SSH

Sicher vom Mac zu athanassiou.me

1 Inhaltsverzeichnis

2 SSH - Secure Shell

Secure Shell oder **SSH** ermöglicht eine sichere Verbindung zwischen zwei Rechnern über ein unsicheres Netzwerk. Dadurch dient es unter anderem als Ersatz für die Vorgänger `rlogin`, `telnet` und `rsh`. Diese übertragen jeglichen Netzverkehr, darunter auch die Passwörter, unverschlüsselt.

SSH hat eine Client–Server-Architektur, eine SSH-Client-Anwendung verbindet sich also mit einem SSH-Server. Die IANA hat dem Protokoll den TCP-Port 22, UDP-Port 22 und SCTP-Port 22 zugeordnet.

2.1 Sicherung der Transportschicht

Noch vor einer Authentifizierung des Clients wird für die Dauer der Sitzung ein geheimer Schlüssel zwischen Server und Client ausgehandelt, mit dem die gesamte nachfolgende Kommunikation verschlüsselt wird.

Dabei identifiziert sich zunächst der Server dem Client gegenüber mit einem RSA-, DSA- oder ECDSA-Zertifikat, wodurch Manipulationen im Netzwerk erkannt werden können (kein anderer kann sich als ein bekannter Server ausgeben).

2.2 Authentifizierung

Nach erfolgter Sicherung der Transportschicht kann sich der Client unter anderem per Public Key Authentifizierung mit einem privaten Schlüssel, dessen öffentlicher Schlüssel auf dem Server hinterlegt wurde, authentisieren. Die Public-Key-Authentifizierung ermöglicht, dass sich Client-Computer auch ohne Benutzerinteraktion auf SSH-Servern einloggen können, ohne dass dabei ein Passwort auf dem Client im Klartext gespeichert werden muss. Zur weiteren Absicherung können die privaten SSH-Schlüssel mit einem Passwort geschützt werden.

3 Public Key

Die Public-Key-Authentifizierung ist eine Authentifizierungsmethode, um Benutzer mit Hilfe eines Schlüsselpaars, bestehend aus privatem (private) und öffentlichem (public) Schlüssel, an einem Server anzumelden. Ein solches Schlüsselpaar ist wesentlich schwerer zu kompromittieren als ein Kennwort.

Bei der Public-Key-Authentifizierung wird der öffentliche Schlüssel auf einem Server gespeichert. Der private Schlüssel wird auf dem eigenen Rechner gespeichert, kann damit geheim gehalten und zusätzlich mit einer Kennung verschlüsselt werden. Die Kennung kann aus mehreren Wörtern bestehen (im Englischen passphrase). Die Berechnung des privaten Schlüssels aus dem öffentlichen ist je nach Wahl der Länge des Schlüssels sehr aufwendig bis praktisch unmöglich.

3.1 Generieren eines Private-Public Key Paares

Das Generieren des Private-Public Key Paares erfolgt mit dem Kommando:

```
ssh-keygen -t rsa -b 2048 -C "<ihrkommentar>"
```

Das Kommando gibt einen malerischen Output:

```
Your identification has been saved in /Users/eathanassiou/.ssh/id_rsa
Your public key has been saved in /Users/eathanassiou/.ssh/id_rsa.pub

The key fingerprint is:
SHA256:J+pvnjAAjEX9AEc+Y3DMbEjw1awX1cnL0YGjHm3pZWk <ihrkommentar>
The key's randomart image is:
+---[RSA 2048]----+
|.+*O=o ..o +..   |
| =.*B +   B .    |
|. +.=+ . + = .   |
|   o.oo o * E    |
|    .. .S+.+     |
|     . ..o.      |
|      +          |
|     . o..       |
|      .++        |
+----[SHA256]-----+
```

Die Ergebnisse werden unter `$HOME/.ssh` abgelegt.

