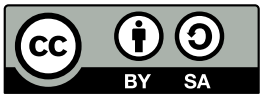
Server-Administration

Zinching Dang

20. Mai 2015



Server-Administration

Server-Administration

- Absicherung des Servers
- Einpflegen von Updates
- Instandhaltung der darauf laufenden Anwendungen
- Überwachung des Servers

Server-Administration

Server-Administration

- üblicherweise über sichere Remote-Verbindung
- viele nützliche Programme, die die Administration erleichtern
  - SSH: sichere Remote-Verbindung
  - Molly-Guard: versehentliches Herunterfahren verhindern
  - Screen: mehrere Terminal-Sessions verwalten
  - NTP: Zeit-Synchronisation
  - Syslog: Log-Files sammeln
  - SMART: Festplattenstatus überwachen
  - top: zeigt eine Systemübersicht an

Server-Administration

SSH-Server

- ermöglicht den Aufbau verschlüsselter Verbindungen über das Netzwerk
- Authentifikation entweder mit eigenem Benutzernamen und Passwort
- oder mit Benutzername und Private/Public Key
- nach Möglichkeit Passwort-basierten Login verbieten, da schwache Passwörter erraten werden können (Brute-Force)
- SSH-Root-Login mit Passwort sollte immer verboten sein

Ein SSH-Server ermöglicht es, vom Netzwerk aus eine verschlüsselte Verbindung zu dem Server, auf dem der SSH-Server läuft, herzustellen. Beim erstmaligen Verbinden zum Server wird der Benutzer aufgefordert, den Fingerprint des Servers zu überprüfen. Nach Bestätigung wird der öffentliche Schlüssel des Servers gespeichert, sodass weitere Verbindungen ohne Nachfrage aufgebaut werden können. Ändert sich der Schlüssel, erscheint beim Verbindungsaufbau eine Warnmeldung.

Das Einloggen geschieht entweder mit Benutzername und Passwort oder mit Benutzername und Private/Public Key. Letztere ist die bevorzugte Variante und für Root-Accounts empfehlenswert, da auf diese Weise keine Passwörter erraten werden können (Brute-Force).

Lab 2.1: SSH-Server

Lab: SSH-Server installieren

- Einloggen per lokaler Konsole (Virt-Manager) als Root
- SSH-Server installieren (OpenSSH-Server)
- den eigenen Public Key auf den Server für den Benutzer Root kopieren (siehe Lab 1.1)
- Verbindungsaufbau ohne Passworteingabe testen
- SSH-Root-Login mit Passwort verbieten
  - Konfigurationsdatei: `/etc/ssh/sshd_config`
  - Man-Page: `sshd_config`
  - Schlüsselwort: `PermitRootLogin`

Es soll der OpenSSH-Server installiert werden. Der Paketname muss ggf. mit `apt-cache` ermittelt werden. Nach der Installation soll nach Lab 1.1 für den Root-Account die Schlüssel-basierte Authentifizierung eingerichtet werden. Dies sollte vor dem nächsten Schritt **unbedingt** getestet werden.

Im folgenden soll der Passwort-basierte Login für den Root-Account verboten werden. Dies ist eine vorbeugende Maßnahme, um das Anmelden mit einem erratenen Root-Passwort zu verhindern. Dazu muss die Konfigurationsdatei `/etc/ssh/sshd_config` bearbeitet werden. In der Man-Page ist unter dem Stichwort `PermitRootLogin` nachzulesen, welche Einstellung dort gesetzt werden muss.

Tools: Molly-Guard

Tools

- Molly-Guard:
  - schützt vor versehentlichem Herunterfahren/Neustarten des Rechners über eine laufende Remote-Verbindung
  - Standard-Konfiguration schützt bereits sinnvoll vor diesen Unfällen

Molly-Guard schützt Rechner vor versehentlichem Herunterfahren und Neustarten über eine Remote-Verbindung. Dies ist insbesondere dann hilfreich, wenn man zu dem Rechner weder physikalischen noch Remote-Konsolen-Zugriff hat.

Stellt Molly-Guard fest, dass das Herunterfahren oder Neustarten über eine SSH-Verbindung ausgelöst wurde, wird der Benutzer aufgefordert, den Hostname des Rechners einzugeben. Dadurch wird vermieden, dass bei mehreren offenen Terminals der `shutdown`-Befehl versehentlich im falschen Terminal ausgeführt wird.

Molly-Guard ist standardmäßig bereits sinnvoll eingestellt. Um dies zu testen, kann der Befehl `shutdown -k now` verwendet werden.

