Secure POP via SSH mini-HOWTO

Manish Singh, <*yosh@gimp.org*>
Vertaald door: Ellen Bokhorst, <*bokkie@nl.linux.org*>
v1.0, 30 september 1998

In dit document wordt uitgelegd hoe veilige POP verbindingen op te zetten met behulp van ssh.

Inhoudsopgave

1 Introductie

Normale POP mailsessies zijn door hun aard onveilig. Het wachtwoord wordt over het netwerk in gewone tekst verzonden wat door iedereen kan worden gelezen. Nu is dit wellicht acceptabel in een vertrouwde of met firewall geconfigureerde omgeving, maar op een publiek netwerk, zoals een universiteit of je ISP, kan iedereen gewapend met een simpele netwerksniffer je wachtwoord direct van de lijn afvangen. Dit opgeteld bij het feit dat veel mensen hun computers zo instellen dat regelmatig op mail wordt gecontroleerd waarbij het wachtwoord tamelijk frequent wordt verzonden, maakt het makkelijk te sniffen.

Met dit wachtwoord kan een aanvaller nu je emailaccount benaderen, waardoor hij wellicht aan gevoelige of privé-informatie kan komen. Het is tevens tamelijk gebruikelijk dat dit wachtwoord hetzelfde is als het wachtwoord van de shellaccount van de gebruiker, dus een risico op nog meer schade.

Door alle POP-verkeer via een versleuteld kanaal te laten gaan, gaat er **niets** in gewone tekst over het netwerk. We kunnen de diverse authenticatiemethoden van ssh gebruiken in plaats van een simpel wachtwoord in gewone tekst. Dat is de werkelijke reden voor het gebruik van deze methode: niet omdat we versleutelde inhoud krijgen (wat nutteloos is op dit punt, aangezien het waarschijnlijk onversleuteld via verscheidene netwerken is gegaan nog voordat het je mailbox bereikt; het beveiligen van die communicatie is de taak van GNU Privacy Guard of PGP, niet ssh), maar de veilige authenticatie.

Er zijn reeds andere methoden om een veilige authenticatie te bereiken, zoals APOP, KPOP, en IMAP. Het gebruik van ssh heeft echter als voordeel dat het met normale POP configuraties werkt, zonder dat daar een speciale client voor nodig is (niet alle mailclients ondersteunen geavanceerde protocollen) of serverondersteuning (behalve dat sshd op de server moet draaien). Je mailprovider kan wellicht niet in staat of niet bereid zijn om een veiliger protocol te gebruiken. Bovendien kun je door gebruik te maken van ssh het verkeer ook comprimeren, wat weer een klein extraatje is voor mensen met langzame verbindingen.

2 De basistechniek

Deze techniek gaat af op een basisfeature van ssh: *port forwarding*

Er zijn veel variaties op dit thema, wat afhangt van je gewenste mailsetup. Voor allen is ssh nodig, wat beschikbaar is vanaf *http://www.ssh.fi/* en mirrors. RPM's zijn beschikbaar op *ftp://ftp.replay.com/pub/crypto/* en Debian packages zijn beschikbaar op *ftp://non-us.debian.org/debian-non-US/* (en respectieve mirrors).

2.1 Opzetten van Port Forwarding

Voer de volgende opdracht uit om port forwarding op te starten:

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

Laten we die opdracht eens wat nader bekijken:

`ssh`

De ssh binary zelf, het magische programma dat het allemaal doet.

`-C`

Hiermee wordt de compressie van de datastroom geactiveert. Het is optioneel, maar gewoonlijk nuttig, vooral voor dialup gebruikers.

`-f`

Zodra ssh de authenticatie heeft uitgevoerd, en de port forwarding tot stand heeft gebracht, plaatst het zichzelf in de achtergrond zodat andere programma's kunnen worden uitgevoerd. Aangezien we slechts de port forwarding features van ssh gebruiken, hoeft er geen tty aan te zijn gekoppeld.

`popserver`

De POP-server waar we een verbinding mee maken.

`-L 11110:popserver:110`

Forward de lokale poort 11110 naar poort 110 op de remote server `popserver`. We gebruiken een hoge lokale poort (11110) zodat elke gebruiker forwardings kan aanmaken.

`sleep 5`

Nadat ssh zichzelf in de achtergrond heeft geplaatst, voert het een opdracht uit. We gebruiken `sleep` zodat de verbinding lang genoeg blijft onderhouden voor onze mailclient om een verbinding met de server te kunnen maken. 5 seconden is hiervoor gewoonlijk genoeg.

Wanneer van toepassing kun je gebruik maken van de meeste andere opties van ssh. Een gebruikelijke instelling kan een gebruikersnaam zijn, aangezien het een andere gebruikersnaam kan zijn dan die op de POP-server.

Hiervoor *moet* sshd op de remote server `popserver` draaien. Je hoeft er echter geen actief shellaccount te hebben. De tijd dit het in beslag neemt de melding "You cannot telnet here" is voldoende om een verbinding op te zetten.

2.2 Testen

Zodra je de details van de opdracht hebt uitgezocht om een port forwarding tot stand te brengen, kun je het uitproberen. Bijvoorbeeld:

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

`popserver` is de oude POP server. Mijn gebruikersnaam op mijn lokale machine is `manish` dus moet ik expliciet de gebruikersnaam `msingh` opgeven. (Als de gebruikersnaam op de lokale en remote computer gelijk zijn dan is het deel `msingh@` onnodig.

