VPN PPP-SSH Mini-HOWTO

Scott Bronson

bronson@trestle.com

Una VPN PPP-SSH è con ogni probabilità il più semplice tipo di VPN da configurare. Non occorre altro che le comuni utility PPP e SSH per creare un tunnel di rete tra due host.

Tradotto da Manuel Spezzani, Alberto Ronzoni, Daniele Gozzi, Laura Ferretti e Daniela Saladino nell'ambito del corso di Inglese Tecnico presso l'Università degli Studi di Modena e Reggio Emilia (a.a. 2004/2005)

Sommario

1. Introduzione

Le tecniche descritte in questo HOWTO utilizzano PPP per convertire i pacchetti in un flusso di caratteri e SSH per crittarlo e trasmetterlo al computer remoto. Alla maggior parte degli amministratori di sistema sono familiari gli strumenti e i file di configurazione necessari per impostare una VPN PPP-SSH.

Benché funzioni bene con carichi moderati su una connessione affidabile, si fa presente che una VPN PPP-SSH è soggetta ad alcuni problemi di scalabilità. Una lista di benefici è inclusa in la Sezione 2.1 e le controindicazioni in la Sezione 2.2 così è possibile decidere autonomamente se una VPN PPP-SSH è idonea alle proprie necessità.

1.1. Copyright

1.2. Disclaimer

You use the information in this document entirely at your own risk. I especially make no guarantees as to the legality or cryptographic strength of the techniques described here. If you feel that you cannot take full responsibility for your setup, then you need to put down this HOWTO and hire one of the many excellent companies who provide accountable, professional VPN service.

1.3. Credits

I took some notes as I adapted Bart Trojanowski's excellent instructions (http://www.jukie.net/~bart/security/vpn/) to a newer version of PPP running on my Debian system. A few weeks later, I converted the notes into SGML. Eventually, those evolved into this HOWTO.

Bart's instructions were based on Arpad Magosanyi's good but now fairly dated VPN Mini-HOWTO (http://www.linuxdoc.org/HOWTO/mini/VPN.html). If you run into troubles and my document doesn't seem to help, or if you're running an older version of the Linux kernel or PPP, you'll definitely want to give his HOWTO a read.

2. Introduzione

2.1. Benefici di PPP-SSH

Dall'installazione di una Virtual Private Network PPP-SSH derivano svariati benefici. È relativamente immediato, impiega strumenti comuni per cui non necessitano modifiche e probabilmente non richiederà un riavvio del sistema per la creazione del collegamento. Segue una lista più completa:

È facile da installare

> Probabilmente non sarà necessario applicare patch al kernel o compilarlo nuovamente, installare LILO, riavviare o eseguire altre attività amministrative potenzialmente pericolose. PPP e SSH sono inclusi nella maggior parte delle distribuzioni, e la maggior parte dei kernel è preconfigurato per utilizzarli.

È facile da configurare

Non dovrebbe essere necessario modificare i file di configurazione esistenti. Semplicemente basterà apportare modifiche al file di script fornito più avanti in questo stesso documento, che contiene tutte i dati di configurazione per la VPN, e poi eseguirlo sulla macchina client. Qualsiasi configurazione preesistente di PPP e SSH dovrebbe continuare a funzionare correttamente.

Nessuna modifica al firewall

Se il protocollo SSH attraversa il firewall, allora anche PPP su SSH lo attraverserà. (Se non si sta già impiegando SSH: perché no? Ai giorni nostri è uno strumento quasi indispensabile per gli amministratori di sistema.)

Non necessita di utilizzare il routing manuale, per evitare danni

pppd imposta automaticamente i parametri di routing. Inoltre, nel caso in cui fosse necessario un routing particolarmente complesso, è molto semplice inserire comandi personalizzati nel file di script.

Non è necessario un indirizzo IP statico

Le VPN PPP-SSH non hanno difficoltà nel gestire indirizzi IP dinamici. Il client deve essere in grado di trovare il server a cui connettersi, chiaramente, quindi basta utilizzare un DNS. Installare una VPN su una connessione dialup non presenta problemi.

È semplice creare Tunnel Multipli

Impostare tunnel multipli verso un unico computer è semplice. Occorre semplicemente assicurarsi che gli indirizzi IP delle interfacce di rete del tunnel siano distinti.

2.2. Inconvenienti del PPP-SSH

Questo tipo di VPN presenta qualche difficoltà. Solitamente, se trascurato non funziona molto bene. Se siete alla ricerca di VPN efficienti che una volta configurate possono essere dimenticate, probabilmente troverete PPP-SSH leggermente frustrante. Alcune alternative sono descritte in la Sezione 2.4.

Provare a gestire una connessione TCP

Nel caso in cui la connessione TCP SSH venisse interrotta per qualsiasi ragione, la VPN cesserebbe di funzionare così come tutti i tunnel delle connessioni TCP. Se avete un collegamento poco affidabile (cioè è difficile effettuare il download di più di poche decine di megabyte in una volta sola) dovrete riavviare la VPN molto spesso.

Inviare pacchetti IP tramite stream TCP

Il protocollo TCP consiste di flussi di dati costruiti sui pacchetti IP. Tentando *successivamente* di inviare i pacchetti IP sullo stream TCP (come si vorrebbe fare), il conflitto fra i due protocolli diventa evidente. Nella maggior parte dei casi, si manifesta attraverso strani ritardi, perdita di informazioni, e variazioni anomale. A volte si potrebbero manifestare problemi durante il caricamento, altre in quasi assenza di traffico. A parte cambiare l'intero modello OSI (improponibile), non c'è molto altro che si può fare a riguardo.

