Übung #9 – Die SSH Suite

1 Ziel

In dieser Übung werden Sie den Umgang mit den verschiedenen Komponenten von *OpenSSH*, einer Impelementierung des SSH-Protokolls, kennen lernen.

Dies ist besonders erstrebenswert, da GNU/Linux Systeme oft als Server eingesetzt wird und man daher nicht davon ausgehen kann, immer physischen Zugriff auf ein System zu haben. SSH, das *S*ecure *Sh*ell Protokoll, ist die gängigste Lösung für dieses Problem.

2 Anleitung

1. Loggen Sie sich per SSH auf dem Rechner Ihres Sitznachbars ein. Benutzen Sie dazu den in einer früheren Übung angelegten Benutzeraccount "cms04".

   Vergewissern Sie sich, dass Sie tatsächlich auf dem entfernten Rechner arbeiten, indem Sie sich die Ausgabe von `hostname` ansehen.

2. Legen Sie auf dem entfernten Rechner in einem beliebigen Unterverzeichnis Ihres Heimverzeichnisses eine zumindest 10 MByte große Datei an. (Dies geht zum Beispiel durch `dd if=/dev/zero of=dateiname bs=1k count=10240`.)

   Kopieren Sie diese Datei nun mittels `scp` mitsamt dem übergeordneten Verzeichnis auf Ihren Rechner.

3. Wie können Sie in vorigem Schritt sicherstellen, dass auch die Dateizugriffsrechte mitkopiert werden? Recherchieren Sie dazu in den Manpages.

4. Führen Sie das Kommando `eject` auf dem Rechner Ihres Nachbarn direkt aus, also ohne vorher eine Shell auf dem entfernten Rechner zu öffnen (`ssh benutzer@rechner kommando`). Warnen Sie allerdings Ihren Nachbarn allerdings unbedingt im Voraus.

   Machen Sie den Effekt durch eine weitere direkte Kommandoausführung wieder rückgängig. Das zu verwendende Kommando lautet `eject -t`.

5. X Forwarding ist ein sehr nützliches Feature von SSH. Testen Sie es aus, indem Sie sich zuerst per SSH auf dem Rechner Ihres Nachbarn einloggen (vergessen Sie nicht die Option `-X` und dort ein graphisches Programm starten. Auf welchem Bildschirm erscheint die Ausgabe? (Tipp: Ein graphisches Programm, das für diesen Zweck vielleicht ganz praktisch ist, heißt `konsole`.)

6. Port Forwarding wurde bereits besprochen und ist ebenfalls ein oftmals nützliches Feature von SSH. Erstellen Sie ein Forwarding, das folgendes realisiert: Jede TCP-Verbindung, die zu Port 80 auf "localhost" (Ihrem lokalen Rechner) aufgebaut wird, soll über SSH auf den Rechner Ihres Nachbarn weitergeleitet werden, von wo aus eine Verbindung zur IP-Adresse 10.21.0.101, Port 80 aufgebaut werden soll.

Besuchen Sie nun mit einem beliebigen Browser `http://localhost/`. Welche Seite sehen Sie?

Führen Sie auf ihrem lokalen Rechner `netstat -t` aus. Welche Verbindungen sehen Sie? Wiederholen Sie das Kommando auf dem entfernten Rechner. Welche Verbindungen sehen Sie dort?

7. Wie bereits erwähnt, unterstützt SSH auch Authentisierung mit öffentlichen Schlüsseln. Dazu generieren Sie zunächst mit `ssh-keygen -t rsa` ein Schlüsselpaar. Anschließen müssen Sie die Datei .ssh/id_rsa.pub auf einen gewünschten Rechner nach .ssh/authorized_keys kopieren.

   Loggen Sie sich nun erneut auf dem entfernten Rechner ein. Was passiert? Welche Unterschiede fallen Ihnen auf?

8. Wie jedes Service, das über Netzwerk erreicht werden kann, muss auch der SSH-Server konfiguriert werden. Finden Sie die Konfigurationsdatei (sie versteckt sich irgendwo in /etc/) und ändern sie Konfiguration folgendermaßen.

   (a) Lassen Sie nur Version 2 des Protokolls zu, denn Version 1 ist nicht sicher gegen *Man in the Middle*-Angriffe.

   (b) Verhindern Sie, dass sich `root` direkt einloggen kann. Das schafft etwas mehr Sicherheit, da ein Angreifer zuerst an das Passwort eines lokalen Benutzers gelangen muss, bevor er das Rootpasswort ausprobieren kann.

   (c) Damit Sie in Zukunft gegen Angriffe Ihrer Nachbarn geschützt sind, verbieten Sie SSH-Logins für den Benutzer “cms04”.