Dan wordt er afgedrukt:

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

En vervolgens typ ik mijn POP-wachtwoord in (je hebt voor je shell en POP misschien verschillende wachtwoorden dus gebruik die van je shell). Nu zijn we klaar! Dus kunnen we proberen:

```
$ telnet localhost 11110
```

wat iets zou moeten afdrukken als:

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

Woohoo! Het werkt! De gegevens worden versleuteld over het netwerk verstuurd, dus de enige gewone tekst gaat over de loopbackinterfaces van mijn lokale box en de POP-server.

3 Gebruik het met je mailsoftware

In deze sectie wordt de instelling van je POP clientsoftware beschreven waarbij het gebruik maakt van de ssh forwarded connectie. Het is primair op fetchmail gericht (ESR's uitstekende mail-ophaal en forwarding utility), aangezien dat de meest flexibele software die ik voor het omgaan met POP heb gevonden. Fetchmail is te vinden op *http://www.tuxedo.org/~esr/fetchmail/*. Je verleent jezelf een grote dienst als je de uitstekende documentatie leest die met fetchmail wordt meegeleverd.

3.1 Opzetten van fetchmail

Hieronder volgt mijn `.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;
```

Tamelijk simpel, toch? fetchmail kent een rijkdom aan opdrachten, maar de belangrijkste zijn de `preconnect` regel en de optie `poll`.

We maken geen directe verbinding met de POP-server, maar in plaats daarvan met localhost en poort 11110. De `preconnect` doet de forwarding elke keer dat fetchmail wordt uitgevoerd, de verbinding 5 seconden open latend, zodat fetchmail zijn eigen verbinding kan opzetten. De rest doet fetchmail zelf.

Dus elke keer dat je fetchmail uitvoert, wordt er gevraagd om je ssh wachtwoord voor de authenticatie. Als je fetchmail in de achtergrond uitvoert (zoals ik doe), dan is dat niet handig. Wat ons brengt bij de volgende sectie.

3.2 Alles automatiseren

ssh kan met veel manieren authenticeren. Een hiervan is een RSA public/private sleutelpaar. Je kunt met `ssh-keygen` een authenticatiesleutel voor je account genereren. Er kan met een authenticatiesleutel een wachtwoord worden geassocieerd of het wachtwoord kan worden leeggelaten. Of je een wachtwoord wilt, hangt af van hoe veilig je meent dat het account is dat je lokaal gebruikt.

Als je denkt dat je machine veilig is, ga dan verder en laat het wachtwoord leeg. Dan werkt de bovenstaande `.fetchmailrc` door het slechts uitvoeren van fetchmail. Je kunt fetchmail dan in daemon modus uitvoeren wanneer je inbelt waarbij de mail automatisch wordt opgehaald. Je bent klaar.

Als je echter meent een wachtwoord nodig te hebben, dan wordt het wat complexer. ssh kan onder besturing van een **agent** worden uitgevoerd, die sleutels kan registreren en wat voor ssh verbindingen er ook onder worden gemaakt authenticeren. Dus maak ik gebruik van het volgende script `getmail.sh`:

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

Wanneer ik inbel, voer ik uit:

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

Eenmaal wordt om mijn wachtwoord gevraagd, vervolgens wordt elke 5 minuten mijn mail gecontroleerd. Wanneer de inbelverbinding wordt gesloten, beëindig ik de ssh-agent. (Dit is geautomiseerd in mijn ip-up en ip-down scripts).

3.3 Geen gebruik van fetchmail

Wat als ik geen gebruik kan/wil maken van fetchmail? Pine, Netscape en nog een aantal andere clients hebben eigen POP-mechanismen. Overweeg eerst het gebruik van fetchmail! Het is veel flexibeler, en mailclients zouden zich hoe dan ook toch niet met dergelijke zaken bezig moeten houden. Zowel Pine als Netscape kunnen worden geconfigureerd dat ze gebruik maken van lokale mailsystemen.

Maar als je erop staat, dan moet je de ssh port forware de gehele tijd dat je verbonden bent actief houden, tenzij je client een preconnect feature heeft zoals fetchmail. Wat betekent het gebruik van `sleep 100000000` om de verbinding te behouden. De systeembeheerders kunnen dit wellicht niet waarderen.

Ten tweede hebben een aantal clients (zoals Netscape) het poortnummmer hardcoded in 110. Dus wellicht dat je root moet zijn om port forwarding te gebruiken vanuit privileged poorten. Ook dit is ergerlijk. Maar het zou moeten functioneren.

4 Diversen

4.1 Disclaimer

Er is geen garantie dat dit document zijn bedoelde doel waarmaakt. Dit is simpelweg voorzien als een vrije bron. Als zodanig kan de auteur van de geleverde informatie geen garantie geven dat de informatie accuraat is. Gebruik het op eigen risico.

Cryptografische software zoals ssh kan afhankelijk van waar je woont zijn onderworpen aan bepaalde restricties. In een aantal landen moet je een licentie hebben om dergelijke software te kunnen gebruiken. Als je niet zeker bent van de plaatselijke wetten, raadpleeg dan alsjeblieft iemand die bekend is met je situatie voor meer informatie.

Het gebruik van de informatie in dit document is zeer waarschijnlijk niet voorzien door je mail service provider. De auteur moedigt het misbruik en onjuist gebruik van netwerkservices niet aan, en voorziet alleen voor informatieve doeleinden in dit document. Als je twijfelt of het gebruik van deze technieken binnen de service-overeenkomst van je mailprovider valt, zorg dan alsjeblieft dat je dat vantevoren helder krijgt.

4.2 Copyright

4.3 Erkenningen

Speciale dank gaat naar Seth David Schoen <*schoen@uclink4.berkeley.edu*>, who enlightened me in the ways of ssh port forwarding.