Hauke Goos-Habermann

Stand 2014/08

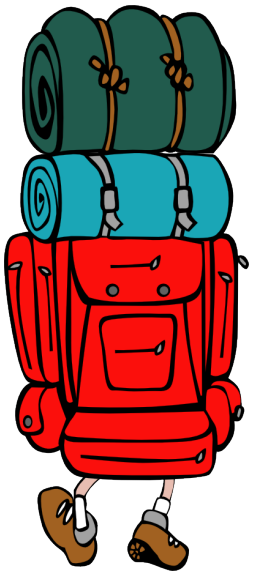
Inhaltsverzeichnis

Kapitel 1

Workshopvorbereitungen

1.1 Was muß ich mitbringen?

Zum Workshop mitzubringen sind: Laptop mit installiertem VirtualBox und darin importierter Debian-Admin-Workshop-Appliance, sowie funktionierendes W-Lan bzw. Netzwerkkarte (+ Kabel) und dieses Workshop-PDF. Sowie eine aktuelle ISO-Datei der Live-Linux-Distribution Knoppix [1] (ca. 700 MB).



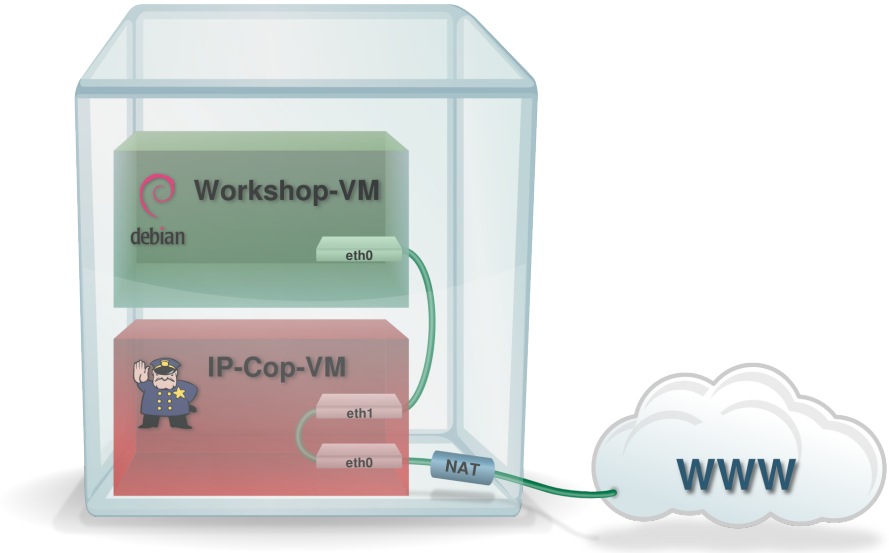
1.2 Debian-Admin-Workshop-Appliance

Für den Workshop wurden extra zwei virtuelle Maschinen (VMs) erstellt und in ein Installationspaket verpackt. Dieses Installationspaket – auch Appliance genannt – kann nun auf anderen Rechnern importiert werden, um die beiden VMs zu erhalten. Das Installationspaket kann von der Projektseite [2] (oder als Kurzurl: `http://is.gd/kljrV7`) heruntergeladen werden.

1.3 Die beiden VMs

Bei den beiden VMs handelt es sich um ein mit m23 [3] aufgesetztes Debian [4] Wheezy (erweitert um systemd [5] aus testing), auf dem die Aufgaben des Workshops gelöst werden sollen und eine VM mit der Firewalldistribution IPCop [6], die als virtueller Router genutzt wird. Die letztere VM muß während des Workshops zwecks Internetverbindung angeschaltet sein, braucht jedoch nicht verändert zu werden.

[1] `http://www.knopper.net/knoppix-mirrors/`
[2] `http://sf.net/projects/dodger-tools/files/vms/Debian-WS2a.ova/download`
[3] `http://m23.sf.net`
[4] `http://debian.org`
[5] `http://de.wikipedia.org/wiki/Systemd`
[6] `http://www.ipcop.org`



1.4 Import

Das Installationspaket läßt sich unter VirtualBox [7] importieren. Hierzu lädt man VirtualBox für das eigene Betriebssystem herunter und installiert es, wie auf der VirtualBox-Seite angegeben. Danach startet man VirtualBox und importiert die Appliance wie folgt:

- Datei > Appliance importieren ...
- Appliance öffnen ...
- Dann die heruntergeladene Datei `Debian-WS.ova` auswählen.
- Weiter >
- Wichtig: "*Zuweisen neuer MAC-Adressen für alle Netzwerkkarten*" bleibt **DEAKTIVIERT**.
- Importieren
- Etwas warten ... (die "Debian-Admin-Workshop"-VM kann absichtlich keine Netzwerkverbindung aufnehmen, daher dauert hier das Starten etwas länger)
- Fertig

1.5 Benutzer und Paßworte

Nutzername / Paßwort für die "Debian-Admin-Workshop"-VM: test / test und root / test. Für die "IPCop"-VM: root / testtest (nur der Vollständigkeit halber erwähnt, wird nicht benötigt). Die Weboberfläche von IPCop ist zudem über HTTPS auf dem Port 8443 mit admin/testtest zu erreichen (https://192.168.1.4:8443) – dies allerdings nur innerhalb des abgetrennten vituellen Subnetzwerkes und nicht direkt vom Gastgeber-Rechner.

[7] `http://www.virtualbox.org`

Kapitel 2

Grundlagen

2.1 "root" werden

"Ich bin root, ich darf das!". Aber wie wird man root? Unter Debian hilft das Kommando `su ↵` , unter Ubuntu und anderen `sudo bash ↵` . Für alles, was nun folgt, werden entweder root-Rechte benötigt oder es stellt keinen Nachteil dar, diese Schritte als root durchzuführen.

Aber immer bedenken: "Mit großer Macht kommt auch große Verantwortung".

Im "wahren Leben" sollte man sich vor jeder Eingabe auf der Kommandozeile gut überlegen, ob Rootrechte notwendig sind und diese so selten wie möglich einsetzen. Wenn man nicht weiß, ob man root sein muß, um eine bestimmte Aktion durchführen zu können, sollte man es zunächst ohne Rootrechte versuchen – normalerweise weist einen das System darauf hin, wenn dies nicht erfolgreich war.

Es ist auch möglich, jedem Root-Befehl ein `sudo ↵` voranzustellen, ohne sich zuvor als root eingeloggt zu haben. Dies erfordert jedoch, daß das Programm "sudo" auf dem Rechner auch installiert ist - und der eigene Benutzer auf Ubuntu und -Derivaten Mitglied der Gruppe "sudo" ist.

2.2 Welche Distribution und Architektur?

Wo bin ich denn hier?

Grundlegende Informationen zum Rechner (Architektur, etc.) bietet die Ausgabe von `uname -a ↵` . Die Bezeichnung der Distribution läßt sich durch ein einfaches `cat /etc/issue ↵` oder `lsb_release -a ↵` ermitteln.

`uname -a ↵` gibt Informationen über den Namen, die Architektur, den Kernel und den Prozessor des Systems aus. Der Parameter -a oder $--$all sagt uname, daß es alle Informationen ausgeben soll.

`cat ↵` ist ein GNU-Werkzeug, das Inhalte von Dateien einliest und auf die Standardausgabe (in diesem Fall das Terminal) ausgibt. In der Datei `/etc/issue` ist der Name der Distribution angegeben.

Über `lsb_release -a ↵` können die Distributions-Informationen über die LSB-Schnittstelle (Linux Standard Base [1]) ausgelesen werden.

2.3 Pfadangaben

Linux kennt zwei verschiedene Arten von Pfadangaben, die zu einer Datei oder einem Verzeichnis führen:

- Die absolute Pfadangabe: `/etc/apt/sources.list`

[1] `http://de.wikipedia.org/wiki/Linux_Standard_Base`

- Die relative Pfadangabe. Hier: Die Datei eine Ebene über dem aktuellen Verzeichnis: `../datei.txt`
- Die relative Pfadangabe. Hier: Die Datei eine Ebene unter dem aktuellen Verzeichnis: `unterverzeichnis/datei.txt`
- Ausführen einer Datei im aktuellen Verzeichnis: `./mein-Shell-Skript.sh` ↵
- Verwenden einer Datei im aktuellen Verzeichnis (mit einem Programm): `nano datei.txt` ↵
- Datei im Heimatverzeichnis (`~`) des aktuellen Benutzers: `~/datei.txt`

2.4 Die Arbeit beschleunigen

Auch wenn es in Filmen "cool" aussieht, wenn "Hacker" wüst auf ihre Tastatur einhacken, sollte man unnötige Schreibarbeit vermeiden, da Tippen fehleranfällig ist und die BASH [2] einem viel Arbeit abnehmen kann.
Hierzu ein paar "Tipps" zum Vermeiden von "Tippen":

- Mit den Tasten ↑ und ↓ kann man durch die bereits eingegebenen Befehlszeilen blättern.
- Die Tasten ← und → bewegen den Cursor in der aktuellen Zeile.
- Strg + r startet die Suche in bereits eingegebenen Kommandozeilen. Danach gibt man einen Suchbegriff ein, worauf die erste passende Zeile als Vorschlag angezeigt wird. Gibt es mehrere auf den Suchbegriff passende Zeilen, so zaubert Strg + r die jeweils nächste Zeile auf den Bildschirm. Möchte man den Vorschlag annehmen, so drückt man abschließend ↵.
- Wenn es um Befehle (und ggf. deren Parameter), Dateien und Verzeichnisse geht, reicht es, die (eindeutigen) Anfangsbuchstaben einzugeben und anschließend ⇥ (Tabulator-Taste) zu drücken. Gibt es nur einen Befehl (, Datei, ...), der mit den Buchstaben beginnt, so erscheint dieser auf der Kommandozeile. Sollte das nicht der Fall sein, so hilft ein weiterer Druck auf ⇥, um alle passenden Befehle (...) aufzulisten. Anschließend hilft die Erweiterung der bereits eingegebenen Buchstaben um weitere, damit der Befehl (...) eindeutig gefunden werden kann. Sind genug eindeutige Buchstaben eingegeben, so vervollständigt ⇥ abschließend.
- Das Paket "*gpm*" erweitert die tastaturbasierte Kommandozeile um einen Mauszeiger. Durch Klicken und Ziehen des Zeigers können Textbereiche markiert und durch Klicken der mitteleren Maustaste (bzw. durch gleichzeitiges Drücken von linker und rechter Maustaste) der Text an der Position des Textcursors wieder eingefügt werden.

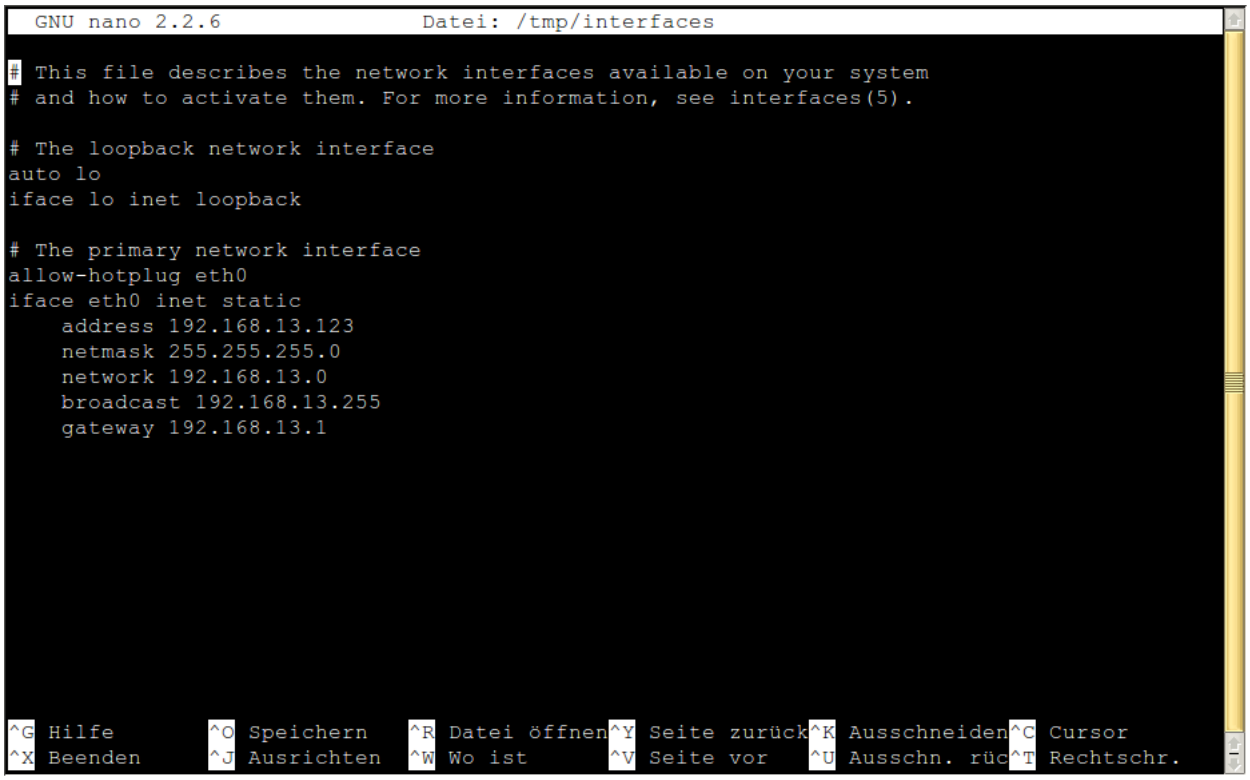
Hinweis

Eine Installation von Software auf der Workshop-VM ist allerdings erst möglich, wenn das Netzwerk repariert wurde! (siehe Seite 17)

2.5 Praktische Helfer

Der Zweispaltendateimanager "Midnight Commander" [3] (Paket "*mc*") ist ein praktisches Werkzeug für viele Dateioperationen (Aufruf mit `mc` ↵).
Wer vi, vim und Konsorten mag und damit umzugehen weiß, wird mit dem wenig kryptischen Texteditor "Nano" wenig anfangen können. Wer es aber lieber einfach mag, findet in diesem einen gut zu bedienenden Editor für alle Textdateien (Aufruf mit `nano <Name der existierenden oder neuen Datei>` ↵). Nano kann sowohl eine vorhandene Datei öffnen, als auch neue anlegen.

[2] `http://de.wikipedia.org/wiki/Bourne-again_shell`
[3] `http://de.wikipedia.org/wiki/Midnight_Commander`



Mit der Tastenkonbination `Strg + x` beendet man Nano wieder. Sollte man Änderungen an der Datei vorgenommen haben, so fragt Nano aber vorher, ob diese gespeichert werden sollen. Die Suchfunktion startet man mit `Strg + w` . Weitere Funktionen stehen unteren Rand im Editor, wobei "^" für die Taste `Strg` .

2.6 Kopieren, verschieben, löschen

Wem "mc" zu komfortabel sein sollte oder wenn "mc" nicht zur Verfügung steht, gibt es natürlich Kommandozeilenwerkzeuge für alle Dateioperationen:

- Verzeichnis(se) anlegen: `mkdir <Verzeichnisname1> <Verzeichnisname2>` ↵
- Datei(en) kopieren: `cp <Datei1> <Datei2> <Zielverzeichnis>` ↵
- Datei kopieren und umbenennen: `cp <Datei> <Datei neuer Name>` ↵
- Datei in anderes Verzeichnis kopieren und umbenennen: `cp <Datei> <Pfad/Datei neuer Name>` ↵
- Verzeichnis(se) kopieren: `cp -r <Verzeichnis1> <Verzeichnis2> <Zielverzeichnis>` ↵
- Datei(en) verschieben: `mv <Datei1> <Datei2> <Zielverzeichnis>` ↵
- Datei umbenennen: `mv <Datei> <Datei neuer Name>` ↵
- Datei(en) löschen: `rm <Datei1> <Datei2>` ↵
- Leere(s) Verzeichis(se) löschen: `rmdir <Verzeichnis1> <Verzeichnis2>` ↵
- Verzeichis(se) mit Unterverzeichnissen und Dateien löschen: `rm -r <Verzeichnis1> <Verzeichnis2>` ↵

Sollen beim Kopieren die Zugriffsrechte mitkopiert werden, so bewirkt dies der zusätzliche Parameter `-a` .

