

Manual de tinc

Preparando una Red Privada Virtual con tinc

Ivo Timmermans <itimmermans@bigfoot.com>

Introducción

Tinc es un demonio de Red Privada Virtual (VPN) que usa túnel y cifrado de datos para crear una red privada segura entre hosts en Internet.

El túnel se realiza en la capa del protocolo IP como un dispositivo de red normal, por lo cual hay necesidad de adaptar el software existente.

Este túnel permite que sitios VPN compartan información entre ellos en Internet sin exponer esta información a otros.

Este documento es el manual de tinc. Incluye capítulos sobre cómo configurar su computadora para usar tinc, así como el proceso de configuración de tinc.

Redes Privadas Virtuales

Una Red Privada Virtual o VPN es una red que sólo puede ser accedida por computadoras elegidas para participar. Esta meta es alcanzable en más de una manera.

Por ejemplo, una VPN puede consistir en una ethernet LAN. O incluso dos computadoras conectadas usando un cable módem nulo(1). En estos casos, es obvio que la red es *privada*, nadie puede acceder a esta desde afuera. Pero si estas si estas computadoras están conectadas a internet, la red deja de ser privada, a menos que use cortafuego para bloquear el tráfico privado. Pero entonces, no hay manera de enviar datos privado a una computadora de confianza en otro lugar de internet.

Este problema puede resolverse usando redes *virtuales*. Las redes virtuales pueden vivir arriba de otras redes, pero no interfieren interfieren entre ellas. En su mayor parte, las redes virtuales se ven como simples LAN, aun cuando pueden extenderse a lo largo del mundo. Pero las redes virtuales pueden no ser seguras aunque se uso cortafuegos, porque el tráfico que fluye a traves de estas hacia internet puede ser visto por otras personas.

Cuando se introduce cifrado de datos, podemos formar una verdadera VPN, Otras personas pueden ver el tráfico cifrado, pero no pueden saber como decifrar este (necesitan conocer la llave para esto), no pueden leer la información que fluye a traves de la VPN, Esto es para lo que tinc fue hecho.

Tinc usa datagramas IP normales para encapsular datos que viajan sobre el enlace de red VPN. En este caso está también claro que la red es *virtual*, porque ningún enlace de red directo tiene que existir entre los participantes.

Como es el caso con cualquier tipo de VPN, alguien podría escuchar secretamente, o peor, alterar datos. Aquí es probablemente aconsejable el cifrar los datos que fluyen sobre la red.

TINC

Yo realmente no recuerdo lo que nos llevó a empezar, pero debe de haber sido idea de Guus. Él escribió una aplicación simple (aproximadamente 50 líneas en C) que usó el dispositivo *ethertap* que linux tiene desde el núcleo 2.1.60. No funcionó inmediatamente y él la mejoró un poco. En esta fase, el proyecto se llamó simplemente `'vpnd'`.

Desde entonces, mucho ha cambiado -- por así decirlo.

Tinc ahora soporta cifrado, consiste en un solo demonio (tincd) para la recepción y el envío de información, se ha vuelto un paquete profesional completo.

Mucho puede ser, y será, mejorado. Hay varias cosas que me gustaría ver en las futuras versiones de tinc. No todo estará disponible en el futuro cercano. Nuestro primer objetivo es hacer que tinc trabaje perfectamente, y luego agregar rasgos más avanzados.

Entretanto, siempre estaremos abiertos y disponibles hacia las nuevas ideas.

Configurando un sistema Linux

Este capítulo contiene información sobre cómo configurar un sistema Linux para el uso de tinc.

Configurando el Núcleo

Dado que esta implementación particular sólo corre en núcleos 2.1 o mayores, debería conseguir uno (los 2.2 son actuales en este momento). Una migración a 2.0 no es posible, a menos que alguien migre los dispositivos ethertap y netlink a 2.0.

Si no está familiarizado con el proceso de configurar y compilar un nuevo núcleo, debería leer primero el [Núcleo COMO](#). ¡Haga eso ahora!