Tools: Screen

Tools

- Screen:
  - verwaltet Terminal-Sessions
  - kann mehrere Sessions in einem Terminal-Fenster öffnen und parallel laufen lassen
  - Sessions können „detacht" werden, während die Programme darin weiterlaufen
  - „detached" Sessions können „(re-)attacht" werden
  - aktuell laufenden Sessions können auch von mehreren Benutzern „attacht" werden

Mit Screen können in einem Terminal-Fenster mehrere Terminal-Sessions verwaltet werden, in denen jeweils Programme parallel laufen können.

Einer der vielen Funktionen von Screen ist, dass man diese Screen-Session „detachen" und wieder „(re-)attachen" kann, während alle Programme im Hintergrund weiterlaufen.

Wird Screen gestartet, werden zuerst allgemeine Hinweise zu Screen angezeigt. Nach Bestätigung mit der Enter-Taste wird eine Shell geöffnet. In den Standard-Einstellungen kann mit der Tastenkombination `Strg-a Strg-c` eine neue Terminal-Session gestartet werden und mit der Kombination `Strg-a n` durch die Terminal-Sessions rotiert werden. `Strg-a w` listet alle offenen Sessions auf. Mit `Strg-a <n>` kann direkt in das Fenster `<n>` gewechselt werden.

Mit `Strg-a d` kann die Screen-Session „detacht" werden. Mit dem Befehl `screen -ls` lassen sich die laufenden Screen-Sessions anzeigen. „Detached" Sessions lassen sich mit `screen -r <name>` wieder „attachen". Wird kein Name angegeben, wird die erste „detached" Session „attacht".

Screen bietet noch viele weitere Funktionen, welche in der ausführlichen Man-Page beschrieben sind.

Tools: NTP

Tools

- NTP:
  - Network Time Protocol
  - synchronisiert die Uhrzeit des Rechners mit Referenz-Uhren
  - verhindert das Auseinanderdriften der Uhrzeiten auf verschiedenen Systemen
  - wichtig, da interne Uhren unterschiedliche Toleranzen aufweisen
  - erleichert die Fehlersuche in vernetzten Umgebungen

Mittels NTP kann die Uhrzeit des Rechners mit Referenz-Uhren synchronisiert werden. Dies verhindert das Auseinanderdriften der Uhrzeiten auf verschiedenen Systemen. Insbesondere bei der Fehlersuche in vernetzten Umgebungen ist dies hilfreich, da alle Rechner die selbe Uhrzeit haben und in den Log-Files Ereignisse synchron erfasst werden.

Bei der Konfiguration kann ein Time-Server angegeben werden, mit dem sich der Rechner synchronisieren soll. Dieser Time-Server sollte aus Netzwerksicht nicht zu weit entfernt sein (ein Time-Server in Australien ist wegen der hohen Latenz weniger geeignet).

Tools: SMART

Tools

- SMART:
  - Self-Monitoring, Analysis and Reporting Technology
  - kann u. a. folgende Festplatten-Eigenschaften abrufen
    - Laufzeit (in Stunden)
    - Temperatur (aktuelle, Min, Max)
    - Reallokierte Sektoren
  - kann Festplatten auf physikalische Fehler testen
  - Benachrichtigung bei Fehlern per Syslog und E-Mail

SMART ist die Abkürzung für „Self-Monitoring, Analysis and Reporting Technology" und bietet eine Überwachung der Festplatte(n). Weiterhin können Informationen wie z. B. Laufzeit, Temperatur oder Anzahl der reallokierten Sektoren angezeigt werden. Außerdem ist eine Prüfung der Festplatte auf physikalische Fehler möglich.

Der SMART-Dienst muss nach der Installation aktiviert werden, sodass auftretende Fehler automatisch festgestellt werden können. Für die Benachrichtigung per E-Mail wird ein korrekt konfigurierter Mailserver auf dem Rechner vorausgesetzt.

Tools: top

Tools

- top:
  - zeigt Systemauslastung und laufende Prozesse an
  - Informationen über Speicherbelegung, Rechenzeit, etc. der einzelnen Prozesse
  - Prozesse können nach CPU- oder RAM-Auslastung sortiert werden
  - Prozesse können auch beendet werden

top zeigt die aktuelle Systemauslastung und laufende Prozesse an. U. a. werden CPU-Auslastung, RAM-Belegung, Anzahl der eingeloggten User, die Systemlast und laufende Prozesse angezeigt. Die Anzeige wird in regelmäßigen Intervallen aktualisiert und kann u. a. nach CPU- oder RAM-Auslastung sortiert werden. Laufende Prozesse können auch beendet werden.