Tendenza alla saturazione della banda

Per qualche motivo, quando il carico della rete aumenta, una connessione TCP ridiretta nel tunnel tende a saturare tutta la banda disponibile mentre le altre vengono ignorate. Questo porta a timeout e perdita di connessioni. Teoricamente, questo problema potrebbe essere risolto.

Non si può avere la certezza della perdita della connessione

I keepalive sono piccoli pacchetti inviati per comunicare all'altra macchina che la connessione è ancora esistente. Se il carico della rete diventa troppo elevato, i keepalive saranno ritardati. L'altra macchina assumerà erroneamente che la connessione sia stata interrotta e chiuderà il proprio lato della connessione.

Senza keepalive, tuttavia, non c'è modo per nessuna delle due macchine di comunicare che la connessione è stata interrotta. Quando una macchina tenta di ripristinare la connessione e l'altra pensa che questo sia già stato fatto, sorgerà la confusione. Molto spesso il risultato è che si vedono molteplici dispositivi per la connessione ppp, route duplicate, e tunnel che apparentemente sono attivi ma che perdono tutti i pacchetti. Un uso indiscriminato di "killall -9 pppd" solitamente riporta le cose a posto. Uno script di avvio più intelligente probabilmente potrebbe migliorare la situazione.

Troppe connessioni simultanee provocano problemi

Quando uso una normale connessione PPP tramite un modem 56K e Postfix apre più di 10 connessioni per inviare la posta, tutto funziona bene. Però, tentando la stessa operazione su una VPN con tunnel su una connessione DSL molto più veloce, si verifica uno stallo. Il tempo di ping si alza vertiginosamente (2 minuti e oltre), il traffico si riduce per un momento, quindi si interrompe completamente. L'unico modo per ristabilire la funzionalità della rete è riavviare il tunnel. Non sono sicuro che questo sia un bug o una limitazione intrinseca. Ridurre il numero di connessioni utilizzate da Postfix per inviare la posta in uscita risolve il problema nel mio caso...

Grande overhead e grande latenza

Il tempo di ping sulla mia connessione a 56K è solitamente nell'ordine di 130-170 ms. Tuttavia, il tempo di ping su una VPN PPP-SSH avviata sulla stessa connessione è nell'ordine di 300-330 ms. Utilizzare la compressione PPP può aiutare abbastanza nel caso in cui si stiano trasmettendo dati comprimibili. Ad esempio, le email sono comprimibili, mentre i file Vorbis no.

2.3. Letture Consigliate

FAQ VPN

La VPN FAQ su http://kubarb.phsx.ukans.edu/~tbird/vpn/FAQ.html è veramente un'ottima risorsa. È completa, ragionevolmente aggiornata, e non ha paura di esprimere un'opinione.

Linux Kernel HOWTO

Se il kernel non dispone delle capacità di PPP e IP Forwarding, il Linux Kernel HOWTO (http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html) spiega come ricompilare un kernel per aggiungerle. Inoltre spiega come inserire e togliere il modulo PPP dal kernel.

PPP HOWTO

Spiega come installare e configurare il demone PPP se la distribuzione non lo fa automaticamente. Inoltre contiene un'ottima sezione che spiega come collegare due reti attraverso PPP. Questo è esattamente quello che stiamo facendo, eccetto per il fatto che stiamo aggiungendo il supporto per la crittazione. Può essere reperito in http://www.linuxdoc.org/HOWTO/PPP-HOWTO/index.html.

SSH HOWTO

Spero che esista un SSH HOWTO! Per adesso, la documentazione contenuta nella propria distribuzione dovrebbe essere un buon inizio. Si può anche visitare il sito web di OpenSSH OpenSSH web site (http://www.openssh.org/).

Documentazione per il networking

Se non si è molto abili con il networking, si può consultare la guida a Linux Network Administrators (http://www.linuxdoc.org/LDP/nag2/index.html). È un'eccellente introduzione ai principali concetti che verranno utilizzati in seguito. Si può inoltre trovare il Linux Networking HOWTO presso http://www.linuxdoc.org/HOWTO/Networking-Overview-HOWTO.html, che è un'utile introduzione, specialmente per le sezioni su TCP/IP, PPP, e sul tunneling.

2.4. Alternative

Ora nel mondo ci sono molte tecnologie VPN. Se PPP-SSH non soddisfa le necessità, si può consultare uno dei seguenti pacchetti.

ipsec

ipsec descrive un insieme di protocolli di basso-livello, ESP (http://andrew2.andrew.cmu.edu/rfc/rfc2406.html) e AH (http://andrew2.andrew.cmu.edu/rfc/rfc2402.html), per eseguire l'autenticazione e crittazione a livello dei pacchetti. Impiega inoltre un protocollo di alto-livello, IKE (http://andrew2.andrew.cmu.edu/rfc/rfc2408.html), per negoziare parametri di connessione e scambiare chiavi di crittazione.

FreeS/WAN è probabilmente il migliore Linux ipsec implementato oggi. Può risultare molto difficile installarlo, specialmente per chi non ha molta familiarità con il networking, tuttavia è molto stabile una volta configurato. Si può saperne di più in FreeS/WAN home page (http://www.freeswan.org/).

Un'altra efficiente e libera implementazione di ipsec è Cerberus (http://www.antd.nist.gov/cerberus/). Sfortunatamente, l'Istituto Nazionale di Standard e Tecnologie (NIST) distribuisce Cerberus solo a cittadini statunitensi e canadesi che attualmente vivono negli Stati Uniti o in Canada. Perciò, a seconda della persona che lo richiede, diventa moderatamente difficile o effettivamente impossibile ottenere Cerberus.