Aquí están las opciones que usted tiene que activar al configurar un nuevo núcleo.

```
Code maturity level options
[*] Prompt for development and/or incomplete code/drivers
Networking options
[*] Kernel/User netlink socket
<*> Netlink device emulation
Network device support
<*> Ethertap network tap
```

Cualquier otra opción no mencionada aquí no es relevante a tinc. Si usted decide construirlos como módulos dinámicos, es una buena idea agregar estas líneas en ``/etc/modules.conf``.

```
alias tap0 ethertap
alias char-major-36 netlink_dev
```

Finalmente, construya el núcleo y reinicie la maquina. Desgraciadamente no es posible insertar estos módulos en un Núcleo que se está ejecutando.

Archivos Necesarios

Archivos de dispositivos

Primero, necesitará el archivo de dispositivo especial que forma la interfaz entre el Núcleo y el demonio.

```
mkknod -m 600 /dev/tap0 c 36 16
chown 0.0 /dev/tap0
```

Los permisos serán ahora tales que sólo el super usuario(root) puede leer y escribir en este archivo. Esto debería ser así, porque es más fácil que se filtre información por aquí. Esto, sin embargo, implica que usted tendría que ejecutar tincd como root.

Si usted quiere, también se puede crear más archivos de dispositivos que se numerarían de 0 a 15 con números menores de dispositivos de 16 a 31. Todos deben pertenecer al root y deben tener permisos 600.

```
`/etc/networks`
```

Puede agregar una línea en ``/etc/networks`` para que sus vpn tengan nombres simbólicos. Por ejemplo:

```
Mi_vpn 10.0.0.0
```

```
`/etc/services`
```

Puede agregar estas líneas en ``/etc/services``. El resultado es que puede proporcionar a una ``tinc`` como un número de puerto válido a algunos programas. El número 655 está registrado en el IANA.

```
tinc          655/tcp      TINC
tinc          655/udp      TINC
#            Ivo Timmermans <itimmermans@bigfoot.com>
62;
```

Preparando los dispositivos

Antes de que pueda empezar a transmitir datos sobre el túnel tinc, debe preparar los dispositivos de red ethertap.

Primero, decida qué direcciones IP quiere asociar con estos dispositivos, y qué máscara de red deben tener. También necesitara estos números cuando configure tinc. Vea la sección [Configurando tinc](#).

No importa mucho que se hace primero, si preparar los dispositivos red o configurar tinc. Pero deben hacerse ante de usar tincd.

La configuración de dispositivo ethertap es bastante simple, simplemente escriba esto:

```
ifconfig tapn hw ether fe:fd:xx:xx:xx:xx
```

El *n* aquí es el número del dispositivo ethertap que quiere usar. Debe ser uno de los mismos *n* que uso para `/dev/tapn`. Las *xxs* son cuatro números hexadecimales (0--ff). En las versiones anteriores de tincd no importaban lo que eran, pero los nuevos Núcleos requieren que sean puestas las direcciones de ethernet. De hecho, el comportamiento estaba equivocado, se requiere que los *xxs* mapeen `Mi_IP_VPN`.

```
ifconfig tapn IP netmask mask
```

Esto activará el dispositivo con una dirección red *IP* y con una máscara de red *mask*.

[Instalando Tinc](#)

Primero consígalo. Esta es la [página principal](#), que tiene la suma de verificación (checksums) de los archivos listados; puede desear verificar éstos con md5sum antes de continuar.

Tinc viene en un paquete autoconf/automake, que simplemente puede tratar como cualquier otro paquete. Sólo tiene que descomprimirlo, escribir "Configure" y luego "Make".

Las instrucciones más detalladas están en el archivo `INSTALL`, que es incluido en la distribución fuente.

[Configurando Tinc](#)

[Redes múltiples](#)

Es perfectamente aceptable correr más de un demonio tinc. Sin embargo, en su forma predefinida, pronto notará que no puede usar dos archivos diferentes de configuración sin la opción `-c`.

Hemos pensado en otra manera de tratar esto: nombres de red. Esto significa que usted llama a tincd con el argumento `-n` que asignará un nombre a este demonio.

El efecto de esto es que los demonios busquen su configuración en `/etc/tinc/nn/`, donde *nn* es un argumento a la opción `-n`. Notará que aparece en syslog como "tincd.nn."

Sin embargo, no es estrictamente un requisito llamar a tinc con la opción `-n`. En este caso, el nombre de la red estaría simplemente vacío, y se usará como tal. Tinc busca archivos ahora en `/etc/tinc/`, en lugar de `/etc/tinc/nn/`; el archivo de configuración debe ser `/etc/tinc/tincd.conf`, y se espera ahora que los `passphrases` (N.T. `passphrases` es una contraseña o palabra de paso (password), que al ser bastante larga se le llama frase de paso) estén en `/etc/tinc/passphrases/`.

Es recomendable usar este rasgo de tinc, porque será él quien decida con que demonio hablar. Asumiremos que lo usa.

[Cómo trabajan las conexiones](#)

Antes de seguir, primero un poco de cómo tinc ve las conexiones.

Cuando tinc se pone en marcha, lee en el archivo la configuración y analiza las opciones de la línea de comandos. Si ve un valor "ConnectTo" en el archivo, intentará conectarse a ese servidor, en el puerto dado. Si esto falla, tinc termina.

[Archivo de configuración](#)

La configuración actual del demonio se hace en el archivo `/etc/tinc/nn/tinc.conf`.

Este archivo consiste en comentarios (las líneas empiezan con #) o asignaciones de la forma:

Variable = Valor.

En los nombres de variables se distingue entre mayúsculas o minúsculas, y se ignora cualquier espacio, etiquetas, nueva línea y retorno de carro. Nota: no se requiere que ponga "=", pero se usa para mejorar la legibilidad. Si lo omite, recuerde reemplazarlo con por lo menos un carácter espacial.

Variables

Aquí está todas las variables válidas, listadas en orden alfabético:

ConnectPort = port

Conéctese al host (dado en la directiva ConnectTo) en el puerto "port". El puerto puede darse en decimal (valor por defecto), octal (cuando es precedido por un solo cero) o hexadecimal (prefijó con 0x). El puerto es el número del puerto para las conexiones UDP y TCP (meta).

ConnectTo = (IP address|hostname)

Especifica a qué host conectarse al arrancar. Si la variable "ConnectPort" se omite, entonces tinc intentará conectarse al puerto 655. Si usted no especifica un host con "ConnectTo", sin tener en cuenta si un valor para "ConnectPort" se da, tinc no se conectará en absoluto, y escuchará en cambio simplemente las conexiones entrantes. Sólo el iniciador de un tinc VPN puede necesitar esto.

ListenPort = port

Escuche en el puerto local "port". La computadora que se conecta a este demonio debe usar este número como el argumento para su "ConnectPort". De nuevo, el valor por defecto es 655.

MyOwnVPNIP = local address[/maskbits]

La dirección local es el número que los demonios propagarán a otro demonios en la red cuando se identifican. Aquí será el nombre del archivo de passphrase que el otro extremo espera encontrar en el passphrase. La dirección local es la dirección IP del dispositivo Tap, no la dirección IP real del host donde tincd esta corriendo. Debido a los cambios en recientes núcleos, es también necesario que usted haga coincidir la dirección ethernet (también conocida como MAC) y la dirección de IP (vea el ejemplo). maskbits es el número de bits en 1 en la parte de la netmask(mascara de red).

MyVirtualIP = local address[/maskbits]

Esto es un alias para "MyOwnVPNIP".

Passphrases = directory

El directorio donde tinc buscara las passphrases cuando alguien intenta conectarse. Por favor vea la pagina del manual de genauth(8) para más información sobre el passphrases usado por tinc.

PingTimeout = number

El número de segundos de inactividad que tinc esperara antes de enviar una sonda(ping) al otro extremo. Si el otro extremo no contesta dentro de esa misma cantidad de segundos, la conexión se termina, y se notificara de esto.

TapDevice = device

El dispositivo ethertap a usar. Note que se puede usar sólo un dispositivo por demonio. La paginas info del paquete tinc contiene más información sobre como configurar un dispositivo ethertap en linux.

VpnMask = mask

La mascara de red que define el alcance de la VPN, Esta opción no es usada por el demonio tinc sino por los script de inicio(startup)para sonigurar los dispositivos ethertap correctamente

Ejemplo

Imagine la situación siguiente. Una compañía A-base quiere conectar tres oficinas en B, C y D usando internet. Las cuatro oficinas tienen una 24/7 (24 horas los 7 días) conexión a internet.

A va a servir como el centro de la red. B y C conectarán a A, y D se conectará a C. Cada oficina asignará sus propios IP de red, 10.x.0.0.

```
A: red 10.1.0.0 máscara 255.255.0.0 gateway 10.1.54.1 internet IP 1.2.3.4
B: red 10.2.0.0 máscara 255.255.0.0 gateway 10.2.1.12 internet IP 2.3.4.5
C: red 10.3.0.0 máscara 255.255.0.0 gateway 10.3.69.254 internet IP 3.4.5.6
D: red 10.4.0.0 máscara 255.255.0.0 gateway 10.4.3.32 internet IP 4.5.6.7
```

"gateway" es la dirección IP VPN_IP de la máquina que está ejecutando el tincd. "internet IP" es la dirección IP del cortafuego que no necesita ejecutar tincd pero debe tener un puerto de forwarding de TCP y UDP en 655 (a menos que configure otro).

En este ejemplo, se asume que eth0 es la interfaz que apunta a la LAN interna de la oficina. Esto podría ser igual que la interfaz que lleva a internet.

Para A

Ase configuraría como esto:

```
ifconfig tap0 hw ether fe:fd:0a:01:36:01
ifconfig tap0 10.1.54.1 netmask 255.0.0.0
ifconfig eth0 10.1.54.1 netmask 255.255.0.0 broadcast 10.1.255.255
```

y en /etc/tinc/tincd.conf:

```
TapDevice = /dev/tap0
MyVirtualIP = 10.1.54.1/16
VpnMask = 255.0.0.0
```

Para B

```
ifconfig tap0 hw ether fe:fd:0a:02:01:0c
ifconfig tap0 10.2.1.12 netmask 255.0.0.0
ifconfig eth0 10.2.43.8 netmask 255.255.0.0 broadcast 10.2.255.255
```

y en /etc/tinc/tincd.conf:

```
TapDevice = /dev/tap0
MyVirtualIP = 10.2.1.12/16
ConnectTo = 1.2.3.4
VpnMask = 255.0.0.0
```

Note aquí que la dirección interna (en eth0) no tiene por que ser igual que el dispositivo tap0. También, ConnectTo se da para que nadie pueda conectarse a este nodo.

Para C

```
ifconfig tap0 hw ether fe:fd:0a:03:45:fe
ifconfig tap0 10.3.69.254 netmask 255.0.0.0
ifconfig eth0 10.3.69.254 netmask 255.255.0.0 broadcast 10.3.255.255
```

y en /etc/tinc/A/tincd.conf:

```
MyVirtualIP = 10.3.69.254/16
ConnectTo = 1.2.3.4
ListenPort = 2000
VpnMask = 255.0.0.0
```

C ya tiene otro demonio que corre en el puerto 655, entonces se reservar otro puerto para los que se conecten. Se usa el nombre de red para distinguir entre los dos. tinc se ejecuta con "tincd -n A".

Para D

```
ifconfig tap0 hw ether fe:fd:0a:04:03:20
ifconfig tap0 10.4.3.32 netmask 255.0.0.0
ifconfig tap0 10.4.3.32 netmask 255.255.0.0 broadcast 10.4.255.255
```

y en /etc/tinc/tincd.conf:

```
MyVirtualIP = 10.4.3.32/16
ConnectTo = 3.4.5.6
ConnectPort = 2000
VpnMask=255.0.0.0
```

D estará conectando a C que tiene un tincd que corre para esta red en el puerto 2000. Aquí se debe poner un ConnectPort.