2.7 Zugriffsrechte

Unter Linux besitzen jede Datei und jedes Verzeichnis Zugriffsrechte, die regeln, wer wie und ob zugreifen darf. Mit `ls -l ↵` kann man sich die ausführliche Liste der Dateien und Verzeichnisse (inklusive Zugriffsrechten) ausgeben lassen.

```
1 lrwxrwxrwx   1 peter  gruppea             13  4. Apr 2012  todo.txt -> /backup/todo.txt
2 -rw-r--r--   1 paul   gruppeb          65536  1. Jan 2012  Paulschreibt-andere-lesen.txt
3 -rw-rw-r--   1 peter  gemeinsamegruppe   128 18. Aug 2013  Gruppentext.txt
4 -rw-------   1 paul   paul              7889 21. Jul 23:26 Paulsgeheime_Gedanken.txt
5 drwxr-xr-x   2 paul   paul              4096 23. Jul 2012  Paulschreibt-andere-lesen
```

Das Schema ist folgendes:

<Typ><BBB><GGG><AAA> <H> <Benutzer> <Gruppe> <Größe> <Änderungszeit> <Datei-/Verzeichnisname>

Der Typbezeichnung, bei der "-" für eine Datei, "d" für ein Verzeichnis und "l" für einen symbolischen Link steht, folgt ein Block von $3 * 3$ Zeichen. Dabei steht der erste Block für die Rechte des Besitzers, der zweite für die Rechte der Gruppe und der letzte für alle anderen. "r" steht für das Leserecht, "w" für das Schreib-/Änderungsrecht und "x" für das Ausführungsrecht. Nach den Rechten folgt eine Angabe der Anzahl der Hardlinks [4] auf die Datei (mind. 1) oder das Verzeichnis (mind. 2). Danach kommen der Name des Besitzers, dann der der Gruppe, die Dateigröße, dann die Änderungszeit und abschließend der Name der Datei bzw. des Verzeichnisses oder Informationen über den symbolischen Link.

`ls` steht dabei für "list" und ist damit wohl einer der am häufigsten genutzten Shell-Befehle. Der Parameter `-l` sorgt dafür, daß soviele Informationen wie möglich über die aufgelisteten Dateien und Verzeichnisse ausgegeben werden. Ein zusätzliches `-a` zeigt alle, also auch versteckte, Dateien an.

2.7.1 Besitzer und Gruppe ändern

- Besitzer ändern: `chown <Neuer Besitzer> <Datei-/Verzeichnisname> ↵`
- Gruppe ändern: `chgrp <Neue Gruppe> <Datei-/Verzeichnisname> ↵`
- Besitzer und Gruppe ändern: `chown <Neuer Besitzer>:<Neue Gruppe> <Datei-/Verzeichnisname> ↵`

Sollen bei den Verzeichnissen die Unterverzeichnisse und enthaltene Dateien mit eingeschlossen werden, so kommt noch der Parameter `-R ↵` hinzu.

2.7.2 Zugriffsrechte ändern

Die Zugriffsrechte für Besitzer, Gruppe und Andere können jeweils als Zahl angegeben werden. Das Leserecht hat den Wert 4, das Schreib-/Änderungsrecht "2" und das Ausführenrecht "1". Sollen mehrere Rechte vergeben werden, so werden die Werte einfach addiert (z.B. Lesen (4) + Schreiben (2) = 6). Das Ändern der Rechte geschieht mit `chmod <BGA> <Datei-/Verzeichnisname> ↵` .

Beispiel: `chmod 754 info.txt ↵` erlaubt dem Besitzer das Lesen, Schreiben und Ausführen, der Gruppe das Lesen und Ausführen und anderen das Lesen der Datei `info.txt`.

Sollen hierbei Verzeichnisse, Unterverzeichnisse und enthaltene Dateien mit eingeschlossen werden, so kommt ebenfalls der Parameter `-R` hinzu.

2.8 Suchen und finden

Häufig ist es nötig, Dateien zu finden. Hierbei helfen die Programme `grep` und `find` . Weiterhin gibt es noch `locate` , was sehr schnell, jedoch auf einen regelmäßig aktualisierten Suchindex (`updatedb`) angewiesen ist, sowie `whereis` , das in Standardpfaden nach Dateien sucht, die zu einem angegebenen Programm gehören.

[4] http://de.wikipedia.org/wiki/Hardlink

- Dateien im aktuellen Verzeichnis und Unterverzeichnissen (`-r`) suchen, deren Inhalt die Zeichenkette "test" enthält: `grep test -r . ↵`
- Groß-/Kleinschreibung von "test" ignorieren (`-i`): `grep test -r -i . ↵`
- Nur Dateinamen (`-l`) mit Treffern ausgeben: `grep test -r -i -l . ↵`
- Datei-/Verzeichnisnamen, die "test" enthalten, im aktuellen Verzeichnis und Unterverzeichnissen suchen: `find | grep test`

Hierbei gibt `find ↵` (ohne Parameter) einfach alle Datei- und Verzeichnisnamen aller Unterverzeichnisse aus und "grep" sucht in dieser Ausgabe nach dem Suchbegriff. Der senkrechte Strich dazwischen (das Pipe-Symbol) sorgt dafür, daß die Ausgabe des ersten Programmes an das nachfolgend angegebene Programm übergeben wird.

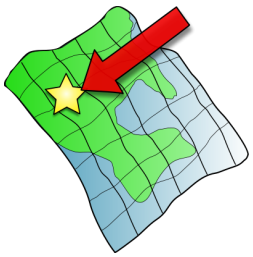
2.9 man und −−help

Wenn mal kein Netz zur Verfügung stehen sollte oder weil es bei einigen Dingen einfach schneller ist, Hilfetexte direkt aus dem System zu bekommen, soll das Kommando `man <Befehl> ↵` nicht unerwähnt bleiben. Neben "man" bieten viele Programme auch eine eingebaute (meist kürzere) Hilfe an, die mittels `<Programm> −−help ↵` aufgerufen wird. Wenn `−−help` nicht zum Erfolg führt, kann der Parameter auch (nach absteigender gefühlter Vorkommenshäufigkeit sortiert) `-help` , `-h` oder `-?` heißen. Alternativ kann das Programm den Hilfetext auch ausgeben, wenn kein Parameter oder ein ungültiger Parameter angegeben wird. Einfach mal ausprobieren...

Kapitel 3

Was ist wo?

Ein kleiner Wegweiser durch den Linux-Verzeichnisbaum.



- Ausführbare Dateien für alle Benutzer: `/bin` und `/usr/bin`
- Ausführbare Dateien für root: `/sbin` und `/usr/sbin`
- Gerätedateien: `/dev`
- Kernel und Initrd: `/boot`
- grub: `/boot/grub`
- Globale Einstellungen: `/etc`
- Globale Programmeinstellungen: `/etc/Programmname` oder `/etc/default/Programmname`
- Benutzerverzeichnisse: `/home/Benutzername`
- Start-/Stop-Skripte: `/etc/init.d`
- Protokolldateien: `/var/log`

Siehe auch Filesystem Hierarchy Standard (FHS) [1]

[1] `http://de.wikipedia.org/wiki/Filesystem_Hierarchy_Standard`

Kapitel 4

Dienste und Dämonen

Die Dämonen sind die dienstbaren Geister des Systems oder einfach ausgedrückt, Programme, die im Hintergrund unauffällig ihren Dienst versehen (bzw. sollten). Der Apache-Webserver z.B. läuft als Dämon, um Seitenanfragen entgegenzunehmen und zu beantworten. Doch auch auf reinen Desktopsystemen laufen eine Reihe von Dämonen wie z.B. ein Dämon zum automatischen Apassen der CPU-Taktfrequenz. Viele Dämonen werden direkt beim Systemstart mitgestartet.

4.1 SysVinit

Das SysVinit [1] ist zwar schon etwas in die Jahre gekommen, aber in der einen oder anderen Form noch auf vielen Systemen vorhanden. Es ist der erste Prozeß, der beim Hochfahren vom Kernel gestartet wird und sorgt dann dafür, daß die anderen Dienste in einer festgelegten Reihenfolge gestartet werden. Daher kann es nicht schaden, sich damit auseinanderzusetzen:

- Liste aller Dämonen ausgeben: `service -status-all` ↵
- (Gestoppten) Dämon starten: `/etc/init.d/<Dämon> start` ↵ oder `service <Dämon> start` ↵
- (Gestarteten) Dämon stoppen: `/etc/init.d/<Dämon> stop` ↵ oder `service <Dämon> stop` ↵
- (Laufenden) Dämon beenden und neu starten: `/etc/init.d/<Dämon> restart` ↵ oder `service <Dämon> restart` ↵

Neben den Kommandos "start", "stop" und "restart" können je nach Dämon weitere hinzukommen. Normalerweise verrät `/etc/init.d/<Dämon>` ↵ die möglichen Kommandos.