PPTP

PPTP (Point-to-Point Tunnelling Protocol) è un protocollo VPN sviluppato da Microsoft RFC2637 (http://andrew2.andrew.cmu.edu/rfc/rfc2637.html). È una tecnologia molto diffusa e di facile comprensione e ha molte implementazioni mature su tutte le piattaforme computer comunemente usate. Comunque PPTP è generalmente considerato un protocollo con una sicurezza alquanto debole (http://www.counterpane.com/pptp.html).

Probabilmente la migliore implementazione di Linux PPTP è PoPToP, che si può trovare presso http://poptop.lineo.com/.

CIPE

CIPE è un protocollo di Olaf Titz che incapsula il traffico IP in pacchetti UDP. Esso ha sia una versione Linux (http://sites.inka.de/sites/bigred/devel/cipe.html) che una versione Windows (http://cipe-win32.sourceforge.net/). Non l'ho ancora utilizzato, ma è in pieno sviluppo e sembra essere molto promettente. Per ulteriori informazioni, CIPE-MASQ Mini-HOWTO (http://www.linuxdoc.org/HOWTO/mini/Cipe+Masq.html) è una lettura concisa ma piena di informazioni.

3. Installazione del Software

3.1. Terminologia

Poiché l'installazione di VPN assomiglia molto a una transazione client-server, mi servirò di questa terminologia per dare un nome al computer alla fine di ogni passaggio:

Server

È un computer che aspetta passivamente l'arrivo di richieste di connessione VPN. Fuunziona completamente incustodito.

Client

È un computer che inizializza le richieste di connessione, chiedendo al Server di creare una VPN.

3.2. Requisiti

- Nel kernel deve essere presente, compilato, il supporto a TCP/IP e PPP. Quasi tutti le distribuzioni includono direttamente il supporto a PPP. Se la vostra non lo include, o se state usando un kernel particlare, includerò ulteriori dettagli a riguardo,in la Sezione 3.4.
- Bisogna installare il demone pppd. Questo probabilmente è incluso nella vostra distribuzione. Io sto usando ppp-2.3.11. Le versioni successive dovrebbero funzionare bene, come anche le versioni precedenti, finché supportano le opzioni "pty". Non è necessario installare chat o alcun altro strumento destinato a funzionare insieme a PPP: è sufficiente avere pppd.
- La macchina client deve avere installato il client ssh. Esistono svariate versioni di ssh in circolazione, ma dovrebbero funzionare tutte quante. Per la redazione di questo HOWTO è stato utilizzato OpenSSH (http://www.openssh.org/) per OpenBSD, versione 2.2.0p1.
- La macchina server deve disporre del demone sshd per fare fronte alle richieste del client. OpenSSH comprende un demone ssh molto buono.
- Per ultima cosa potrebbe essere richiesto di installare sudo sul server. È possibile trovare ulteriori informazioni riguardo a sudo in Linux Administrator's Security Guide (http://www.seifried.org/lasg/), nella sezione Tool Amministrativi e in Linux Security HOWTO (http://www.linuxdoc.org/HOWTO/Security-HOWTO.html),nella sezione Root Security. Inoltre potrebbe essere di aiuto leggere sudo's home page (http://www.courtesan.com/sudo/).

3.3. Progettazione

Per impostare un collegamento PPP-SSH occorre specificare i seguenti parametri:

Hostname del server

Qual'è lo hostname o l'indirizzo IP del server VPN?

Utente del server VPN

Sul vostro server, con quale nome utente il software per la VPN funzionerà? Questo HOWTO comprende istruzioni riguardanti la creazione di un utente di nome "vpn" specificamente per questo scopo. Il software *non* deve essere eseguito da root! Per esigenze di sicurezza e di registrazione degli accessi, si dovrebbe usare un account riservato.

Indirizzo IP dell'interfaccia del server

La VPN PPP-SSH necessita di interfacce di rete dedicate sia sul client che sul server. L'interfaccia sarà pppN, dove N è il numero della prima interfaccia ppp inutilizzata (per esempio sarà ppp1 se si sta già utilizzando ppp0 per una chiamata con il modem).

Sarà necessario specificare l'indirizzo IP per l'interfaccia sul server. Questo indirizzo sarà visibile solo da client e server (e dalle macchine sulle subnet a cui il client o il server potrebbero inoltrare i pacchetti de-mascherati).

Se non è noto l'indirizzo IP da specificare, si legga il capitolo 2.2 in Linux Network Administrators Guide (http://www.linuxdoc.org/LDP/nag2/index.html) e si guardi in particolare alla tabella 2-1. Per esempio, 192.168.0.1 è una buona scelta.

Indirizzo IP dell'interfaccia del client.

È necessario impostare l'indirizzo IP dell'interfaccia sul client. Deve ovviamente apartenere alla stessa rete dell'indirizzo del server. Non deve essere in conflitto con altre reti dal lato del client e non può essere lo stesso indirizzo IP dell'interfaccia di rete del server. Se è stato scelto 192.168.0.1 per la precedente risposta, probabilmente qui si dovrebbe usare 192.168.0.2.

La mia configurazione è:

```
SERVER_HOSTNAME = eldivino.domain.com
SERVER_USERNAME = vpn
SERVER_IFIPADDR = 192.168.3.1
CLIENT_IFIPADDR = 192.168.3.2
```

3.4. Configurare PPP

Il codice per PPP può essere compilato nel kernel stesso o può essere presente in moduli caricabili nel kernel. Se è stato compilato nel kernel, si può saltare al passo successivo: non occorre altro. Invece, se si sta caricando PPP sotto forma di moduli, bisogna assicurarsi che vengano caricati in modo appropriato.

