Secure POP via SSH mini-HOWTO

Manish Singh, <*yosh@gimp.org*> v1.0, 30 Settembre 1998

Questo documento spiega come impostare delle connessioni POP sicure usando SSH.

Contents

1 Introduzione

Le normali sessioni di posta POP, per la loro vera natura, sono insicure. Le password viaggiano attraverso la rete in chiaro e tutti le possono vedere. Ora, questo può essere perfettamente accettabile in un ambiente fidato o dietro a un firewall. Ma su una rete pubblica, come un'università o il vostro consueto ISP, chiunque armato di un semplice "annusatore di rete" può arraffare immediatamente la vostra password dal filo. Questo è comprovato dal fatto che molti impostano il loro computer per controllare la posta a intervalli regolari, cosicché la password è trasmessa davvero di frequente, questo rende facile "annusarla".

Con questa password, un aggressore può ora accedere al vostro account di posta che potrebbe contenere informazioni delicate o private. È anche del tutto comune che la password sia la stessa che usate per la shell utente del vostro account così c'è la possibilità per danneggiarvi di più.

Facendo in modo che tutto il traffico POP usi un canale criptato, **niente** gira in chiaro sulla rete. Possiamo usare i diversi metodi di autenticazione SSH, anziché usare una semplice password in plain/text. Questo è lo scopo reale per usare questo metodo, non perché cripta il contenuto (che è inutile a questo punto, poiché probabilmente esso è ormai stato inviato non criptato su molte reti prima di raggiungere la vostra casella di posta; rendere sicure queste comunicazioni è compito del GNU Privacy Guard o PGP, non di SSH), ma perché consente l'autenticazione sicura.

Ci sono altri modi ormai per ottenere l'autenticazione sicura, quali APOP, KPOP e IMAP. Tuttavia usando SSH si ha il vantaggio che funzioni con la normale configurazione POP, senza richiedere client speciali (non tutti i client di posta supportano i protocolli avanzati) o supporti sul server (eccetto che sshd sia in esecuzione sul server). Il vostro server di posta potrebbe non comprendere o non essere configurato per usare un protocollo più sicuro. Per di più, usando SSH potete comprimere ancora di più il traffico, e questo è un piccolo grazioso extra per gli utenti con connessioni lente.

2 La tecnica di base

Questa tecnica fa affidamento su una caratteristica fondamentale di SSH: il *port forwarding* (l'inoltro di una porta)

Ci sono molte variazioni su questo tema che dipendono dalle configurazioni di posta che desiderate. Tutto questo richiede SSH, che è disponibile su *http://www.ssh.fi/* e mirror. i pacchetti RPM si possono trovare su *ftp://ftp.replay.com/pub/crypto/*

e i pacchetti Debian su *ftp://non-us.debian.org/debian-non-US/*

(e sui loro rispettivi mirror).

2.1 Impostare il Port Forwarding

Per iniziare il port forwarding, lanciate il seguente comando:

```
ssh -C -f popserver -L 11110:popserver:110 sleep 5
```

Esaminiamo il comando:

`ssh`

Il binario ssh stesso, il magico programma che fa tutto questo.

`-C`

Questo abilita la compressione del flusso di dati. Esso è opzionale, ma generalmente utile, specialmente per gli utenti dialup.

`-f`

Una volta che SSH ha dato l'autenticazione e ha stabilito il port forwarding, spostatelo in background così gli altri programmi potranno funzionare. Da quando stiamo usando le proprietà del port forwarding di SSH, non abbiamo più bisogno di assegnare ad esso una tty.

`popserver`

Il server POP cui ci stiamo connettendo.

`-L 11110:popserver:110`

Inoltra la porta locale 11110 alla porta 110 sul server remoto `popserver`. Noi usiamo una porta locale alta (11110) così ogni utente può effettuare l'inoltro.

`sleep 5`

Dopo che SSH ha spostato se stesso in background, esso avvia un comando. Noi usiamo `sleep` affinché la connessione sia mantenuta per un tempo sufficiente per consentire al nostro client di posta di instaurare la connessione al server. 5 secondi sono solitamente un tempo sufficiente perché questo succeda.

Potete usare molte altre opzioni per SSH quando appropriato. Un'impostazione frequente può essere il nome utente (username), poiché potrebbe essere diverso sul server POP.

Questo *richiede* che sshd sia in esecuzione sul server remoto `popserver`. Tuttavia non avete bisogno di avere li un account con una shell attiva. Il tempo che si prende per stampare il messaggio "You cannot telnet here" (non potete collegarvi in telnet qui) è sufficiente per impostare la connessione.

2.2 Testarlo

Una volta che avete compreso minuziosamente i comandi da avviare per stabilire il port forwarding, potete provarlo. Per esempio:

```
$ ssh -C -f msingh@popserver -L 11110:popserver:110 sleep 1000
```

`popserver` è il vostro server POP. Il mio nome utente sulla mia macchina locale è `manish` così ho bisogno di specificare esplicitamente il nome utente `msingh`. (Se il vostro nome utente locale è uguale a quello remoto, la parte `msingh@` non è necessaria.

Poi esso stampa:

```
msingh@popserver's password:
```

E digito li la mia password POP (potete avere password diverse per la shell e per il POP oppure usare solo quella della shell). Abbiamo finito! Così possiamo provare:

```
$ telnet localhost 11110
```

Il quale potrebbe stampare qualcosa del tipo:

```
QUALCOMM POP v3.33 ready.
```

Woohoo! Funziona! I dati sono trasmessi sulla rete criptati, cosicché il solo testo in chiaro è sull'interfaccia di loop-back della mia macchina locale e del server POP.

3 Usarla con il vostro software di posta.

Questa sezione descrive come impostare il software del vostro client POP per usare la connessione SSH inoltrata. Il suo obiettivo principale è fetchmail (un'eccellente utility ESR per l'inoltro e la consegna), poiché è il software più flessibile che ho trovato per la distribuzione con il POP. Fetchmail può essere trovato su *http://www.tuxedo.org/~esr/fetchmail/* . Questo vi darà l'occasione per leggere l'eccellente documentazione distribuita con fetchmail.

3.1 Impostare fetchmail

Il seguente è il mio `.fetchmailrc`

```
defaults
        user msingh is manish
        no rewrite

poll localhost with protocol pop3 and port 11110:
        preconnect "ssh -C -f msingh@popserver -L 11110:popserver:110 sleep 5"
        password foobar;
```

Molto semplice, vero? fetchmail è ricco di comandi, ma quelli più importanti sono la riga `preconnect` e l'opzione `poll`.

Non ci connettiamo direttamente con il server POP, ma con l'host locale (localhost) e la porta 11110. La `preconnect` da l'inoltro ogni volta che si avvia fetchmail, lasciando aperta la connessione per 5 secondi, così fetchmail può instaurare la sua connessione. Il resto lo fa fetchmail stesso.

Ogni volta che avviate fetchmail, siete incitati a dare la vostra password per l'autenticazione SSH. Se avviate fetchmail in background (come ho fatto io), avrete l'inconveniente di dover fare questo. Questo ci porta alla prossima sezione.

3.2 Automatizzare il tutto

SSH può autenticare usando molti metodi. Uno di questi è l'utilizzo di una coppia di chiavi pubblica/privata RSA. Potete generare una chiave d'autenticazione per il vostro account usando `ssh-keygen`. Una chiave d'autenticazione può avere una passphrase (frase password) associata ad essa, oppure la passphrase può essere vuota. Il volere o no una passphrase dipende da quanto pensiate che sia sicuro l'account che state usando localmente.

Se voi pensate che la vostra macchina sia sicura, proseguite e avrete una passphrase vuota. Poi il su riportato `.fetchmailrc` lavora appena avviate fetchmail. In seguito potete far funzionare fetchmail in modalità demone quando voi chiamate e la posta è consegnata automaticamente. Avete fatto.

Tuttavia, se pensate di aver bisogno di una passphrase, le cose diventano più complicate. SSH può funzionare sotto il controllo di un **agente**, il quale può registrare le chiavi e autenticare qualsiasi connessione SSH fatta sotto di esso. Per questo motivo io ho questo script `getmail.sh`:

```
#!/bin/sh
ssh-add
while true; do fetchmail --syslog --invisible; sleep 5m; done
```

Quando chiamo, io avvio:

```
$ ssh-agent getmail.sh
```

Mi chiede la password una volta sola, poi controlla la posta ogni 5 minuti. Quando la connessione dialup viene chiusa, termino ssh-agent. (Questo è automatico nei miei script ip-up e ip-down)

3.3 Non usare fetchmail

E se non posso o non voglio usare fetchmail? Pine, Netscape e molti altri client hanno il loro proprio meccanismo POP. Primo, prendete in considerazione l'utilizzo di fetchmail! Esso è di gran lunga più flessibile e i client di posta non possono fare ogni volta tutte queste cose. Sia Pine che Netscape possono essere configurati per usare il sistema di posta locale.

Ma se proprio dovete farlo, a meno che il vostro client abbia una caratteristica di preconnessione come fetchmail, state andando nella direzione di dover mantenere attivo l'SSH port forwarding per l'intero tempo che state connessi. Questo significa usare `sleep 100000000` per mantenere attiva la connessione. Questo potrebbe non farvi andare troppo d'accordo con il vostro amministratore di rete.

Secondariamente, alcuni client (come Netscape) hanno il numero della porta preimpostato a 110, così avrete bisogno di essere root per fare il port forwarding da porte privilegiate. Questo è anche irritante. Ma dovrebbe funzionare.

4 Miscellanea

4.1 Liberatoria

Non ci sono garanzie che questo documento raggiunga lo scopo previsto. Questo è semplicemente fornito come risorsa libera. Per questo l'autore dell'informazione in esso contenuta non può dare alcuna garanzia in merito all'accuratezza della stessa. Usatela a vostro rischio e pericolo.

Il software per la crittografia, come SSH, può essere soggetto ad alcune restrizioni che dipendono dal luogo in cui vivete. In molti paesi dovete avere una licenza per usare tale software. Se non siete sicuri delle vostre leggi locali, per favore consultate qualcuno che abbia familiarità con la vostra situazione per ulteriori informazioni.

L'uso dell'informazione contenuta in questo documento spesso non è previsto dal vostro fornitore di servizi di rete. L'autore non incoraggia l'uso scorretto o l'abuso dei servizi di rete e fornisce questo documento a puro titolo informativo. Se avete dubbi sul fatto che queste tecniche cadano o meno entro l'accordo di servizio stipulato col vostro fornitore di servizi e-mail, per favore chiariteli, prima di metterle in pratica.

4.2 Copyright

This document is copyright © 1998 Manish Singh <*yosh@gimp.org*>

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that this copyright notice is included exactly as in the original, and that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions.

Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All trademarks used in this document are acknowledged as being owned by their respective owners.

4.3 Ringraziamenti

Uno speciale ringraziamento va a Seth David Schoen <*schoen@uclink4.berkeley.edu*> , che mi ha illuminato sulla via dell'ssh port forwarding.