4.2 Upstart

Upstart [2] ist eine Alternative zum klassischen SysVinit und wird unter Anderem bei Ubuntu verwendet. Teilweise sind auf ein und demselben System auch beide – SysVinit und upstart – anzutreffen, was auch gelegentlich zu Problemen durch eine von der Planung abweichende Startreihenfolge führen kann.

- Liste aller Dämonen ausgeben: `initctl list` ↵

[1] http://de.wikipedia.org/wiki/SysVinit
[2] http://de.wikipedia.org/wiki/Upstart

- (Gestoppten) Dämon starten: `initctl start <Dämon>` ↵
- (Gestarteten) Dämon stoppen: `initctl stop <Dämon>` ↵

4.3 systemd

systemd [3] ist ein weiteres init-System, das ab Debian 8 Jessie und in den kommenden Ubuntu-Veröffentlichungen zum Standard wird. Angeblich soll es sogar den BAfH-Ausredekalender und einen Quake-Server enthalten...

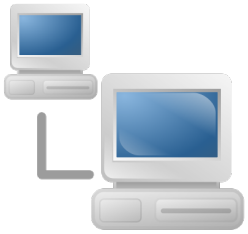
- Liste aller Dämonen ausgeben: `systemctl -all` ↵
- (Gestoppten) Dämon starten: `systemctl start <Dämon>` ↵
- (Gestarteten) Dämon stoppen: `systemctl stop <Dämon>` ↵
- Status- und Loginformationen über Dämon abfragen: `systemctl status <Dämon>` ↵
- Loginformationen über Dämon abfragen: `journalctl -u cron` ↵

[3] http://de.wikipedia.org/wiki/Systemd

Kapitel 5

Netzwerk

Was wäre das Leben ohne Netzwerk? Ziemlich öde oder vielleicht auch deutlich produktiver (ohne die ganzen Ablenkungen)... Jedenfalls kann es nützlich sein, wenn man weiß, wie man das Netzwerk selbst einrichtet. Bei der Workshop-VM ist es zumindest zunächst kaputt...



5.1 Aktuelle Netzwerkeinstellungen

Das Kommando `ifconfig ↵` gibt einen einfachen Überblick über die aktuellen Einstellungen aller aktiven Netzwerkkarten:

```
1  eth0      Link encap:Ethernet  Hardware Adresse 00:f3:ab:77:3d:28
2            inet Adresse:192.168.1.123  Bcast:192.168.1.255  Maske:255.255.255.0
3            inet6-Adresse: fe80::200:f0fe:fe83:83f6/64 Gueltigkeitsbereich:Verbindung
4            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
5            RX packets:850053 errors:0 dropped:0 overruns:0 frame:0
6            TX packets:533491 errors:0 dropped:0 overruns:0 carrier:0
7            Kollisionen:0 Sendewarteschlangenlaenge:1000
8            RX bytes:1021113415 (973.8 MiB)  TX bytes:63432614 (60.4 MiB)
9            Interrupt:19
10
11 lo        Link encap:Lokale Schleife
12           inet Adresse:127.0.0.1  Maske:255.0.0.0
13           inet6-Adresse: ::1/128 Gueltigkeitsbereich:Maschine
14           UP LOOPBACK RUNNING  MTU:16436  Metrik:1
15           RX packets:917067 errors:0 dropped:0 overruns:0 frame:0
16           TX packets:917067 errors:0 dropped:0 overruns:0 carrier:0
17           Kollisionen:0 Sendewarteschlangenlaenge:0
18           RX bytes:725063143 (691.4 MiB)  TX bytes:725063143 (691.4 MiB)
19
20 wlan0     Link encap:Ethernet  Hardware Adresse 00:2e:2d:56:34:cc
21           UP BROADCAST MULTICAST  MTU:1500  Metrik:1
22           RX packets:0 errors:0 dropped:0 overruns:0 frame:0
23           TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
24           Kollisionen:0 Sendewarteschlangenlaenge:1000
25           RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

In diesem Beispiel kennt das System drei Netzwerkanschlüsse:

- Die "echte" Netzwerkkarte für kabelgebundenes Netzwerk (ab Zeile 1)
- Die "lokale Schleife" oder das "Loopback" (ab Zeile 11), eine "Art Netzwerk" (tm), das die Kommunikation von lokal installierten Servern und Clients untereinander erlaubt (sodaß man z.B. auf die nur lokal vorhandenen Webseiten, die von einem lokal installierten Server generiert werden, auch lokal zugreifen kann)

- Die WLAN-Karte (ab Zeile 20). Diese hat aktuell keine "inet Adresse" bzw. "inet6-adresse", also keine IP - sie ist damit nicht aktiv.

5.2 Netzwerkfehler identifizieren

Wenn das Netzwerk streikt, kann folgende Testreihe zum Erkennen des Fehlers beitragen:

- Hat die Netzwerkkarte(n) eine "richtige" IP?: `ifconfig ↵`
- Welche Router/Gateway(s) sind gesetzt?: `route -n ↵`
- Kann der Router/Gateway "gepingt" werden?: `ping <Router-IP> ↵`
- Kann ein Server im Internet erreicht werden?: `ping 8.8.8.8 ↵`
- Funktioniert die Namensauflösung?: `ping heise.de ↵`

Sollten alle Tests bestanden sein, so liegt anscheinend kein Fehler vor. Ansonsten kann auch schon einmal das Netzwerkkabel herausgerutscht sein (abgebrochene Plastiknasen bei den Steckern), der Router ausgefallen oder falls `ifconfig ↵` die Netzwerkkarte gar nicht auflistet, der Netzwerkkartentreiber fehlen.

Hat die Netzwerkkarte die falsche Bezeichnung (z.B. eth1 statt eth0), so kann dies daran liegen, daß durch udev [1] ein neuer Name vergeben wurde. In diesem Fall sollte man:

- Die udev-Regeldatei für Netzwerkkarten als Backup in ein anderes Verzeichnis verschieben. Die Datei ist `/etc/udev/rules.d/` wobei die Shell das Sternchen automatisch durch die Zahl ersetzt, die die Auswertungsreihenfolge der Regeln im Verzeichnis beschreibt und auf verschiedenen Systemen unterschiedlich sein kann. Das Verschieben geschieht schließlich mit `mkdir /etc/udevbackup ↵` und

  `mv /etc/udev/rules.d/*-persistent-net.rules /etc/udevbackup/*-persistent-net.rules ↵`

- Danach den Rechner neu starten: `reboot ↵`

5.3 Netzwerkeinstellungen per Hand

Wenn gar nichts mehr geht, kann man einen Rechner relativ einfach wieder ins Netz bringen. Dieses sollte man auf der Workshop-VM unbedingt tun, damit weitere Programme (mc, gpm, ...) installiert und auch das Debian-System aktualisiert werden können.

5.3.1 Statische IP

Mit drei Zeilen werden alle nötigen Einstellungen vorgenommen. Im Beispiel soll die erste Netzwerkkarte (eth0) des eigenen Rechners die IP "192.168.1.123" zugewiesen bekommen. Die IP des Routers/Gateways ist "192.168.1.4" und der öffentliche nicht zensierende DNS-Server besitzt die IP "85.88.19.10". Diese Einstellungen sind jedoch temporär und gehen nach dem nächsten Booten wieder verloren.

- IP setzen: `ifconfig eth0 192.168.1.123 ↵`
  Hiermit wird für die Netzwerkkarte (eth0) die IP auf 192.168.1.123 gesetzt.
- Gateway: `route add -net default gw 192.168.1.4 ↵`
  Jetzt teilt man dem System noch mit, über welchen Computer (Router) es das Internet erreichen kann. In diesem Fall ist das die IPCop-VM.

[1] `http://de.wikipedia.org/wiki/Udev`

- DNS: `echo 'nameserver 85.88.19.10' > /etc/resolv.conf` ↵
  Nun benötigt das System noch eine Angabe dazu, welcher DNS-Server zur Namensauflösung genutzt werden soll. Dieser wird in die Datei /etc/resolv.conf eingetragen.

Hinweis

Eine falsche Route kann mit `route del -net default gw <IP des falschen Routers>` ↵ gelöscht werden.

5.3.2 Dynamische IP

Werden Netzwerkeinstellungen dynamisch durch einen DHCP-Server vergeben, so holt `dhclient` ↵ die Informationen direkt vom DHCP-Server. Sollte das nicht funktionieren, so hilft ggf. `dhclient -r eth0` ↵ , wodurch die Bindung der 1. Netzwerkkarte an den DHCP-Server gelöst wird und dann `dhclient eth0` ↵ zum erneuten Konfigurieren der Netzwerkkarte mit den Angaben des DHCP-Servers.

Hinweis

Bei der Workshop-VM funktionieren beide Methoden, da die IPCop-VM auch einen DHCP-Server bereitstellt.

Hinweis

Wenn einmal nicht klar ist, welche Maschine eigentlich als DHCP-Server fungiert, kann man dies in der Datei /var/lib/dhcp/dhclient.leases nachschauen. Die IP des DHCP-Servers findet sich in der Zeile, die mit "option dhcp-server-identifier" beginnt.

5.4 Netzwerkeinstellungen über Dateien

Damit die Einstellungen nicht bei jedem Neustart verlorengehen, werden diese in der Datei `/etc/network/interfaces` definiert. Als Orientierung können diese beiden Beispiele (statische und dynamische Adreßzuweisung) für /etc/network/interfaces dienen:

```
1  #Statische IP
2  allow-hotplug eth0
3  iface eth0 inet static
4          address 192.168.1.123
5          netmask 255.255.255.0
6          network 192.168.1.0
7          broadcast 192.168.1.255
8          gateway 192.168.1.100
9
10 #Dynamische IP (DHCP):
11 allow-hotplug eth0
12 iface eth0 inet dhcp
```

Die IP des DNS-Servers landet in der Datei `/etc/resolv.conf` mit folgender Notation: `nameserver 85.88.19.10`
Diese schreibt man z.B. mit `nano /etc/resolv.conf` ↵ oder
`echo 'nameserver 85.88.19.10' > /etc/resolv.conf` ↵ .

Daß die Änderungen wirksam werden, bewirkt `/etc/init.d/networking restart` ↵ . Sollte das nicht funktionieren, hilft ein Neustart mit `reboot` ↵ .

5.5 Fernzugriff

Auf die Shell von Linux-Systemen kann (bei entsprechender Einstellung) auch über das Netzwerk zugegriffen werden. Dieser Zugriff erfolgt in der Regel mittels Secure Shell (SSH [2]). Das Installationspaket, das hierfür benötigt wird heißt "openssh-server" (auf dem Zielrechner zu installieren) bzw. "openssh-client" (auf dem Rechner, an dem man arbeitet zu installieren).

Hinweis

Da die Workshop-VM sich in einem eigenen, vom Netzwerk des Gastgeber-Rechners getrennten, virtuellen Netzwerk befindet, ist eine direkte SSH-Verbindung vom Host-Rechner, auf dem die Virtualisierungssoftware läuft, zur Workshop-VM ohne weitere Einstellungen an der IPCop-Maschine / Änderungen an der Netzwerkkonfiguration der VM nicht möglich. Zum Ausprobieren empfiehlt sich daher hier das Herstellen einer Verbindung von der Workshop- zur IPCop-VM, auf der ein SSH-Server (auf Port 8022) läuft.

Ein "Anruf" mit SSH sieht dann so aus:

```
ssh -p <Port> <Benutzer>@<Rechner> ↵
```

Die Angabe `-p <Port>` wird nur dann benötigt, wenn der SSH-Server auf einem anderen als dem Standard-Port 22 lauscht. Mit "<Benutzer>" ist der Benutzer auf dem entfernten Rechner gemeint. Sind der Benutzername auf lokalem und entfernten System identisch, kann die Angabe von "<Benutzer>@" entfallen. "<Rechner>" steht für die IP des Rechners oder dessen auflösbaren Namen. Wird als zusätzlicher Parameter `-X` angegeben, so können sogar grafische Programme auf dem entfernten Rechner ausgeführt und die Ausgabe auf dem lokalen System benutzt werden, falls auf diesem ein X-Server installiert ist.

Zum Einloggen auf der IPCop-VM führt man folgende Zeile aus:

```
ssh -p 8022 root@192.168.1.4 ↵
```

Die dabei erscheinende Frage, ob man mit dem Verbindungsaufbau fortfahren möchte, ist mit "yes" zu Bejahen. Das Paßwort für den root-Benutzer ist "testtest". "root@ipcop" am Anfang der Eingabezeile zeigt da, daß man sich als Benutzer "root" auf dem Rechner "ipcop" befindet.

5.6 Kopieren über das Netzwerk

Für das Kopieren von Dateien und Verzeichnissen zwischen entferntem und lokalem System gibt es die Programme SCP [3] und rsync [4], wobei rsync mächtiger (Fortsetzen von abgebrochenen Kopieraktionen, "intelligentes" Kopieren, ...), aber auch nicht auf jedem System installiert, ist.

Beispiele:

- Datei /etc/issue von der IPCop-VM in das Verzeichnis /tmp auf die Debian-VM kopieren (Achtung: Großes "P" bei der Portangabe): `scp -P 8022 root@192.168.1.4:/etc/issue /tmp ↵`
- Lokales Verzeichnis "Adminzeug" mit Unterverzeichnissen `-r` in das Heimatverzeichnis von "admin" kopieren: `scp -r Adminzeug admin@entfernteKiste:~ ↵`
- Dasselbe mit rsync: `rsync -Pazy Adminzeug admin@entfernteKiste:~ ↵`

Die Parameterkombination `-Pazy` steht dabei für Fortschrittsanzeige `-P` , Archivmodus `-a` (mit Unterverzeichnissen, symbolische Links als symbolische Links, Berechtigungen erhalten, Änderungszeiten erhalten, Gruppenzugehörigkeit erhalten, Besitzer erhalten, Gerätedateien und Spezialdateien erhalten), Dateien komprimiert übertragen `-z` , Optimierung der übertragenen Datenmenge `-y` .

[2] http://de.wikipedia.org/wiki/Ssh
[3] http://de.wikipedia.org/wiki/Secure_Copy
[4] http://de.wikipedia.org/wiki/Rsync

Kapitel 6

Paketverwaltung

(Fast) Jede Linux-Distribution verwendet eine Paketverwaltung, um Software zu installieren, deinstallieren oder zu aktualisieren. Bei Debian und abgeleiteten Distributionen wird APT [1] verwendet.

6.1 Quellen einrichten

Damit APT weiß, von wo die Pakete heruntergeladen/kopiert werden können, werden in der Datei `/etc/apt/sources.list` und/oder in `.list`-Dateien unter `/etc/apt/sources.list.d/` die Quellen festgelegt.