3.1.1 Private Key

Der Pivate-Key steht in der Datei `id_rsa`, und sieht so ähnlich aus (ist natürlich nicht die echte Datei:

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEAzDnYWqCECo6m25qfY58izZ5H46TuhQhqSe1rKhZ2tk/1d+wQSPHK
4/4/BO9WlSeWibl2fTPoDtEybJMS1s8y+09u0/U1coBr7PHN7f+oFTEemFW2dqBXiEl/ve
FJLexZHVP2Hwp/dK0evJHz84+WzQd9Q+/LCeMWtsC8KFxWHtHZ8BiwiPXz1aqpnpdBx3N/
w9pYgmmO8IW2X0A2MIxozWxPIHQBc2txsJEmVu8Qfrik+0b7RwL31b4taWvA8JxSN7Fj0m
asDFgasd/89fvVHmkc/asdldnqs/j8E71aVJ5aJuXZ9M+gO0TJskxLWzzL7T27T9TVyHi
gGvs8c3t/6gVMR6YVbZ2oFeISX+94Ukt7FkdU/YfCn90rR68kfPzj5bNB31D78sJ4xa2wL
woXFYe0dnwGLCI9fPVqqmel0HHc3/D2liCaY7whbZfQDYwjGjNbE8gdAFza3GwkSZW7xB+
…
bLdpQqe/xAuCPOid4vneFgbPzrvok79YVoPvgpT2y3KaJ5fRQT8nBjEPv2Ry0JYEMen+wD
my1QDdZSR0EqwAAACBAOrhkDPFuTyg7kod+qJsZR24g29dfQUvhmWn9nT1IjLPfzYbYZgi
MdiX+Sl4b2iGRoEezwxoq7IR3QhIFGe0xA2Sd4HeRuLgdB4Ailihiu4o+SgcPmDm6RvT7a
LDNn1gM6Vw7viiD9md3QkbIKrTP73EI1Yk1d7sAeTuE36CNgfdAAAAgQDelqtEMZF+tbNk
0j0I2Jcb0qFU28V3atI6qtZNS0DohSsVBv4yu3UuSVcmNQKyGT3iVcLbpDHR5Hfsbd1kZP
WvJR0e/UjmXrYNm8/eVIofK9QNwlxMdPZsXaeHcTuGkq5X6klIfT10jWSFTzcN2QvC7eIV
qoEhlTA5k4/61qEkFwAAAA48aWhya19tbWVudGFyPgECAw==
-----END OPENSSH PRIVATE KEY-----
```

3.1.2 Public Key

Der Public-Key steht in der Datei `id_rsa.pub`:

```
ssh-rsa
AAAAB3NzaC1yc2EAABBBDAQABAAABAQDMOdhaoIQKjqbbmp9jnyLNnkfjpO6FCGpJ7WsqFna2T/
V37BBI8crj//6gVMR6YVbZ2oFeISX+94Ukt7FkdU/YfCn90rR68kfPzj5DfS234NB31D78sJ4xa
2wLwoXFYe0dnwGLCI9fPVqqmel0HHc3/D2liCaY7whbZfQDYwjGjNbEFzHIhIHIhik88e2a3Gwk
SZW7xB+uKT7 j8E71aVJ5aJuXZ9M+gO0TJskxLWzzL7T27HJJowc3t RvtHA/vfVvi1/pa8Dwn
FI3sWPSYtxcbyRSVbwxD1TJ0NWTvZiv/9fGMSaxTOVjItX97I37iRZonigw9YBtO7pXGWuf8R4R
+Zugz78sjb <ihrkommentar>
```

4 Remote login mit ssh und private key

Der öffentliche Schlüssel kann zur automatischen Anmeldung (Authentifizierung) genutzt werden. Dabei entfällt der interaktive Dialog zur Kennworteingabe. Hierzu wird auf der User-Seite mit Hilfe seines privaten Schlüssels eine Signatur erzeugt, welche daraufhin auf der Server-Seite mit dem dort hinterlegten öffentlichen Schlüssel des Users verifiziert wird.

4.1 Hinterlegen Public-Key auf Server

Bei dem Strato-Server wurde der Public Key über das Strato-Portal mit Copy-Paste hinterlegt.

4.2 Public Key auf Server übertragen

Wenn es nicht per Portal geht, gibt es dafür Linux Kommandos. Für das Übertragen des öffentlichen Schlüssels auf den Server wird im ersten Schritt noch die SSH-Verbindung mittels Passwort-Authentifizierung genutzt.

Das Kommando `ssh-copy-id` kopiert das entsprechende File auf den Server:

```
ssh-copy-id -i .ssh/key_rsa.pub root@athanassiou.me
root@athanassiou.me's password:

Now try logging into the machine, with "ssh root@athanassiou.me", and check
in:
  ~/.ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
```

Das obige Prozedere hat am **Server** in der Datei `/root/.ssh/authorized_keys` folgenden Eintrag erstellt:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDMOdhaoIQKjqbbmp9jnyLNnkfjpO6FCGpJ7Wsq
Fna2T/ +gO0TJskxLWzzL7T27T9TVygGvs8c3t/6gV…V37BBI8crj/j8E71aVJ5a
JuXZ9MAvfVvi1pa8DwnFI3sWPSYtxcbyRSVbwxD1TJ0NWTvZiv/9fGMSaxTOVjItX97I
37iRZonigw9RvYBtO7pXGWuf8R4R+Zugz78sjb <ihrkommentar>
```

4.3 Aufruf ssh vom eigenen Rechner

Jetzt muss man nur noch sich mit dem Kommando `ssh` mit dem Host verbinden.

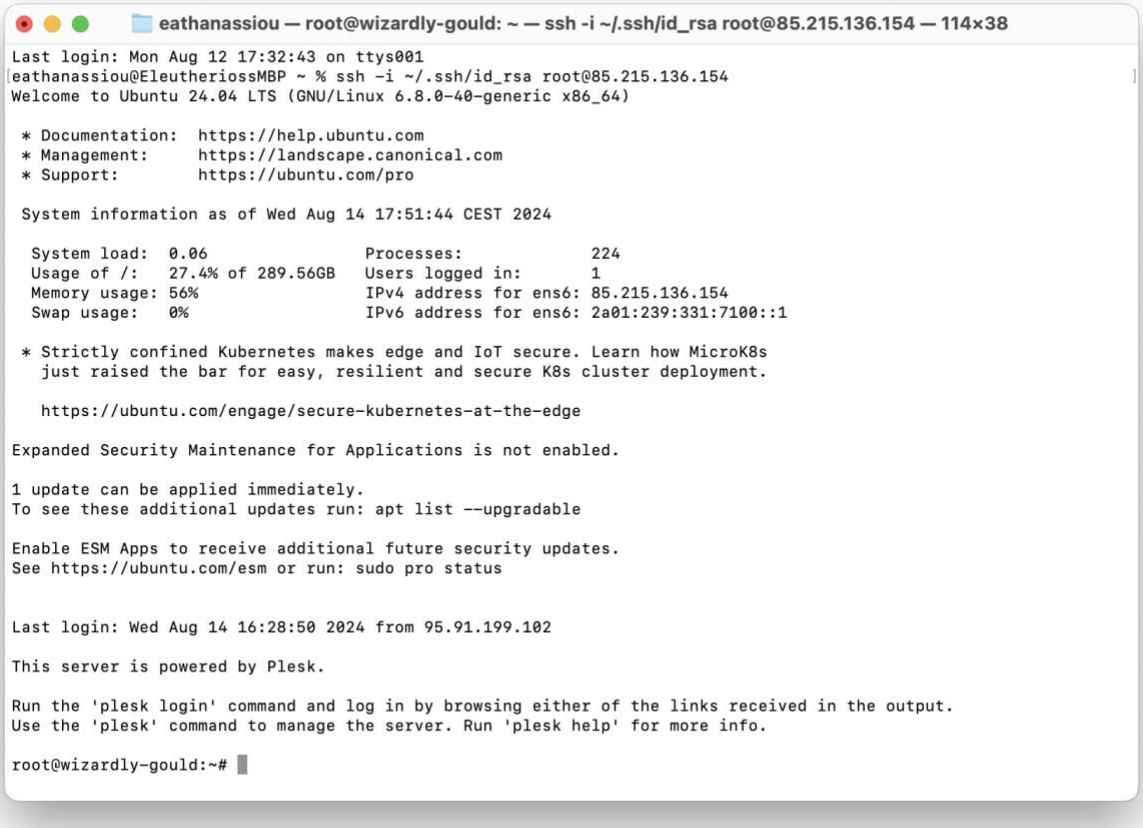
Der Aufruf:

```
ssh -i ~/.ssh/id_rsa root@85.215.136.154
```

bzw.

```
ssh -i ~/.ssh/id_rsa root@athanassiou.me
```

“does the job”:



5 Cyberduck

SSH wird aber auch benutzt um Anwendungen mit einem Server zu verbinden. Cyberduck greift so auf den STRATO-Server als sftp-Client vom Mac aus zu: