar dónde está instalado el mandato. Suele instalarse fuera de la ruta de ejecución automática ($PATH). En la distribución Red Hat se instala en `/usr/share/ssl/misc/CA`.

Ahora, cree primero su aotridad de certificación.

```
$ mkdir certs
$ cd certs
$ /usr/share/ssl/misc/CA -newca
CA certificate filename (or enter to create) <enter>

Making CA certificate ...
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
................++++++
.............++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase: contraseñaCA
Verifying password - Enter PEM pass phrase: contraseñaCA
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [NRW]:
Locality Name (eg, city) [Steinfurt]:
Organization Name (eg, company) [Spenneberg.com]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:RootCA 2004
Email Address []:ralf@spenneberg.net
```

Introduzca los valores apropiados cuando se le pregunte por el nombre del país, etc. Si desea que se propongan, de manera predeterminada, los valores correctos (como en mi caso) edite su fichero `openssl.cnf`. En Red Hat Linux suele encontrarse en `/usr/share/ssl/openssl.cnf`.

El certificado generado sólo es válido durante un año. A menudo se desea obtener certificados de CA con un tiempo de vida mayor. Como los certificados que se firman después suelen tener un tiempo de vida más corto, no suele ser práctico editar el fichero `openssl.cnf`. Es mejor cambiar el tiempo de vida de manera manual:

```
$ cd demoCA/
$ openssl x509 -in cacert.pem -days 3650 -out cacert.pem
-signkey ./private/cakey.pem
Getting Private key
Enter PEM pass phrase: capassword
$ cd ..
```

La autoridad de certificación ya está lista. Ahora crearemos las solicitud de firma para el certificado:

```
$ /usr/share/ssl/misc/CA -newreq
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
..............................++++++
.....................................++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase: contraseñacert
Verifying password - Enter PEM pass phrase: contraseñacert
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [NRW]:
Locality Name (eg, city) [Steinfurt]:
Organization Name (eg, company) [Spenneberg.com]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:VPN-Gateway
Email Address []:ralf@spenneberg.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
```

El fichero `newreq.pem` contiene la solicitud de firma del certificado y la clave privada cifrada. Este fichero puede emplearse después como clave privada para FreeS/WAN o Racoon. Una vez que la petición se crea, podemos emplear la autoridad de certificación para firmarla.

```
$ /usr/share/ssl/misc/CA -sign
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase: capassword
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'NRW'
localityName          :PRINTABLE:'Steinfurt'
organizationName      :PRINTABLE:'Spenneberg.com'
commonName            :PRINTABLE:'VPN-Gateway'
emailAddress          :IA5STRING:'ralf@spenneberg.net'
Certificate is to be certified until Apr 29 06:08:56 2004 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Dependiendo de la versión del mandato **CA** el certificado se imprimirá por salida estándar. Deberá ser similar al siguiente certificado:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=DE, ST=NRW, L=Steinfurt, O=Spenneberg.com,
CN=RootCA 2003/Email=ralf@spenneberg.net
        Validity
            Not Before: Apr 30 06:08:56 2003 GMT
            Not After : Apr 29 06:08:56 2004 GMT
        Subject: C=DE, ST=NRW, L=Steinfurt, O=Spenneberg.com,
CN=VPN-Gateway/Email=ralf@spenneberg.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:c5:3b:9c:36:3a:19:6c:a9:f2:ba:e9:d2:ed:84:
                    33:36:48:07:b2:a3:2d:59:92:b0:86:4c:81:2c:ea:
                    5c:ed:f3:ba:eb:17:4e:b3:3a:cc:b7:5b:5d:ca:b3:
                    04:ed:fb:59:3c:c5:25:3e:f3:ff:b0:22:10:fb:de:
                    72:0a:ee:42:4b:9a:d3:27:d3:b6:fb:e9:88:10:c8:
                    47:b7:26:4f:71:40:e4:75:c4:c0:ee:6b:87:b8:6f:
                    c9:5e:66:cf:bb:e7:ad:72:68:b8:6d:fd:8f:4c:1f:
                    3a:a2:0d:43:25:06:b9:92:e7:20:6c:86:15:a0:eb:
                    7f:f7:0b:9a:99:5d:14:88:9b
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                CB:5C:19:9B:E6:8A:8A:FE:0E:C4:FD:5E:DF:F7:BF:3D:A8:
18:7C:08
            X509v3 Authority Key Identifier:
                keyid:01:BB:C6:33:BE:F5:9A:5E:B0:0C:5D:BD:41:E9:78:
6C:54:AD:66:8E
                DirName:/C=DE/ST=NRW/L=Steinfurt/O=Spenneberg.com/
CN=RootCA 2003/Email=ralf@spenneberg.net
                serial:00

    Signature Algorithm: md5WithRSAEncryption
        6f:89:2b:95:af:f1:8d:4d:b7:df:e8:6d:f7:92:fb:48:8c:c4:
        1a:43:68:65:97:01:87:a6:84:b5:a1:38:bd:62:74:70:db:9e:
        78:19:d9:0c:af:18:ad:13:77:56:7d:3f:19:61:da:ba:74:30:
        8e:c5:50:0e:e3:eb:ff:95:cd:8d:d6:7e:c3:0e:ab:5b:34:94:
```

```
        bc:16:0f:ef:dc:de:40:bb:7d:ba:a2:b8:5d:f9:74:e7:28:58:
        75:a0:66:d2:8d:85:ba:38:82:08:10:33:ef:be:29:c9:31:9d:
        63:a9:f7:e0:99:ea:a7:ed:b6:b5:33:1b:1c:4a:a4:05:40:6e:
        40:7b
-----BEGIN CERTIFICATE-----
MIIDjDCCAvWgAwIBAgIBATANBgkqhkiG9w0BAQQFADCBgjELMAkGA1UEBhMCREUx
DDAKBgNVBAgTA05SVzESMBAGA1UEBxMJU3RlaW5mdXJ0MRcwFQYDVQQKEw5TcGVu
bmViZXJnLmNvbTEUMBIGA1UEAxMLUm9vdENBIDIwMDMxIjAgBgkqhkiG9w0BCQEW
E3JhbGZAc3Blbm5lYmVyZy5uZXQwHhcNMDMwNDMwMDYwODU2WhcNMDQwNDI5MDYw
ODU2WjCBgjELMAkGA1UEBhMCREUxDDAKBgNVBAgTA05SVzESMBAGA1UEBxMJU3Rl
aW5mdXJ0MRcwFQYDVQQKEw5TcGVubmViZXJnLmNvbTEUMBIGA1UEAxMLVlBOLUdh
dGV3YXkxIjAgBgkqhkiG9w0BCQEWE3JhbGZAc3Blbm5lYmVyZy5uZXQwgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAMU7nDY6GWyp8rrp0u2EMzZIB7KjLVmSsIZM
gSzqXO3zuusXTrM6zLdbXcqzBO37WTzFJT7z/7AiEPvecgruQkua0yfTtvvpiBDI
R7cmT3FA5HXEwO5rh7hvyV5mz7vnrXJouG39j0wfOqINQyUGuZLnIGyGFaDrf/cL
mpldFIibAgMBAAGjggEOMIIBCjAJBgNVHRMEAjAAMCwGCWCGSAGG+EIBDQQfFh1P
cGVuU1NMIEdlbmVyYXRlZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUy1wZm+aKiv4O
xP1e3/e/PagYfAgwga8GA1UdIwSBpzCBpIAUAbvGM771ml6wDF29Qel4bFStZo6h
gYikgYUwgYIxCzAJBgNVBAYTAkRFMQwwCgYDVQQIEwNOUlcxEjAQBgNVBAcTCVN0
ZWluZnVydDEXMBUGA1UEChMOU3Blbm5lYmVyZy5jb20xFDASBgNVBAMTC1Jvb3RD
QSAyMDAzMSIwIAYJKoZIhvcNAQkBFhNyYWxmQHNwZW5uZWJlcmcubmV0ggEAMA0G
CSqGSIb3DQEBBAUAA4GBAG+JK5Wv8Y1Nt9/obfeS+0iMxBpDaGWXAYemhLWhOL1i
dHDbnngZ2QyvGK0Td1Z9Pxlh2rp0MI7FUA7j6/+VzY3WfsMOq1s0lLwWD+/c3kC7
fbqiuF35dOcoWHWgZtKNhbo4gggQM+++KckxnWOp9+CZ6qfttrUzGxxKpAVAbkB7
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
```

Ahora se recomienda renombrar los ficheros `newreq.pem` y `newcert.pem` a algo más representativo.

```
$ mv newcert.pem vpngateway_cert.pem
$ mv newreq.pem vpngateway_key.pem
```

Tras esto, podemos divertirnos creando certificados para cada participante en la VPN.

En caso de que una clave privada sea robada o resulte comprometida, deberá revocarla ya que, según su período de vida aún es válida. Las claves revocadas se almacenan en una lista de revocación de certificados (CRL). Primero, cree una lista (vacía):

```
$ openssl ca -gencrl -out crl.pem
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase: contraseña-ca
```

Para revocar un certificado necesitará tener el fichero del certificado. Éste también se almacena en `demoCA/newcerts/`. El nombre del certificado puede obtenerse de `demoCA/index.txt`. Después, emplee la siguiente orden.

```
$ openssl ca -revoke compromised_cert.pem
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase: contraseña-ca
Revoking Certificate 01.
Data Base Updated
```

Una vez que el certificado se ha revocado, la lista de revocación debe regenerarse empleando la orden de más arriba.

[<<< Previous](#) [Home](#)
Núcleo Linux 2.5/2.6 empleando isakmpd de OpenBSD