SSH mit Public Key-Authentifizierung

Einleitung

SSH-Verbindungen über den Uni-Server `login.uni-freiburg.de` (beispielsweise zum Zugriff auf das Homeverzeichnis) verwenden normalerweise eine klassische Authentifizierung mit Benutzernamen (= Benutzerkennung) und Passwort. Für die meisten Anwendungen reicht dies aus, doch in einem Fall entsteht ein Problem: Will man den Zugriff auf diesen Server automatisieren – beispielsweise um automatisch Backups durchzuführen –, so muss man sein Uni-Account-Passwort auf dem heimischen Rechner im Klartext hinterlegen. Dass das keine gute Idee ist, dürfte offensichtlich sein.

Public Key-Authentifizierung

Glücklicherweise beherrscht SSH noch eine andere Möglichkeit, sich zu authentifizieren: sogenannte Schlüsselpaare mit Private und Public Key. Grob gesagt handelt es sich dabei um zwei sehr lange Zeichenketten, die als 'Identität' eines Anwenders fungieren. Der Public Key ist dabei, wie der Name schon sagt, öffentlich und kann auf jedem beliebigen Server, mit dem man sich verbinden will, platziert werden. Den Private Key muss man dagegen so geheim halten wie ein Passwort, deswegen ist unbedingt sicherzustellen, dass kein Anderer auf diesen zugreifen kann. Weitere Informationen zur Funktionsweise der Authentifizierung mit Schlüsselpaaren finden sich beispielsweise auf Wikipedia oder ausführlicher in Johannes Frankens SSH-Artikel.

> Aufgrund besonderer Sicherheitsvorkehrungen erfordert die Public-Key-Authentifizierung auf dem Login-Server der Uni (login.uni-freiburg.de), dass bei der Authentifizierung eine Login-Session bereits läuft oder kurz zuvor gelaufen ist. Somit ist ein kein vollautomatischer Zugriff auf den Server möglich, sondern es muss unmittelbar zuvor einmalig ein normaler Login mit Eingabe des Passworts erfolgen.

Schlüsselpaar generieren

> Die folgende Anleitung bezieht sich ausschließlich auf Unix-Systeme. Analoge Anleitungen für Windows und Mac sollten sich aber im Netz finden lassen. Es werden Grundkenntnisse der Shell vorausgesetzt und Sie sollten den normalen SSH-Login schon einmal verwendet haben.

Die Erzeugung eines Schlüsselpaars übernimmt das Kommando `ssh-keygen`. Ohne Parameter aufgerufen, fragt es nach einigen Daten, die entsprechend einzugeben sind:

```
$ ssh-keygen
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/jannis/.ssh/id_rsa):
/home/jannis/.ssh/uni # hier möglichst aussagekräftigen Dateinamen wählen
Enter passphrase (empty for no passphrase): # Hier <Enter> drücken
Enter same passphrase again: # hier ebenfalls
```

Danach sollte folgende Erfolgsmeldung ausgegeben werden:

```
Your identification has been saved in /home/jannis/.ssh/uni.
Your public key has been saved in /home/jannis/.ssh/uni.pub.
```

Den Rest der Ausgabe können Sie ignorieren.

> Erhält ein Angreifer Zugriff auf die Datei `uni`, so kann er sich ebenso wie Sie ohne Passwortabfrage auf dem Uni-Server einloggen. Schützen Sie diese Datei also so gut wie irgend möglich!

Öffentlichen Schlüssel hochladen

Nun müssen Sie den öffentlichen Schlüssel auf dem Server ablegen, damit der Ihre Identität überprüfen kann. Dazu bietet sich beispielsweise das Programm `sftp` an. (Möglicherweise ist dieses bei Ihnen nicht vorinstalliert, es sollte sich aber in den Paketquellen finden lassen.)

Um sich mit dem Server zu verbinden, gehen Sie analog zu SSH vor, wobei `UserID` durch Ihre Benutzerkennung zu ersetzen ist:

```
$ sftp UserID@login.uni-freiburg.de
$ UserID@login.uni-freiburg.de's password: # hier Uni-Account-Passwort
eingeben
$ sftp>
```

Nun sehen Sie die sftp-Eingabeaufforderung. Wechseln Sie hier lokal in das Verzeichnis, in dem sich der generierte Public Key befindet:

```
sftp> lcd /home/jannis/.ssh/
```

Nun wechseln Sie auf dem Server in das Verzeichnis, in dem der Schlüssel abgelegt werden soll:

```
sftp> cd /home/UserID/.ssh # UserID wieder ersetzen
```

Und schließlich können Sie ihn hochladen und die sftp-Sitzung schließen.

> Achten Sie darauf, dass Sie die Datei mit der Endung `.pub` hochladen, nicht die Datei ohne Endung!

```
sftp> put uni.pub
Uploading uni.pub to /home/UserID/.ssh/uni.pub
uni.pub
100%  402     0.4KB/s   00:01
sftp> exit
```

Öffentlichen Schlüssel registrieren

Sie sind fast am Ziel. Verbinden Sie sich nun per SSH mit dem Login-Server und wechseln Sie in das versteckte Verzeichnis `/home/UserID/.ssh`. Dort müsste sich nun der Public Key `uni.pub` finden.

Prüfen Sie, ob im gleichen Verzeichnis die Datei `authorized_keys` existiert. Normalerweise sollte das nicht der Fall sein.

Wenn authorized_keys **existiert**, kopieren Sie den Public Key in die Datei hinein:

```
$ cat uni.pub >> authorized_keys # wichtig ist das ">>" statt ">"!
```

Wenn authorized_keys **nicht existiert**, benennen Sie `uni.pub` einfach in `authorized_keys` um:

```
$ mv uni.pub authorized_keys
```

> Passen Sie nun unbedingt die Rechte für `authorized_keys` an und löschen Sie die Schlüsseldatei.

```
$ chmod 660 authorized_keys
$ rm uni.pub
```

Automatisierten Login verwenden

Nun können Sie sich automatisch auf Ihrem Uni-Server einloggen. Das Kommando hierzu lautet (mit auf Ihr System angepasstem Pfad zum Schlüssel):

```
$ ssh -i /home/jannis/.ssh/uni UserID@login.uni-freiburg.de
```

Dieses können Sie nun in beliebigen Skripten nutzen um beispielsweise Backups zu erstellen oder Prozesse zu starten.

Server - Artikelübersicht, Sicherheit - Artikelübersicht, Artikel zum tag: login, Linux - Artikelübersicht, Netz - Artikelübersicht

From:
https://wiki.uni-freiburg.de/rz/ - **RZ**

Permanent link:
https://wiki.uni-freiburg.de/rz/doku.php?id=ssh_mit_public_key-authentifizierung

Last update: **2011/11/19 00:07**