Gecomprimeerde TCP/IP-Sessies met op SSH lijkende tools

Sebastian Schreiber <Schreib@SySS.de>
Vertaald door: Ellen Bokhorst bokkie@nl.linux.org 2.2.2000

1 Introductie

In het verleden, gebruikten we gecomprimeerde bestanden om diskruimte te besparen. Tegenwoordig is diskruimte goedkoop - maar is bandbreedte beperkt. Door het comprimeren van gegevensstromen bereik je twee doelen:

1) Je bespaart op bandbreedte/transfered volume (dat is belangrijk als je voor verkeer moet betalen of als je netwerk is geladen).

2) Versnellen van low-bandwith connecties (Modem, GSM, ISDN).

In deze HowTo wordt uitgelegd hoe je op zowel bandbreedte als verbindingstijd kan beparen door gebruik te maken van tools als SSH1, SSH2, OpenSSH of LSH.

2 Comprimeren van HTTP/FTP,...

Mijn kantoor is verbonden met een 64KBit ISDN lijn naar het internet, dus de maximum transportsnelheid is ongeveer 7K/s. Je kunt de connectie versnellen door het te comprimeren: als ik bestanden download, toont Netscape een transportsnelheid tot aan 40K/s (Logbestanden zijn comprimeerbaar tot een factor 15). SSH is een tool welke hoofdzakelijk is ontworpen om veilige verbindingen tot stand te brengen over onbeveiligde netwerken. Bovendien is het met SSH mogelijk connecties te comprimeren en behoort port forwarding tot de mogelijkheden (zoals rinetd of redir). Dus is het de juiste tool voor het comprimeren van iedere eenvoudige TCP/IP-connectie. "Eenvoudig"betekent dat er slechts één TCP-connectie geopend is. FTP-connecties of een connectie tussen between M$-Outlook en MS-Exchange zijn niet eenvoudig aangezien er verscheidene verbindingen tot stand worden gebracht. SSH gebruikt het LempleZiv (LZ77) compressie algoritme - dus je zal dezelfde hoge compressie-verhouding bereiken als met winzip/pkzip. Om alle HTTP-connecties vanaf mijn intranet naar het internet te comprimeren, hoef ik op mijn dial-in computer slechts één opdracht uit te voeren:

```
ssh -l <login ID> <hostname> -C -L8080:<proxy_at_ISP>:80 -f sleep 10000
```

`<hostname>` = host bij mijn ISP. SSH-toegang is vereist.

`<login ID>` = mijn login-ID op `<hostname>`

`<proxy_at_ISP>` =de web proxy van mijn ISP

Mijn browser is zodanig geconfigureerd dat het `localhost:8080` als proxy gebruikt. Mijn laptop is verbonden met dezelfde socket. De verbinding wordt gecomprimeerd en door SSH naar de echte proxy doorgestuurd. De infrastructuur ziet er ongeveer zo uit:

```
                  64KBit ISDN
Mijn PC-----------------------------Een PC (Unix/Linux/Win-NT) bij mijn ISP
SSH-Client         gecomprimeerd     SSH-Server, Poort 22
```

```
Poort 8080                          |
|                                   |
|                                   |
|                                   |
|10MBit Ethernet                    |100MBit
|niet gecomprimeerd                 |niet gecomprimeerd
|                                   |
|                                   |
Mijn tweede PC                      ISP's WWW-proxy
met Netscape,...                    Poort 80
(Laptop)
```

3 Email comprimeren

3.1 Inkomende Emails (POP3, IMAP4)

De meeste mensen halen hun email vanaf de mailserver op via POP3. POP3 is een protocol met vele nadelen:

1. POP3 transporteert het wachtwoord in gewone tekst. (Er zijn SSL-implementaties van POP/IMAP en een challenge/response authenticatie, gedefinieerd in RFC-2095/2195).
2. POP3 veroorzaakt veel protocol overhead: ten eerste verzoekt de client om een bericht, dan stuurt de server het bericht op. Daarna verzoekt de client het getransporteerde artikel te verwijderen. De server bevestigt de verwijdering. Daarna is de server pas klaar voor de volgende transactie. Dus voor iedere email zijn 4 transacties nodig.
3. Alhoewel email hoog comprimeerbaar (factor=3.5) is, transporteert POP3 de mail ongecomprimeerd. (factor=3.5).