Lab 2.2: Tools installieren

Lab: Tools installieren

- folgende Tools installieren (ggf. in der Paket-Datenbank danach suchen):
  - Molly-Guard
  - Screen
  - NTP
  - Vim
- SMART ist bei virtuellen Festplatten nicht sinnvoll

Installiere die Programme Molly-Guard, Screen, NTP und Vim. Suche ggf. in der Paketdatenbank nach den jeweiligen Paketnamen. Das Program `top` ist bereits vorinstalliert, SMART ist bei virtuellen Festplatten nicht sinnvoll.

Vim

Nützliche Programme

- Vim:
  - effizienter und leistungsfähiger text-basierter Editor
  - Vim steht für „vi improved”, vi ist der rudimentärere Vorgänger
  - verschiedene Modi:
  - Normal mode:
    - Short-Cuts
    - Befehle
  - Insert mode
  - Visual mode
  - siehe auch `vimtutor`

Vim steht für „vi improved” und ist ein sehr leistungsfähiger und effizienter text-basierter Editor. In Vim existieren mehrere Bearbeitungs-Modi. Die wichtigsten sind: Normal mode, Insert mode und Visual mode.

Der Normal mode ist gegliedert in Short-Cuts, welche aus einzelnen oder einer Reihe von Tasten(kombinationen) bestehen, und Befehlen, welche mit einem führenden „:” eingegeben werden. Short-Cuts sind z. B. ein Zeichen löschen (`x`) oder in den Insert mode wechseln (`i`). Befehle sind z. B. Vim beenden (`:q`) oder Änderungen speichern (`:w`).

Der Insert mode verhält sich wie die meisten Editoren: es kann Text eingegeben werden. Hierbei ist zu beachten, dass die Befehle und Short-Cuts aus dem Normal mode nicht funktionieren. Mit ESC kann der Insert mode wieder verlassen werden.

Der Visual mode ermöglicht das Arbeiten über mehrere Zeilen oder Spalten. Um in den Visual mode zu wechseln, muss aus dem Normal mode der Short-Cut `v` für zeilenweise Operationen und `Strg-v` für spaltenweise Operationen verwendet werden.

Systemkonfiguration

Konfiguration

- `/etc/default/rcS`
  - Debian-spezifisch
  - Konfigurationsdatei, in der das Boot-Verhalten verändert werden kann
    - automatisches Reparieren des Dateisystems, falls Fehler gefunden werden
    - Löschverhalten des `tmp`-Verzeichnisses

In der Konfigurationsdatei `/etc/default/rcS` kann das Boot-Verhalten angepasst werden. U. a. kann das automatische Reparieren des Dateisystems, falls bei einer Dateisystemüberprüfung Fehler gefunden werden, aktiviert werden. Dies kann bei Servern nützlich sein, da der Boot-Vorgang sonst unterbrochen wird bis der Administator das Reparieren auf der lokalen Konsole bestätigt.

Ein weiteres Beispiel ist die Einstellung `TMPTIME`, mit der festgelegt werden kann, dass die Dateien in `/tmp` beim Hochfahren nur gelöscht werden, wenn sie ein bestimmtes Alter erreicht haben.

Lab 2.3: Vim und Systemkonfiguration

Lab: Vim und Konfiguration

- bearbeite die Datei `/etc/default/rcS` so, dass das Dateisystem bei Fehlern automatisch repariert werden
- siehe dazu auch in der Man-Page zu `rcS`
- konfiguriere den NTP-Client so, dass er sich mit den Time-Servern der Uni synchronisiert
- siehe dazu die Man-Page zu `ntpd` an
- mache dich mit Vim vertraut, schaue dir dazu `vimtutor` an

Bearbeite die Konfigurationsdatei `/etc/default/rcS`. Bei der Überprüfung des Dateisystems sollen Fehler automatisch repariert werden, sodass das Booten ohne Benutzerinteraktion möglich ist. Benutzte hierbei screen, um zwischen der Man-Page zu `rcS` und der Konfigurationsdatei zu wechseln.

Bearbeite die Konfigurationsdatei des NTP-Clients. Es sollen andere Time-Server eingetragen werden. Schaue dir die Man-Page zu `ntpd` an, um herauszufinden, in welcher Konfigurationsdatei die neuen Server eingetragen werden müssen.

Arbeite dich ein wenig in Vim ein. Starte dazu `vimtutor`.