Si può controllare se ppp è incluso nella lista di tutti i moduli correntemente caricati eseguendo lsmod. Si ricordi di controllare che il modulo PPP sia caricato sia sul client che sul server.

```
server$ /sbin/lsmod
```

```
Module                  Size  Used by
ppp                    20780   0 (unused)
slhc                    4376   0 [ppp]
3c59x                  21636   1

client$ lsmod
Module                  Size  Used by
ppp_deflate            40308   0  (autoclean)
bsd_comp                4076   0  (autoclean)
ppp                    20844   2  [ppp_deflate bsd_comp]
slhc                    4376   1  [ppp]
```

Se si è sicuri di avere compilato ppp come modulo, ma non è caricato nel kernel, si provi a caricarlo con modprobe.

```
# modprobe ppp
```

Se modprobe non ritorna alcun errore, si controlli di nuovo lsmod: a questo punto ppp dovrebbe essere presente nella lista. Se è così, significa che il modulo ppp non viene caricato al momento dell'avvio. Ciò non costituisce un problema se si ha intenzione di eseguire il demone per il kernel, poiché i moduli PPP verranno caricati su richiesta. Se non si intende fare così, si renderà necessario caricare i moduli al momento dell'avvio inserendo una riga contenente la sola parola "ppp" nel file /etc/modules.

Si veda Linux Kernel HOWTO (http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html) per maggiori informazioni al riguardo.

3.5. Permettere a SSH l'attraversamento del firewall

Il traffico di rete tra le due macchine (risultante dal tunnel, naturalmente) sarà basato sul solo protocollo SSH.

SSH impiega solo flussi TCP, nessun pacchetto UDP o ICMP. Il server ssh (sshd) normalmente è in ascolto sulla porta 22. Il client (dato che useremo il flag -P) utilizzerà solo le porte non privilegiate dalla 1024 alla 65535. Questa descrizione dovrebbe fornire informazioni sufficienti per potere impostare un firewall.

Per esempio, ecco i comandi per ipchains necessari a permettere le connessioni ssh verso il server. Consentiremo di passare alle connessioni dirette alla porta 22 sulla macchina locale e provenienti da qualsiasi porta sulla macchina remota. SI sostituisca eth0 con l'interfaccia di rete su cui viaggerà il traffico ssh e $IPADDR con l'indirizzo IP di quella interfaccia.

```
ipchains -A input  -i eth0 -p tcp -d $IPADDR 22 -j ACCEPT
ipchains -A output -i eth0 -p tcp ! -y -s $IPADDR 22 -j ACCEPT
```

I comandi seguenti sono necessari per impostare il firewall sulla macchina client. Non sono permesse le connessioni ssh in ingresso, mentre viene consentito al protocollo di passare tra la porta 22 della macchina remota e le porte non privilegiate su questa macchina. Ancora una volta, si sostituisca a eth0 l'interfaccia che trasporterà il traffico ssh, e a $IPADDR l'indirizzo IP di tale interfaccia.

```
ipchains -A input  -i eth0 -p tcp ! -y --source-port 22 -d $IPADDR 1024:65535 -j ACCEPT
ipchains -A output -i eth0 -p tcp -s $IPADDR 1024:65535 --destination-port 22 -j ACCEPT
```

4. Configurare il server

È necessario configurare il server affinché risponda alle richieste del client per creare il tunnel.

4.1. Creazione di un utente ad uso VPN

Le richieste di VPN con SSH devono essere dirette ad un particolare utente sul server. Per sicurezza e tracciabilità, si raccomanda di riservare un utente per le sole risposte alle richieste di VPN. Con i passi seguenti si creerà un utente di sistema con il nome "vpn" per questo scopo.

1. Come prima cosa, si crei l'account utente. Esistono account con ID in due intervalli: quello di sistema (solitamente tra 100 e 999) e quello degli utenti comuni (1000 e oltre). "--system" comunica a adduser di aggiungere l'utente nell'intervallo di sistema e di assegnargli /bin/false come shell di login. "--group" sta a indicare di creare un gruppo con lo stesso nome dell'utente, e di aggiungervi l'utente stesso.

   ```
   server# adduser --sytem --group vpn
   ```

2. Dato che l'utente vpn necessita di effettuare il login via ssh, si cambi la shell di vpn da /bin/false a /bin/bash nel file /etc/passwd. È possibile modificare /etc/passwd utilizzando vi o qualsiasi altro editor di testo.
3. Si crei una password per l'utente vpn. Può e dovrebbe essere molto complessa, dato che verrà digitata solo alcune volte mentre si imposta la VPN. Dopo non sarà più necessario digitarla di nuovo.

   ```
   server# passwd vpn
   Enter new UNIX password:
   Retype new UNIX password:
   passwd: password updated successfully
   ```

4. Ora si provi a collegarsi al server per assicurarsi di avere creato l'account correttamente.

   ```
   client% ssh eldivino.domain.com -l vpn
   vpn@eldivino's password:
   Linux eldivino 2.2.19 #6 Mon Jun 4 10:32:19 PDT 2001 i686 unknown
   No mail.
   vpn@eldivino:~$
   ```

   Potrebbe essere necessario un po' di tempo a ssh per collegarsi se non si è impostato correttamente il DNS inverso. Si può correggere tale problema in qualsiasi momento. Esso causerà soltanto un ritardo nell'avvio della VPN: non le impedirà di funzionare.

   Se il programma ssh rimane bloccato, allora il protocollo ssh viene probabilmente scartato da un firewall tra le due macchine. Si veda nuovamente la sezione la Sezione 3.5.

4.2. Configurazione di un login autenticato

Sarebbe scomodo dovere digitare una password ogni volta che si desidera avviare il collegamento VPN, quindi imposteremo l'autenticazione tramite RSA di SSH. Si salti questa sezione se non si considera impraticabile la soluzione di digitare una password ogni volta.

1. È necessario assicurarsi che l'account root sulla macchina client abbia una chiave pubblica nella propria home directory (~/root/.ssh/identity.pub). Se tale file non esiste è necessario crearlo. Come utente root si esegua ssh-keygen:

   ```
   # ssh-keygen
   Generating public/private rsa1 key pair.
   Enter file in which to save the key (/root/.ssh/identity):
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /root/.ssh/identity.
   Your public key has been saved in /root/.ssh/identity.pub.
   The key fingerprint is:
   15:61:57:7e:5c:26:91:09:5c:e6:10:b7:a1:74:bd:25 root@paradis
   ```

2. Ora, si copi identity.pub al posto del file authorized_keys nell'account vpn sul server. Quasi sicuramente sarà da creare eseguendo i seguenti comandi sul server.

   ```
   server# cd ~vpn
   server# mkdir .ssh
   server# chown root.vpn .ssh
   server# chmod 755 .ssh
   server# cd .ssh
   ```

   Si copi il file /root/.ssh/identity.pub sul client nella posizione ~vpn/.ssh/authorized_keys (è costituito da una sola riga). È possibile aggiungere altre linee ad authorized_keys, una per ogni client, se si desidera consentire più connessioni da client diversi.

   ```
   server# chown root.vpn authorized_keys
   server# chmod 644 authorized_keys
   ```

3. Diventando root sul client si provi a collegarsi con SSH al server. Si può utilizzare o meno l'opzione -P, a seconda di come è configurato il firewall sul client. Se la porta 22 è bloccata sul client (che non è una brutta idea se si ha un server ssh in esecuzione), l'opzione -P dice a ssh di usare una porta non privilegiata anche se sta eseguendo come utente privilegiato.

   ```
   client# ssh -P eldivino.domain.com -l vpn
   Linux eldivino 2.2.19 #6 Mon Jun 4 11:03:22 PDT 2001 i686 unknown
   No mail.
   vpn@eldivino:~$
   ```

   In questo modo ci si è appena autenticati con RSA. Bisogna tenere segreta la propria chiave privata (~root/.ssh/identity file sulla macchina client). Chiunque abbia accesso a questo file può collegarsi all'account vpn sul server.

4.3. Configurare sudo

pppd richiede di essere avviato come root. Comunque, sul server, stiamo avviando tutto come utente "vpn". Come può l'utente vpn eseguire pppd?

Ci sono molti modi di risolvere questo problema. Uno è utilizzare il bit suid, ed impostare i permessi per i gruppi. Però questo può condurre a confusione e difficoltà nell'amministrare agevolmente, creando falle non intenzionali nella sicurezza del sistema. Personalmente, ho trovato che l'utility sudo sia la soluzione migliore.

sudo dà agli utenti ordinari i privilegi del superutente, ma solo per una serie molto limitata di comandi. L'amministratore di sistema può decidere quali comandi sono consentiti e quanto log produrre. Nel nostro caso, vogliamo consentire all'utente "vpn" di avviare pppd con i privilegi di superutente, ma non vogliamo consentirgli di fare altro.

1. Occorre editare il file di configurazione di sudo, /etc/sudoers. Per impostare le restrizioni adeguate, in modo da prevenire inconvenienti e corse critiche, si usi il comando visudo che permette di editare /etc/sudoers. Chi non ha familiarità con vi, può consultare VIM HOWTO (http://www.linuxdoc.org/HOWTO/Vim-HOWTO.html).

   ```
   server# visudo
   ```

   Aggiungere queste due righe alla fine del file:

   ```
   Cmnd_Alias VPN=/usr/sbin/pppd
   vpn ALL=NOPASSWD: VPN
   ```

2. Ora, occorre verificare che sudo sia configurato correttamente. Per farlo basta provare a lanciare, come utente "vpn", pppd utilizzando sudo:

   ```
   server# su - vpn
   server$ sudo /usr/sbin/pppd noauth
   ~9}#ÀZ}!}!} }9}"}k} }r} }'}%}zt2-·}'}"}
   ```

   Se si ottiene un mucchio di spazzatura PPP sullo schermo (come l'ultima linea riportata sopra), questo è un risultato positivo. Significa che l'utente vpn è autorizzato a lanciare pppd. Ora è possibile andare su un altro terminale e terminare il processo, oppure si può aspettare che pppd termini l'esecuzione. Dovrebbe smettere di provare a connettersi dopo circa 30 secondi.

   Invece, se si ottiene "bash: /usr/sbin/pppd: Permission denied" o qualche altro tipo di errore, o chiede una password, allora sudo probabilmente non funziona. Occorrerà cercare di trovare cosa non va. Si verifichi che pppd sia in /usr/sbin e che il file sudoers sia stato compilato correttamente.

5. Configurazione del Client

Se ppp e ssh sono configurate sul client, e il server è pronto per la connessione, allora tutto quello che dovete fare sul client è di creare lo script per attivare il link.

5.1. Installare lo Script

La connessione VPN può essere inizializzata utilizzando il seguente script vpn-pppssh.

1. Salvate questo file sul client (non è importante dove, ad esempio /usr/local/bin/vpn-pppssh) e rendetelo eseguibile attraverso il comando "chmod a+x vpn-pppssh".
2. Impostate i campi all'inizio del file con i valori che avete deciso in la Sezione 3.3.

Ricordate che viene eseguito attraverso una shell bash, per cui dovete evitare di inserire spazi tra i segni di uguale, utilizzate le virgolette dove necessario e i metacaratteri di escape come $. Per maggiori dettagli leggete BASH Programming Introduction (http://www.linuxdoc.org/HOWTO/Bash-Prog-Intro-HOWTO.html) o Advanced Bash Scripting Guide (http://www.linuxdoc.org/LDP/abs/html/index.html).

```
SERVER_HOSTNAME=eldivino.domain.com
SERVER_USERNAME=vpn
SERVER_IFIPADDR=192.168.3.2
CLIENT_IFIPADDR=192.168.3.1
```

Eseguite "vpn-pppssh config" per visualizzare una lista delle possibili configurazioni. In questo modo potete essere certi che le vostre impostazioni siano state interpretate correttamente.

5.2. Lo Script vpn-pppssh

Qui trovate lo script vpn-pppssh. È composto da una sola linea di codice (quella che comincia con "PPPD" nel paragrafo iniziale). Tutto il resto del file è solamente codice di supporto, non indispensabile.

```
#!/bin/sh
# /usr/local/bin/vpn-pppssh
#
# This script initiates a ppp-ssh vpn connection.
# see the VPN PPP-SSH HOWTO on http://www.linuxdoc.org for more information.
#
# revision history:
# 1.6 11-Nov-1996 miquels@cistron.nl
# 1.7 20-Dec-1999 bart@jukie.net
# 2.0 16-May-2001 bronson@trestle.com


#
# You will need to change these variables...
#


# The host name or IP address of the SSH server that we are
# sending the connection request to:
SERVER_HOSTNAME=eldivino.domain.com

# The username on the VPN server that will run the tunnel.
# For security reasons, this should NOT be root.  (Any user
# that can use PPP can intitiate the connection on the client)
SERVER_USERNAME=vpn

# The VPN network interface on the server should use this address:
SERVER_IFIPADDR=192.168.3.2

# ...and on the client, this address:
CLIENT_IFIPADDR=192.168.3.1
```

```
# This tells ssh to use unprivileged high ports, even though it's
# running as root.  This way, you don't have to punch custom holes
# through your firewall.
LOCAL_SSH_OPTS="-P"


#
# The rest of this file should not need to be changed.
#



PATH=/usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/bin/X11/:

#
# required commands...
#

PPPD=/usr/sbin/pppd
SSH=/usr/bin/ssh

if ! test -f $PPPD  ; then echo "can't find $PPPD";  exit 3; fi
if ! test -f $SSH   ; then echo "can't find $SSH";   exit 4; fi


case "$1" in
  start)
    # echo -n "Starting vpn to $SERVER_HOSTNAME: "
    ${PPPD} updetach noauth passive pty "${SSH} ${LOCAL_SSH_OPTS}
${SERVER_HOSTNAME} -l${SERVER_USERNAME} -o Batchmode=yes sudo ${PPPD}
nodetach notty noauth" ipparam vpn ${CLIENT_IFIPADDR}:${SERVER_IFIPADDR}
    # echo "connected."
    ;;

  stop)
        # echo -n "Stopping vpn to $SERVER_HOSTNAME: "
        PID=`ps ax | grep "${SSH} ${LOCAL_SSH_OPTS} ${SERVER_HOSTNAME}
-l${SERVER_USERNAME} -o" | grep -v ' passive ' | grep -v 'grep ' | awk
'{print $1}'`
        if [ "${PID}" != "" ]; then
          kill $PID
          echo "disconnected."
        else
          echo "Failed to find PID for the connection"
        fi
    ;;

  config)
    echo "SERVER_HOSTNAME=$SERVER_HOSTNAME"
    echo "SERVER_USERNAME=$SERVER_USERNAME"
    echo "SERVER_IFIPADDR=$SERVER_IFIPADDR"
    echo "CLIENT_IFIPADDR=$CLIENT_IFIPADDR"
  ;;
```

```
  *)
    echo "Usage: vpn {start|stop|config}"
    exit 1
    ;;
esac

exit 0
```

6. Attivare il Link

Adesso tutto dovrebbe essere impostato. Fate un respiro profondo e provate ad attivare il collegamento.

1. Diventate root sulla macchina client ed eseguite lo script vpn-pppssh.

   ```
   client# /usr/local/bin/vpn-pppssh start
   ```

2. Ci metterà un momento a connettersi, poi dovrebbe visualizzare qualcosa di simile a

   ```
   Using interface ppp1
   Connect: ppp1 <--> /dev/pts/1
   local  IP address 192.168.3.1
   remote IP address 192.168.3.2
   ```

3. Ha funzionato? Prima di tutto provate a inviare un ping all'interfaccia VPN del client:

   ```
   client$ ping 192.168.3.2
   ```

4. Se ha funzionato, potete raggiungere l'interfaccia del client. Non entusiasmatevi troppo ancora: questa era la parte facile. Adesso, provate a inviare un ping verso l'interfaccia VPN del server:

   ```
   client$ ping 192.168.3.1
   ```

   Se vi risponde, allora congratulazioni! La vostra PPP-SSH VPN sembra essere attiva. I pacchetti stanno transitando con successo in entrambe le direzioni. Ora potreste provare a effettuare un login nel client e a inviare un ping al client dal server, ma arrivati fino a questo punto è praticamente sicuro che funzioni.

Potete interrompere VPN attraverso "vpn-pppssh stop".