Autenticación

A, B, C y D generan su passphrase con genauth 2048, la salida se guarda en /etc/tinc/passphrases/local, salvo C, donde debe ser /etc/tinc/A/passphrases/local.

A guarda una copia del passphrase de B en /etc/tinc/passphrases/10.2.0.0

A guarda una copia del passphrase de C en /etc/tinc/passphrases/10.3.0.0

B guarda una copia del passphrase de A en /etc/tinc/passphrases/10.1.0.0

C guarda una copia del passphrase de A en /etc/tinc/A/passphrases/10.1.0.0

C guarda una copia del passphrase de D en `/etc/tinc/A/passphrases/10.4.0.0`

D guarda una copia del passphrase de C en `/etc/tinc/passphrases/10.3.0.0`

Ejecución

A tiene que ejecutar su tincd primero. Entonces viene B y C, donde C tiene que proporcionar la opción "-n A", porque aquí se tiene más de una red del tinc. Finalmente, el tincd de D se ejecuta.

Corriendo Tinc

Ejecutar tinc no es así de fácil como teclear "tincd" y esperar que todo funcione de la manera que se quiso. En cambio, el uso de tinc es un proyecto que involucra relaciones de confianza en más de una computadora.

Manejo Llaves

Antes de intentar ejecutar tinc, se tienen que crear los passphrases. Cuando tinc intenta hacer una conexión, intercambia algunos datos sensibles. Antes de hacer esto, le gusta saber si el otro extremo es confiable.

Para hacer esto, ambos extremos deben tener un poco de conocimiento sobre el otro. En el caso de tinc éste es la autenticación passphrase.

Este passphrase es un número que es escogido al azar. Este número se envía entonces a las otras computadoras que quieren hablar directamente con nosotros. Para evitar romper seguridad, esto debe hacerse sobre un cauce seguro conocido (como ssh o similar).

Todos los passphrases se guardan en el directorio de passphrases que normalmente es `/etc/tinc/nn/passphrases/`, pero este puede cambiarse usando la opción "Passphrases" en el archivo de configuración.

Para generar un passphrase, ejecute "genauth". genauth toma un argumento que es la longitud del passphrase en bits. La longitud del passphrase debe estar en el rango de 1024--2048 para una llave de 128 bits. genauth crea un número al azar de la longitud especificada, y lo pone en la salida estándar.

Cada computadora que quiere participar en el VPN debe hacer esto, y guardar la salida en el directorio de passrasphes, en el archivo `'local'`.

Cuando cada computadora tiene su propia llave local, debe copiarlo a la computadora con la que quiere hablar directamente. Esto debe hacerse vía un cauce seguro, porque es información sensible. Si esto no se hace con seguridad, alguien podría forzar la entrada después de usted.

Esos archivos de passphrase no locales deben tener el nombre de la dirección IP VPN con la que ellos se anuncian. Por ejemplo, si una computadora nos dice que es 10.1.1.3 con netmask 255.255.0.0, el archivo debería llamarse 10.1.1.3, y no 10.1.0.0.

Opciones en Tiempo de Ejecución

Además de las opciones en el archivo de configuración, tinc también acepta algunas opciones de línea de comandos.

Esta lista es una versión más larga que la de la pagina del manual. Lo ultimo se genera automáticamente, para que pueda ser más actualizable.

`-c, --config=FILE`

Lee opciones de configuración desde el archivo FILE. El valor por defecto es `'/etc/tinc/nn/tinc.conf'`.

`-d`

Incrementa el nivel de depuración. El más alto conseguido es en el que más se registra en los logs, todo vía syslog. 0 es el valor por defecto, sólo información básica de algunos intentos de conexión se registran. Poniéndolo a 1 se registra un poco más, todavía no muy alentador. Con dos opciones -d, tincd registrara información protocolar que puede ponerse bastante ruidosa. Tres o más opciones -d, harían que cada paquete que sale o entra probablemente genere más datos que los paquetes en si.

`-k, --kill`

Intente matar un tincd y termina. Una señal TERM(15) se envía al demonio que tiene su PID en `/var/run/tincd.nn.pid`.

Como mata sólo un tincd, usted debe usar -n aquí si normalmente lo usa.

`-n, --net=NETNAME`

Conectar a la red NETNAME. Vea la sección [redes Múltiples](#).
-t, --timeout=TIMEOUT
Segundos a esperar antes de dar una interrupción. No debe ponerse demasiado bajo, porque a cada rato tinc dará interrupción, se desconectara y re-conectara de nuevo, lo que causará tráfico de la red innecesario y mensajes de log.
--help
Despliegue un recordatorio corto de estas opciones de tiempo de ejecución y termina.
--version
Muestra información de versión y termina.

Información técnica

Filosofía básica del modo de trabajo de tinc

Tinc es un demonio que toma datos VPN y transmite estos a otra computadora Host sobre la infraestructura existente de Internet.

Una vista previa del modo de trabajo de tinc

Los propios datos se leen de un archivo de dispositivo de caracteres, el dispositivo llamado *ethertap*. Este dispositivo es asociado con una interfaz de la red. Puede leerse cualquier dato enviado a la interfaz de dispositivo, y cualquier dato escrito al dispositivo se envía a la interfaz. Datos a y desde el dispositivo se estructuran como si fuera una tarjeta ethernet normal, cada marco es precedido por dos direcciones MAC y un campo de *tipo de marco*.

Para que cuando tinc lea un marco del dispositivo ethernet determine su tipo. Actualmente, tinc puede manejar sólo marcos del Protocolo Internet versión 4 (IPv4), están haciéndose planes para soportar otros protocolos. Cuando tinc sabe que qué tipo de marco ha leído, también puede leer la dirección de origen y de destino de este.

Ahora es cuando el marco se cifra. Actualmente el único algoritmo de cifrado disponible es blowfish.

Cuando el cifrado está listo, es tiempo de transportar el paquete realmente a la computadora de destino. Se hace esto enviando el paquete sobre una conexión UDP al host de destino. Esto se llama *encapsulación*, el paquete de VPN (aunque ahora cifrado) se encapsula en otro datagrama IP.

Cuando el destino recibe este paquete, ocurre lo mismo, sólo que al revés. Así que realiza un descifrado del contenido del datagrama UDP, y escribe la información descifrada en su propio dispositivo ethertap.

La meta-conexión

Teniendo sólo una conexión de UDP disponible no es bastante. Aunque conveniente para transmitir datos, queremos poder enviar otra información confiablemente, como rutear y cifrar información a alguien.

TCP es una alternativa mejor, porque ya contiene protección contra información que es perdida, no como UDP.

Así que nosotros establecemos dos conexiones. Una para los datos cifrados de VPN, y una para la otra información, los meta-datos. Aquí, llamamos a la segunda conexión la meta-conexión. Podemos estar ahora seguros que la meta-información no se pierde en el camino a la otra computadora.

Como con cualquier comunicación, debemos tener un protocolo, para que todos sepan como comunicarse y cómo se debe reaccionar. Como tenemos dos conexiones, también tenemos dos protocolos. El protocolo usado para los datos de UDP es el "data-protocolo" y el otro es el "meta-protocolo".

La razón por la que no se usa TCP para ambos protocolos es que UDP es mucho mejor para encapsulación, incluso mixta este es menos confiable. El problema real es que cuando TCP es usado para encapsular un stream TCP que esta sobre la red privada, para cada paquete enviado habria tres ACK's enviados en lugar de uno. Aun mas, si se da un timeout, ambos TCP streams serian sensibles al timeout, y ambos reenviarían los paquetes.

Algo de cifrado en tinc y otros problemas de seguridad relacionados.

tinc consiguio su nombre de "TINC", iniciales de There Is No Cabal (Allí No Hay Conspiración); el hubo/hay de la Conspiración alega a una organización que se decía que tenia ojos en Internet. Como esto es exactamente lo que usted *no* quiere, nombramos al proyecto tinc después de TINC.

Pero para ser "inmune" a las escuchas secretas, usted tendrá que cifrar sus datos. Como tinc es un demonio VPN Seguro

(SVPN), hace eso exactamente: cifrar.