```
1  #Binaer-Pakete
2  deb http://ftp.de.debian.org/debian/ wheezy main non-free contrib
3  deb http://security.debian.org/ wheezy/updates main contrib non-free
4  deb http://www.deb-multimedia.org wheezy main non-free
5
6  #Quellcode-Pakete
7  deb-src http://ftp.de.debian.org/debian/ wheezy main non-free contrib
8
9  #Lokale Paketquelle
10 deb file:/meinmirror/ wheezy main non-free contrib
11
12 #CD/DVD
13 deb cdrom:[Debian DVD]/  wheezy main non-free contrib
```

Hinter der Angabe der jeweiligen Art der Quelle ("*deb*" für fertige binäre Pakete, "*deb-src*" für Quellcode) folgen die Angabe der Quelladresse, der Version (hier wheezy + testing) und der Archivbereiche, aus denen Software geladen werden soll.

"*main*" beschreibt hierbei den Teil der Quelle, der ausschließlich Software enthält, die fest zur Distribution gehört. "*non-free*" enthält Software, die nach den Debian Free Software Guidelines [2] als unfrei gilt, und "*contrib*" schließlich enthält Inhalte, die zwar selbst nach dieser Richtlinie als frei gelten, aber von Software aus non-free abhängig sind.

6.2 Software verwalten

6.2.1 (De)Installation und Aktualisierung

- Informationen über installierbare Pakete aktualisieren: `apt-get update` ↵
- Installierbare Pakete suchen: `apt-cache search <Suchbegriff>` ↵

[1] `http://de.wikipedia.org/wiki/Advanced_Packaging_Tool`
[2] `http://de.wikipedia.org/wiki/Debian_Free_Software_Guidelines`

- Paket(e) installieren: `apt-get install <Paket1> <Paket2>` ↵
- Paketdatei(en) installieren: `dpkg -i <Paket1.deb> <Paket2.deb>` ↵
- Paket(e) entfernen: `apt-get remove <Paket1> <Paket2>` ↵
- Installierte Pakete aktualisieren: `apt-get upgrade` ↵
- Installierte Pakete aktualisieren (ggf. werden zusätzliche Pakete installiert): `apt-get dist-upgrade` ↵

6.2.2 Informationen zu Paketen

- Liste (de)installierter Pakete: `dpkg --get-selections` ↵
- Liste der installierten Pakete für die Benutzung mit `apt-get install` :

```
dpkg --get-selections | grep -v deinstall$ | tr -d '[:blank:]' | sed 's/install$//g' | awk -v ORS='' '{print($0" ")}'
```

- Alle Dateien in einem installierten Paket auflisten: `dpkg -L <Paket>` ↵
- Welches *installierte* Paket enthält Datei/Verzeichnis?: `dpkg -S <Datei/Verzeichnis>` ↵
- Informationen zu einem installierten Paket abfragen: `dpkg -s <Paketname>` ↵
- Paketinhalte von verfügbaren Paketen aktualisieren: `apt-file update` ↵
- Welches *verfügbare* Paket enthält Datei/Verzeichnis?: `apt-file search <Datei/Verzeichnis>` ↵

6.2.3 Problemlösung

Wurde die Installation von Paketen unterbrochen (z.B. wegen nicht erfüllter Abhängigkeiten nach einer manuellen Paketinstallation mittels `dpkg -i Paket.deb` ↵), so kann diese mit `apt-get install -f` ↵ wieder aufgenommen werden.
In seltenen Fällen (meist, wenn in der Paketquellenliste Quellen kombiniert werden, die nicht zusammenpassen) kann der "Holzhammer" nötig sein, um ein Paket dennoch zu installieren: `dpkg -i --force-all Paketname.deb` ↵
Dies sollte aber nur der letzte Lösungsansatz sein.
Wurde das Paket bereits mit APT heruntergeladen, so befindet es sich im Verzeichnis `/var/cache/apt/archives/`.

6.3 debconf

Einige Pakete bringen Konfigurationsdialoge mit. Die dort eingegebenen Daten werden in der debconf [3] abgelegt und dann von einem Skript im jeweiligen Paket verarbeitet. Möchte man die Konfiguration eines Paketes nachträglich ändern, so geschieht dies über den Befehl `dpkg-reconfigure <Paketname>` ↵ . Die debconf kann mittels `debconf-get-selections` ↵ (aus dem Paket "*debconf-utils*") ausgegeben bzw. in eine Datei umgeleitet (z.B. `debconf-get-selections > debconf.bak` ↵) werden. Einzelne Werte oder auch die komplette debconf können mit `debconf-set-selections <Debconf-Datei>` ↵ wiederhergestellt werden, wobei sich die Datei an dieselbe Syntax wie die Ausgabe von "debconf-get-selections" halten muß.

[3] http://de.wikipedia.org/wiki/Debconf_%28Software%29

6.4 Installation automatisieren

Betreut man eine Reihe von Rechnern oder Webservern, wird es lästig, da (Sicherheits-)Aktualisierungen regelmäßig eingespielt werden wollen. Für diese Zwecke gibt es Spezialisten wie z.B. das Softwareverteilungssystem m23 [4], die dem (genervten, nachlässigen, ...) Administrator die Arbeit erleichtern.

Möchte man die Systeme aber nur automatisch aktualisieren, reicht auch das Paket "*cron-apt*" . Nach der Installation muß es aber noch konfiguriert werden, da es ansonsten nur neue Pakete herunterlädt, aber nicht installiert. Im Verzeichnis `/etc/cron-apt/action.d` legt man dazu die Dateien mit dem folgenden Inhalt ab bzw. ändert deren Inhalt:

- 0-update

```
update -o quiet=2
```

- 3-download

```
autoclean -y
upgrade -d -y -o APT::Get::Show-Upgraded=true
```

- 5-upgrade

```
upgrade -y
```

- 9-notify

```
-q -q --no-act upgrade
```

Nun läuft cron-apt einmal täglich (bei Debian z.B. jeden Morgen um 4 Uhr), was aber auch in der Datei `/etc/cron.d/cron-apt` anders eingestellt werden kann.

In der Datei `/etc/cron-apt/config` stellt man bei den Parametern `MAILTO="<Mail>"` die eMail-Adresse desjenigen ein, der Informationen über den Abschluß der cron-apt-Aktionen zugeschickt bekommen soll und bei `MAILON="<Typ>"` die Art der Benachrichtigung. Hierbei kann "*<Typ>*" "*always*" sein, wenn man über jede Aktion informiert werden möchte oder "*upgrade*", um nur dann informiert zu werden, wenn Pakete aktualisiert wurden. Damit die Benachrichtigung funktioniert, muß allerdings ein funktionsfähiger Mailserver auf dem System vorhanden sein.

[4] `http://m23.sf.net`

Kapitel 7

Pakete bauen

Nur vorgefertigte Pakete zu benutzen wäre doch langweilig, oder? Also die Systemwerkzeuge angeschmissen und flugs ein Paket gebastelt.

7.1 Pakete aus Debian-Quellen

Am einfachsten bekommt man den Quelltext eines Debian-Paketes, indem man `apt-get source <Paketname>` ↵ in einem neuen Verzeichnis ausführt. APT lädt daraufhin den Quelltext herunter und entpackt ihn auch gleich. Damit dies funktioniert, müssen in der Paketquellenliste (siehe Seite 21) aber nicht nur die Binärquellen, sondern auch die Quelltextquellen eingetragen sein. Ein Aufruf von `apt-get build-dep <Paketname>` ↵ versorgt das System zudem mit allen Entwicklerpaketen, die benötigt werden, um das Paket zu bauen. Nachdem alles an Ort und Stelle ist, wechselt man in das Quell-Verzeichnis des Paketes `cd <Quellverzeichnis>` ↵ und startet den Kompilierungs- und Paketierungsvorgang mit `dpkg-buildpackage -us -uc` ↵ . Nach einiger Zeit sollten sich die Debian-Pakete (Endung `.deb`) im Verzeichnis darüber befinden und installieren lassen.
Doch wozu das Ganze? Manchmal kann es nützlich sein, ein Paket selbst zu erstellen, nachdem man einen Patch auf den Quelltext angewandt oder anderweitige Veränderungen vorgenommen hat.

7.2 Pakete aus Quelltexten

Häufiger dürfte der Fall sein, daß es die gewünschte Software nicht oder nur in einer älteren Version in der Paketquelle der Distribution gibt. Das Vorgehen läuft häufig (ggf. mit kleinen Abwandlungen) wie folgt ab:

- Quelltext herunterladen und in einem neuen Verzeichnis entpacken.
- In das Quelltextverzeichnis wechseln.
- `./configure --prefix=/usr` ↵
- `make` ↵
- Nachdem die Software erfolgreich kompiliert wurde, kann diese nun mit `checkinstall -D` ↵ in ein Debian-Paket verwandelt werden.

Die Installation der deb-Datei und das Entfernen erfolgt wie im Kapitel "Paketverwaltung" (siehe Seite 21) beschrieben.

Kapitel 8

Notfallkoffer

Jeder Administrator sollte – für den Fall eines Falles – einen "*Notfallkoffer*" bereithalten. Doch was sollte darin sein?

- Eine halbwegs aktuelle Knoppix-CD(-RW) [1], sowie ein Knoppix, das mittels UNetbootin [2] auf einen USB-Stick geschrieben wurde – denn es soll ja bekanntlich auch Rechner ohne optische Laufwerke geben...
- Genauso sollte auch eine Clonezilla [3]-CD, sowie ein USB-Stick nicht fehlen, um Festplatten und Partitionen zu kopieren, sichern und wiederherzustellen.
- Zum Partitionieren und Formatieren eignet sich GParted Live [4], welches als CD- und USB-Abbild verfügbar ist.
- Je nach Geschmack gibt es noch eine Reihe weiterer Live-Linuxe mit verschiedenen Schwerpunkten.

8.1 Ausfälle anderer Art

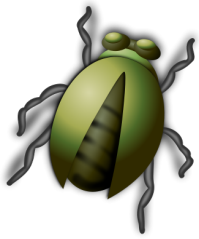
Ein Administrator sollte sich außerdem darüber Gedanken machen, ob – falls ihm etwas zustößt oder er aus anderen Gründen verhindert ist – es anderen Personen möglich ist, seine Aufgaben zu übernehmen. Für diesen Fall sollte er – je nach Wichtigkeit der administrierten Rechner – einen Notfall-Umschlag mit Paßwörtern oder eine verschlüsselte Datei an einem sicheren Ort deponieren, der den möglichen Vertretern jedoch bekannt sein sollte. Bei einigen Accounts sind hierbei natürlich die AGB des jeweiligen Anbieters zu beachten, die eine Paßwortweitergabe untersagen. Ein Ausweg kann hier das Anlegen zusätzlicher Administrator-Accounts sein.

[1] `http://www.knoppix.org/`
[2] `http://unetbootin.sourceforge.net/`
[3] `http://clonezilla.org/`
[4] `http://gparted.sourceforge.net`

Kapitel 9

Fehler identifizieren (und lösen)

Fehler sind lästig, aber letztendlich bei komplexen Systemen leider nicht zu vermeiden. Bei den meisten Fehlern ist man zum Glück nicht auf sich allein gestellt und das Internet und die Suchmaschine des Vertrauens (bzw. des geringsten Mißtrauens) bietet für fast jeden Fehler eine passende Lösung. Um diese zu finden sind gute Suchbegriffe nötig. Neben der Fehlersuche und Testreihen geht es in diesem Kapitel um die Gewinnung von möglichst guten Suchbegriffen.



9.1 Suchmaschinen füttern

Als Faustregel gilt, daß die Suchmaschine mit exakten Fehlermeldungen (ggf. bereinigt von persönlichen Informationen wie Rechner- oder Benutzername, etc.) gefüttert werden sollte. Zu der Fehlermeldung selbst sollte der Name des fehlerproduzierenden Programmes hinzugenommen werden. Sollten die Suchergebnisse zu umfangreich oder unpassend ausfallen, so ist es sinnvoll, die Suchanfrage um den Namen der Distribution (Debian, Ubuntu, ...) und ggf. den Releasenamen (z.B. Wheezy, Quantal, ...) oder ganz allgemein um "Linux" zu erweitern.

9.2 "Programmgesprächigkeit" erhöhen

Viele (Kommandozeilen-)Programme bieten die Möglichkeit, diese "gesprächiger" zu machen und so weitergehende Informationen über den Programmablauf (inklusive genauerer Fehlermeldungen) zu gewinnen. Bei vielen Programmen verbirgt sich diese Funktionalität hinter einer "*verbose*"-Option, die oft mit `-v` oder `-V` aktiviert wird. Alternativ kann diese auch unter den Begriff "*debug*" oder "*log*" fallen. Der genaue Name der Option und weiterer Parameter stehen entweder in der Manpage oder können über die eingebaute Hilfefunktion (siehe Seite 11) des Programmes abgerufen werden.

9.3 Systemfehler finden

Eine wichtige Anlaufstelle für Systemfehler sind die Ausgabe von `dmesg ↵` und die Dateien unter `/var/log` und in den (ggf. existierenden) Unterverzeichnissen. Hier befinden sich Logdateien diverser Programme. Bei dieser Vielzahl an Logdateien kann aber schon mal die Übersicht verlorengehen. Hier hilft der Einzeiler

`ls -lt /var/log/ | head ↵` ,

der die Dateien nach der Änderungszeit sortiert, wobei die neuesten ganz oben stehen. Dies sollte man möglichst schnell nach dem Auftreten eines Fehlers machen, da ansonsten andere Programme neuere Logdateien schreiben könnten und somit die interessante Datei aus der Auflistung "herauswandert". Schließlich beschränkt `head ↵` die Ausgabe auf 10 Zeilen.

Die so gefundene Datei sollte nun ausgiebig mit `less` oder `nano` begutachtet werden, um Fehlermeldungen und andere Hinweise auf das Problem zu finden.

9.4 Was läuft?

Auf einem Linuxrechner läuft eine Vielzahl von Programmen gleichzeitig. Welche diese sind, verrät `ps -A ⏎` . Sollte sich ein (eigentlich) laufendes Programm nicht darunter befinden, so sollte man diesem seine besondere Aufmerksamkeit schenken.

Möchte man ein Programm (bzw. Prozeß) beenden, so geschieht dies durch `kill <Prozeßnummer> ⏎` bzw. `killall <Prozeßname> ⏎` . Sollte sich der Prozeß nicht zum Beenden bewegen lassen, so hilft der zusätzliche Parameter `-9` . Also: `kill -9 <Prozeßnummer> ⏎` bzw. `killall -9 <Prozeßname> ⏎`

9.5 Wer sendet?

Genauso interessant wie die Frage nach den laufenden Programmen kann die Frage danach sein, wer wohin sendet bzw. wer auf welchen Ports lauscht. Die Frage beantwortet `netstat -a ⏎`

9.6 Programme belauschen

NSA läßt grüßen: Das Programm `strace ⏎` belauscht andere Programme und erstellt ein detailliertes Protokoll. Da strace eine Menge Informationen sammelt, sollten diese von der Fehlerausgabe direkt in eine Datei umgeleitet werden. Ein Aufruf könnte folgendermaßen aussehen:

```
strace -f -o /tmp/nsa.log <zu belauschendes Programm zzgl. Paramerter> ⏎
```

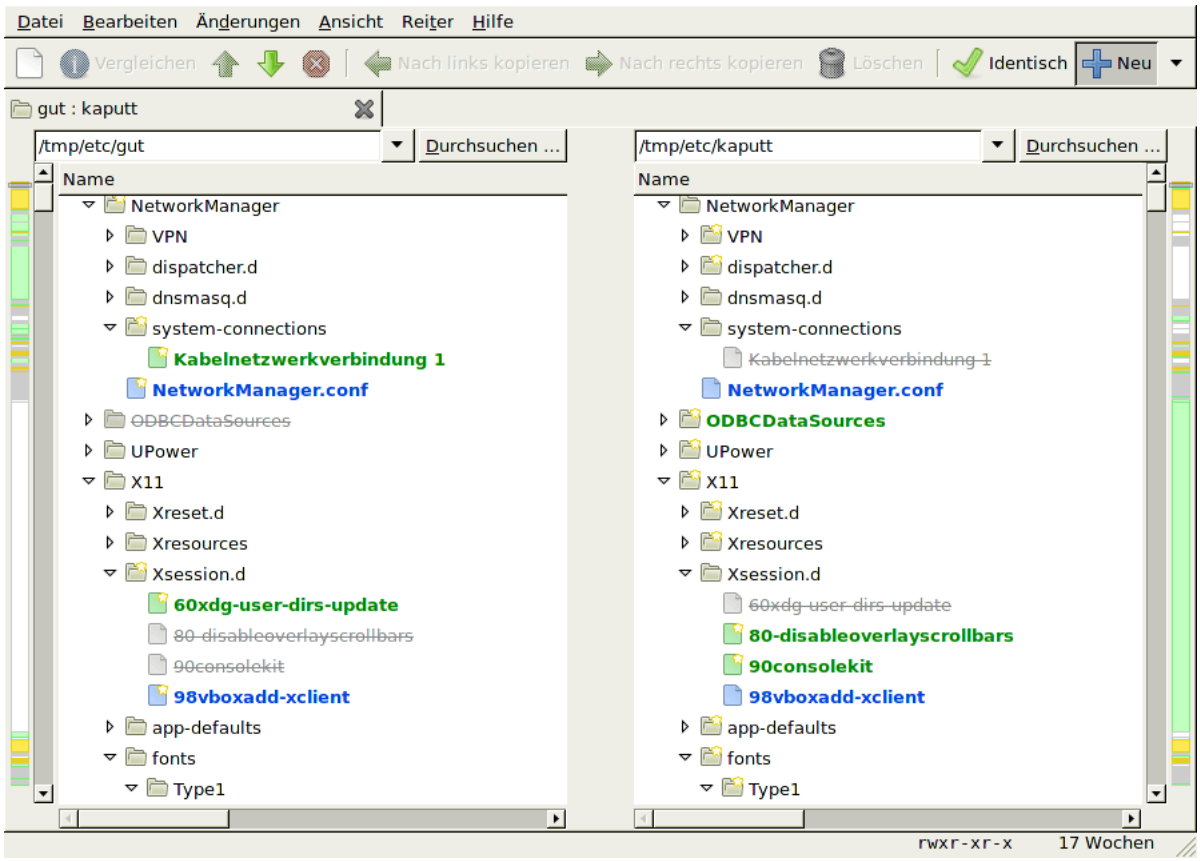
Dieser Aufruf startet das angegebene Programm und schreibt alle Informationen (`-f` : auch der Kindprozesse) in die Datei `/tmp/nsa.log`. Diese Datei sollte man nun genau begutachten. Geht es z.B. um Fehler mit nicht gefundenen Dateien, so kann man diese in der Logdatei leicht mit den Suchbegriffen "*ENOENT*" und "*open*" finden.

9.7 Unterschiede zwischen Systemen

Es kommt leider auch immer wieder vor, daß zwei eigentlich gleiche Systeme sich unterschiedlich verhalten oder eines sogar mit Fehlermeldungen um sich schmeißt und einfach nicht das tut, was es tun sollte. Kann dann keine Neuinstallation vorgenommen werden (oder steht der Ehrgeiz des Administrators dieser Lösungsmöglichkeit entgegen), so kann ein Vergleich beider Maschinen auf das dahinterliegende Problem hinweisen.

Meist liegt dann das Problem in der Konfiguration und somit im Verzeichnis `/etc`. Für die Analyse sollte man dann die beiden `/etc`-Verzeichnisse auf den eigenen Arbeitsplatzrechner kopieren (z.B. mit rsync oder scp (siehe Seite 20)) und dann mit einem Vergleichswerkzeug wie "*Meld*" [1] analysieren.

[1] `http://meldmerge.org/`


Datei Bearbeiten Änderungen Ansicht Reiter Hilfe
Vergleichen
Nach links kopieren
Nach rechts kopieren
Löschen
Identisch
Neu
gut : kaputt
/tmp/etc/gut
Durchsuchen ...
/tmp/etc/kaputt
Durchsuchen ...
Name
NetworkManager
VPN
dispatcher.d
dnsmasq.d
system-connections
Kabelnetzwerkverbindung 1
NetworkManager.conf
ODBCDataSources
UPower
X11
Xreset.d
Xresources
Xsession.d
60xdg-user-dirs-update
80-disableoverlayscrollbars
90consolekit
98vboxadd-xclient
app-defaults
fonts
Type1
rwxr-xr-x
17 Wochen

Kapitel 10

Bootmanager und Live-Linux

Der Bootmanager (meist kommt GRUB [1] zum Einsatz) sorgt dafür, daß ein oder auch mehrere Betriebssysteme starten können. Aus verschiedenen (eher seltenen) Gründen kann es dazu kommen, daß der Bootmanager beschädigt wird und somit kein Booten mehr möglich ist. Eine Neuinstallation des Betriebssystems ist normalwerweise nicht notwendig. Mit ein bißchen "Adminmagie" läßt sich der GRUB-Bootmanager "wiederbeleben".

10.1 Live-Linux booten

In diesem Abschnitt geht es um das Booten einen Live-Linux und den Wechsel auf das System der Festplatte. Dieses Vorgehen ist nicht nur bei GRUB-Problemen nützlich, sondern auch bei anderen Fehlern, bei denen das installierte System nicht mehr gestartet werden kann.

- Live-Linux booten (z.B. Knoppix [2]) und bei Knoppix direkt beim Startbildschirm `knoppix 2 ↵` (Textmodus) eingeben. Bei anderen Live-Linuxen eine root-Shell starten.
- Nun die richtige Festplatte mit der Partition des Betriebssystems ermitteln: `parted /dev/sda print ↵` (wenn dies nicht die richtige Festplatte sein sollte: Weiter mit sdb, sdc, ...).
- Die Partition des Betriebssystems (z.B. an Größe und dem Dateisystem (ext3, ext4, ...) zu erkennen) finden. Der komplette Gerätename der Partition setzt sich aus `/dev/<Festplattengerät><Partitionsnummer>` zusammen. Also z.B. `/dev/sda1`
- Im Live-Linux ein Verzeichnis zum Einhängen anlegen: `mkdir /mnt/linux ↵`
- Die Partition im Verzeichnis einhängen: z.B. `mount /dev/sda1 /mnt/linux ↵`
- Das aktuelle System wechseln: `chroot /mnt/linux ↵`
- BASH starten: `bash ↵`
- Sicherstellen, daß das (neue) root-Dateisystem wirklich beschrieben werden kann: `mount -o remount,rw / ↵`
- Die sys- und proc-Pseudodateisysteme einhängen: `mount /proc; mount /sys ↵`

[1] http://de.wikipedia.org/wiki/Grand_Unified_Bootloader
[2] http://www.knoppix.org/

10.2 GRUB reparieren

Dies ist wirklich GRUB-spezifisch...

- Falls `/dev` leer sein sollte oder bei GRUB-Fehlern: `cd /dev; MAKEDEV generic` ↵
- GRUB neu in Systemplatte (hier `/dev/sda`) installieren: `grub-install -f /dev/sda` ↵
- GRUB-Einstellungen aktualisieren: `update-grub` ↵

10.3 Aufräumarbeiten

Das hier sollte am Schluß einer jeden Live-Linux-Sitzung gemacht werden:

- Sicherstellen, daß die Daten auf die Festplatte geschrieben wurden: `sync` ↵
- Rechner neu starten: `reboot -f` ↵
- CD/DVD aus dem Laufwerk nehmen bzw. USB-Stick abziehen.

Kapitel 11

Paßwort vergessen

Es sollte natürlich nicht vorkommen (vor allem nicht beim Administrator), aber in der Praxis geschieht es angeblich öfter, daß Paßwörter vergessen werden...
Wenn es sich um die Paßwörter von lokalen Benutzern (auch "root" gehört dazu) handelt, ist es nicht schwer, ein neues Paßwort zu setzen. Das Vorgehen ist dabei – bis auf den GRUB-spezifischen Teil – wie im Kapitel "Bootmanager und Live-Linux" (siehe Seite 33). An Stelle des GRUB-Teiles tritt ein simpler Aufruf von `passwd` ↵ , wenn das root-Paßwort neu gesetzt werden soll. Hat hingegen ein Benutzer sein Paßwort vergessen, so wird dieses mit `passwd <Benutzername>` ↵ geändert.

Kapitel 12

Lizenz und weitere Informationen

12.1 Lizenz

Namensnennung-NichtKommerziell 3.0 Deutschland
Sie dürfen:

- den Inhalt vervielfältigen, verbreiten und öffentlich aufführen
- Bearbeitungen anfertigen

Zu den folgenden Bedingungen:

Namensnennung. Sie müssen den Namen des Autors/Rechtsinhabers nennen.

Keine kommerzielle Nutzung. Dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.

- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter die dieser Inhalt fällt, mitteilen.
- Jede dieser Bedingungen kann nach schriftlicher Einwilligung des Rechtsinhabers aufgehoben werden.

Die gesetzlichen Schranken des Urheberrechts bleiben hiervon unberührt. Hier ist eine Zusammenfassung des Lizenzvertrags in allgemeinverständlicher Sprache: `http://creativecommons.org/licenses/by-nc/3.0/de/legalcode`

12.2 Weitere Informationen

Weitere Informationen zu den Schulungs- und Beratungsangeboten von goos-habermann.de erhalten Sie unter `http://www.goos-habermann.de/index.php?s=SchulungBeratung`. Die Entwicklungsdienstleistungen finden Sie unter `http://www.goos-habermann.de/index.php?s=Entwicklung`.