Adesso che il tunnel funziona, potete integrarlo all'interno del vostro sistema in modo che si attivi automaticamente come descritto in la Sezione 7. Inoltre, se volete inviare dei pacchetti da un'intera sottorete attraverso il link (piuttosto che solo i pacchetti creati dal client e dal server come avete configurato adesso) leggete la Sezione 8.

6.1. Difficoltà e problemi

Lo script stesso è abbastanza semplice. L'intero sistema, comunque, coinvolge molte piccole parti. Se anche solo una di esse non è configurata, può impedire che la vostra VPN funzioni, senza nemmeno messaggi che spieghino il perché. Questa è una lista di cose da controllare se ci si trova in difficoltà:

- Fare un doppio o triplo controllo sui propri parametri di rete. Si provi a lanciare "vpn-pppssh config" per assicurarsi che la configurazione sia corretta e che la shell non abbia alterato alcuni valori.
- Ripercorrere ogni passo e assicurarsi che tutto sia a posto.
- Provare temporaneamente a spegnere qualsiasi firewall sul client, sul server, e sulle macchine intermedie in modo da vedere se qualcuno di questi sta bloccando il traffico (è improbabile che sia questa la causa se si può utilizzare SSH tra le due macchine).
- Assicurarsi che le proprie route siano corrette. Si può avere una lista delle proprie route usando "route -n". Visitare Linux Network Administrators Guide (http://www.linuxdoc.org/LDP/nag2/index.html) e http://www.linuxdoc.org/HOWTO/Adv-Routing-HOWTO.html per maggiori informazioni.

6.1.1. sendto: Operation not permitted

Quando si cerca di fare ping alle interfacce VPN, se si ottiene l'errore "sendto: Operation not permitted", si sta probabilmente incappando in un firewall sulla macchina locale che scarta i pacchetti prima che giungano alla interfaccia di rete. Il firewall deve consentire il traffico SSH sulla rete normale *e* deve fare passare tutto il traffico sull'interfacia VPN.

I comandi per ipchains che creano una apertura nel firewall per l'interfaccia PPP assomiglieranno ai seguenti:

```
ipchains -I input  1 -i ppp1 -s 192.168.3.0/24 -j ACCEPT
ipchains -I output 1 -i ppp1 -d 192.168.3.0/24 -j ACCEPT
```

Ovviamente, ppp1 deve essere l'interfaccia di rete della VPN PPP-SSH, e l'indirizzo IP deve coincidere con l'indirizzo dell'interfaccia locale. Occore controllare che il traffico sia consentito sia sul client che sul server.

Consultare Linux Firewall HOWTO (http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html), IPChains HOWTO (http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html) per il kernel 2.2, o la documentazione di iptables per il kernel 2.4.

7. Integrazione della VPN nel proprio sistema

Avviare il collegamento manualmente risulta laborioso dopo un po'. Probabilmente si desidera che la VPN venga creata all'avvio del computer o quando si instaura una connessione dial-up.

7.1. Connessione all'avvio del sistema

È piuttosto semplice fare in modo che questo script venga eseguito al momento dell'avvio. Considereremo di impiegare la comune configurazione System V initscript. Se non è questo il caso, si dovrà individuare da soli un modo diverso per integrare lo script nel proprio sistema.

1. Si copi o si crei un link simbolico di vpn-pppssh in /etc/init.d.

   ```
   cp /usr/local/bin/vpn-pppssh /etc/init.d/vpn-pppssh
   ```

2. Togliere il segno di commento alle righe echo presenti nelle funzioni start e stop dello script vpn-pppssh per abilitare i messaggi "Starting" e "done" all'avvio.
3. Inserire "> /dev/null 2>&1" dopo la linea che inizia con "${PPPD}" nella sezione start dello script. Questo impedisce che i lunghi messaggi di pppd rendano confusa la schermata di avvio. È possibile anche ridirigere i messaggi di pppd (che possono includere informazioni sugli errori) in un file di log o, se non si hanno pretese di tipo estetico, si può ignorarli e lasciare lo schermo confuso.
4. A questo punto si può creare semplicemente un link al proprio script in ciascuno dei sei runlevel.

   ```
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc0.d/K10vpn-pppssh
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc1.d/K10vpn-pppssh
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc2.d/S99vpn-pppssh
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc3.d/S99vpn-pppssh
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc4.d/S99vpn-pppssh
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc5.d/S99vpn-pppssh
   client$ ln -s /etc/init.d/vpn-pppssh /etc/rc6.d/K10vpn-pppssh
   ```

Adesso, quando riavviate la macchina, la vpn dovrebbe attivarsi verso la fine del processo di boot. Quando viene raggiunto questo script, la macchina attenderà che la VPN sia attiva prima di continuare il boot. Se questo è un problema, potete scrivere il vostro script /etc/init.d/vpn-pppssh che chiama lo script /usr/local/bin/vpn-pppssh in background. Il collegamento sarà attivo non appena la macchina avrà finito il boot.

Per attivare e disattivare manualmente il collegamento, basta eseguire lo script vpn-pppssh direttamente da /etc/init.d:

```
client$ /etc/init.d/vpn-pppssh stop
client$ /etc/init.d/vpn-pppssh start
```

7.2. Connettersi attraverso Dial-Up

Se vi state collegando ad internet tramite PPP, potete attivare la VPN ogni volta che attivate la connessione. Questo non è difficile, ma richiede una versione recente di pppd, che supporti sia l'opzione ipparam che le directory ip-up.d e ip-down.d.