Este capítulo es una mezcla de ideas, razonamientos y explicaciones, por favor no lo tome demasiado en serio.

Manejo de Llaves

Usted no puede enviar simplemente una llave de cifrado privada a su par, porque alguien podría estar escuchando. De modo que tendrá que negociar sobre una llave compartida pero confidencial. Una manera de hacer esto es usar el protocolo de "Intercambio de Llaves Diffie-Hellman" (<http://www.rsa.com/rsalabs/faq/html/3-6-1.html>). La idea es como sigue.

Usted tiene dos participantes A y B que quiere estar de acuerdo sobre una llave de cifrado confidencial compartida. Ambas partes tienen algún número primo grande p y un generador g . Estos números pueden conocerse al mundo externo, y pueden ser incluidos en la distribución de la fuente.

Ambas partes generan una llave confidencial entonces. A genera a , y calcula $g^a \text{ mod } p$. Este es entonces mandado a B; mientras B calcula $g^b \text{ mod } p$, y transmite este a A. a y b deben ser mayores que $p-1$.

Estas llaves privadas se generan en el inicio, y ellas no se cambian mientras la conexión existe. Un posible rasgo en el futuro es cambiar las llaves dinámicamente, todas las horas por ejemplo.

Ambas partes calculan $g^{ab} \text{ mod } p = k$, k es la nueva llave compartida, pero secreta.

Para obtener un k importante de una longitud suficiente (128 bits en nuestro vpnd), p debe tener $2^{129}-1$ o más.

Autenticación

Como el protocolo de Diffie-Hellman es en sí mismo vulnerable al "el ataque del hombre-en-el-medio," nosotros debemos introducir un sistema de autenticación.

Nosotros permitiremos que A transmita un passphrase que también conoce B cifrado con g^a , antes que A le envíe esto a B. De esta manera, B puede verificar si A realmente es A o simplemente alguien más.

Este passphrase deben ser de 2304 bits para un sistema de cifrado simétrico. Pero como un sistema asimétrico es más seguro, nosotros podríamos hacerlo con 2048 bits. Esto sólo sirve si el passphrase es muy el azar.

Estos passphrases podrían guardarse en un archivo solo leíble por el root; ej. `'/etc/vpn/passphrases'`.

La única cosa que necesita tener en cuidado es cómo A le anuncia su passphrase a B.

Protegiendo sus datos

Ahora nosotros hemos escondido nuestros datos firmemente. Pero un cracker malévolo todavía puede molestarnos alterando los datos cifrados al azar que el intercepta.

Sobre Nosotros

Informacion de Contacto

La página principal de tinc está en <http://tinc.nl.linux.org/>, este servidor se localiza en los Países Bajos.

Nosotros tenemos un canal IRC en la red IRC de Open Projects. Conectarse a irc.openprojects.net, y entrar en el canal #tinc.

Autores

Ivo Timmermans (zarq) (itimmermans@bigfoot.com)

El codificador/hacker principal y el que mantiene el paquete.

Guus Sliepen (guus)

Originador de todo esto, coautor.

Wessel Dankers (Ubiq)

Ofuscador general del código.

Gracias a: Dekan, Emphyrio, vDong

Saludos a: braque, Fluor, giggles, macro, smoke, tribbel

Traductor

Carrasco Matias (mcarrasco@softwork.com.ar)

Indice de Conceptos

Ir a: [a](#) - [c](#) - [d](#) - [e](#) - [m](#) - [p](#) - [s](#) - [t](#) - [v](#)

a

- [ataque hombre-en-el-medio](#)

c

- [Cabal](#)
- [Clave secreta](#)
- [Conexión](#)

d

- [data-protocol](#)
- [Diffie-Hellman](#)

e

- [encapsulación](#)
- [ethertap](#)

m

- [meta-protocol](#)

p

- [passphrase](#)
- [privado](#)

s

- [SVPN](#)

t

- [tinc](#)
- [tincd](#)
- [Tipo de Marco](#)

v

- [virtual](#), [virtual](#)
-