Je zou POP3 kunnen comprimeren door localhost:110 via een gecomprimeerde connectie naar je ISP's POP3-socket door te sturen. Daarna moet je je mail-client vertellen een verbinding te maken met localhost:110 om mail te downloaden. Dat beveiligt en versnelt de connectie – maar de downloadtijd ondergaat nog steeds de aan POP3 inherente protocol overhead.

Het heeft zin POP3 door een wat efficiënter protocol te vervangen. De gedachte hierachter is de gehele mailbox in één keer te downloaden zonder het genereren van de protocol overhead. Bovendien heeft het zin de connecties te comprimeren. De juiste tool welke beide mogelijkheden biedt is SCP. Je kunt je mailbestand als volgt downloaden:

`scp -C -l loginId:/var/spool/mail/loginid /tmp/newmail`

Maar er is een probleem: wat gebeurt er als een nieuwe email arriveert op de server tijdens het downloaden van je mailbox? De nieuwe mail zou verloren gaan. Daarom heeft het meer zin de volgende opdrachten te gebruiken:

`ssh -l loginid mailserver -f mv /var/spool/mail/loginid /tmp/loginid_fetchme`

`scp -C -l loginid:/tmp/my_new_mail /tmp/loginid_fetchme`

Een verplaatsing (mv) is een elementaire bewerking, dus je geraakt niet in moeilijkheden als je tijdens de uitvoering van de opdrachten nieuwe mail ontvangt. Maar als de directory's /tmp/ en /var/spool/mail op de mailserver niet op dezelfde disk staan, krijg je wellicht problemen. Een oplossing hiervoor is een lockbestand op de server aan te maken voordat je de mv uitvoert: `touch /var/spool/mail/loginid.lock`. Je zou het erna moeten verwijderen. Een betere oplossing is het bestand loginid naar dezelfde directory te verplaatsen:

```
ssh -l loginid mailserver -f mv /var/spool/mail/loginid /var/spool/mail/loginid_fetchme
```

Daarna kun je formail gebruiken in plaats van procmail om /tmp/newmail naar de juiste foler(s) te filteren:

```
formail -s procmail < /tmp/newmail
```

3.2 Uitgaande Email (SMTP)

Je stuurt email over gecomprimeerde en versleutelde SSH-connecties om te: to:

- Besparen op netwerkverkeer
- De connectie te beveiligen (Dit heeft geen zin als de mail later via niet te vertrouwen netwerken wordt getransporteerd).
- De authenticiteit van de zender te bevestigen. Veel mailservers weigeren het heruitzende van mail om misbruik te voorkomen. Als je email over een SSH-connectie zendt, denkt de remote mailserver (b.v. sendmail of MS-exchange) lokaal een verbinding te hebben.

Als je SSH-toegang op de mailserver hebt, heb je de volgende opdracht nodig:

```
ssh -C -l loginid mailserver -L2525:mailserver:25
```

Als je geen SSH-toegang tot de mailserver hebt, maar tot een server die het toestaat dat je je mailserver als relay gebruikt, is de opdracht:

```
ssh -C -l loginid other_server -L2525:mailserver:25
```

Daarna kun je je mailclient (of mailserver: zie "smarthost") zodanig configureren dat het mail naar de localhost poot 2525 stuurt.

4 Gedachten over de performance.

Uiteraard neemt de compressie/encryptie CPU-tijd. Het bleek dat een oude Pentium-133 ongeveer 1GB/uur kon versleutelen en comprimeren – dat is heel wat. Als je SSH met de optie "–with-none" compileert, kun je SSH laten weten geen encryptie te gebruiken. Dat bespaart wat van de performance. Hier is een vergelijking tussen verscheidene download methoden (tijdens de test, werd een ongecomprimeerd 6MB-bestand getransporteerd vanaf een 133MHz-Pentium-1 naar een 233MHz Pentium2 laptop over een 10MBit ethernet zonder andere load).

	FTP	versleuteld	gecompri- meerd	gecomprimeerd & versleuteld
Verstreken tijd	7.6s	26s	9s	23s
Doorvoer	790K/s	232K/s	320K/s	264K/s
Compressie Factor	1	1	3.8	3.8

5 Groeten

Met dank aan Harald König <koenig@tat.physik.uni-tuebingen.de>, wie rcp gebruikte om complete mailboxen te downloaden. De laatste versie van deze howto is beschikbaar op `http://www.syss.de/howto`.