1. Create il file "vpn-up" in /etc/ppp/ip-up.d:

   ```
   #!/bin/sh

   if [ "$PPP_IPPARAM" = "vpn" ]; then
           # Don't bring up the vpn if we're bringing up the vpn.
           exit 0
   fi

   /usr/local/bin/vpn start
   ```

L'espressione if gestisce la ri-invocazione. Se si sta creando il normale collegamento PPP, si desidera che anche la VPN venga avviata. Ma la VPN stessa è un collegamento PPP! Se non ne venisse tenuto conto, PPP verrebbe istanziato ricorsivamente fino a condurre la macchina ad un arresto.

Il segreto sta nel parametro "ipparm vpn" all'interno dello script vpn-pppssh. Esso imposta la variabile IPPARAM per la corrente invocazione di "vpn", che poi verrà controllata nello script di avvio. Se è impostata a vpn, sappiamo di avere già l'avvio della vpn in corso, quindi si esce senza riportare errori. In caso contrario si esegue l'avvio.

2. Se si desidera creare una apertura nel proprio firewall appositamente per l'avvio della VPN, è sufficiente creare un file /etc/ppp/ip-up.d/vpn-fw con il seguente contenuto. Tutte le variabili della shell indicate vengono fornite da pppd, quindi dovrebbe essere possibile utilizzare questo script senza modifiche.

```
#!/bin/sh

# Punch a hole in the firewall for the VPN

if [ "$PPP_IPPARAM" = "vpn" ]; then
        ipchains -I input 1 -i $PPP_IFACE -s $PPP_REMOTE -d $PPP_LOCAL -j ACCEPT
        ipchains -I output 1 -i $PPP_IFACE -s $PPP_LOCAL -d $PPP_REMOTE -j ACCEPT
fi
```

3. Si crei il file "vpn-down" in /etc/ppp/ip-down.d/vpn-pppssh:

```
#!/bin/sh

if [ "$PPP_IPPARAM" = "vpn" ]; then
        # Don't bring down the VPN if we're bringing down the vpn.
        exit 0
fi

/usr/local/bin/vpn stop
```

Tutti gli script citati devono essere resi eseguibili (chmod a+x /etc/ppp/ip-up.d/vpn-pppssh). A questo punto, quando si avvia il collegamento PPP, contemporaneamente dovrebbe essere avviata la VPN. Al momento di arrestarlo la VPN scomparirà. È facilissimo.

8. Inoltro tra subnet

La lettura di questa sezione è necesssaria solo se si sta cercando di connettere intere reti, non solo singoli host. Si assume che il tunnel da host a host sia già funzionante, al punto che computer client e server possano scambiarsi dei ping senza problemi. A questo punto è necesario consentire anche alle reti connesse alle macchine client e server di utilizzare il tunnel.

8.1. Inoltro

Come prima cosa, ci si deve assicurare che ai pacchetti venga consentito l'inoltro attraverso le interfacce di rete. Si può attivare questa opzione attraverso l'interfaccia di configurazione proc. È opportuno fare ciò al momento dell'avvio, ma si può anche inserirlo nello script vpn-pppssh, o perfino creare uno script nella directory /etc/init.d/ip-up.d (si veda la Sezione 7.2).

```
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/conf/ppp1❶/forwarding
```

❶ Ovviamente è necessario sostituire a ppp1 l'interfaccia corretta (quella associata a SERVER_IFIPADDR o a CLIENT_IFIPADDR, a seconda che si stia operando sul server o sul client).

8.2. Rendersi gateway

Su tutti i computer della sottorete, si deve impostare l'host locale della VPN come il gateway per tutte le reti situate dall'altra parte del tunnel. Questo dice ai computer "Se avete pacchetti destinati all'estremo opposto della VPN, inviateli all'host locale della VPN". Si tenga presente che se l'host VPN è già il gateway predefinito di tutti i computer non occorre fare altro: i pacchetti saranno inoltrati automaticamente.

Nell'esempio riportato sotto, il mio host VPN ha indirizzo IP 192.168.1.1 sulla rete locale e indirizzo IP 192.168.3.2 sulla rete VPN. Quest'ultima, contenente le interfacce VPN del client, del server e di tutti i computer sul lato opposto del collegamento, ha indirizzo 192.168.3.0/24. Qundi, su ogni computer locale a cui si desidera consentire di inviare pacchetti attraverso la VPN è necessario eseguire il seguente comando:

```
# route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.1
```

Ora, qualsiasi pacchetto destinato alla rete 192.168.3.0/24 da questa macchina verrà consegnato all'host 192.168.1.1 sulla rete locale per l'inoltro attraverso la VPN.

8.3. Routing

Non dovrebbe essere necessario impostare un routing personalizzato: pppd si occupa di tutto. Tuttavia, se trovate che pppd non soddisfi completamente i vostri bisogni, i campi per personalizzare il routing si trovano all'interno dello script vpn-pppssh. Per cambiare il routing sul client, eseguite semplicemente route. Per cambiare il routing sul server utilizzate ssh, mandando i comandi necessari. Qui potete trovare un esempio:

```
route add -net $NET1 gw $SERVER_IFIPADDR
ssh -o Batchmode=yes $SERVER_HOSTNAME -l$SERVER_USERNAME route add -net $NET2 gw $CLIENT_IFIPADDR
```

8.4. Mascheramento

È persino possibile configurare uno o entrambi gli host per mascherare tutte le connessioni attraverso il tunnel vpn. Consultare IP Masquerade HOWTO (http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO.html) per ulteriori dettagli.

```
# ipchains -A forward -i ppp1 -s 192.168.0.0/24 -j MASQ
```

8.5. E ora proviamolo

Questo dovrebbe essere tutto quello che serve per inviare pacchetti all'altra macchina da una sottorete connessa al client o al server. La vostra VPN PPP-SSH potrà lavorare silenziosamente e in maniera affidabile negli anni a venire.