HOCHSCHULE FURTWANGEN UNIVERSITY HFU

Projektarbeit

in der Fakultät

Informatik

Access Point und Router mit embedded Board Banana Pi R1

Referent	:	Dr. Jiri Spale
Vorgelegt am	:	??.??.2017
Vorgelegt von	:	Bastian Elias Jakob Jonas Tom

Abstract

English Abstract

Diese Dokumentation ist Teil des Projektes "Access Point und Router mit embedded Board Banana Pi R1", welches im Sommersemester 2017 an der Hochschule Furtwangen abgehalten wird. Das Projekt besteht daraus, das Vorgängerprojekt mit eigenen Implementierungen zu erweitern. Projektziele sind, das testen passender Betriebssysteme, sowie die Implementierung verschiedener Funktionen wie VPN, Radius, Samba, einem Mailserver und einer Displaystatusanzeige.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

1 Projektübersicht

1.1 Kontext

1.2 Ziel

2 Betriebssystemeübersicht

2.1 OpenWRT

2.2 Bananian

2.3 IPFire

2.4 Armbian

3 Implementierung

3.1 Backup und Wiederherstellung des Betriebssystems

Um die Implementationen und Konfigurationen des Pi's abzusichern, wird eine Backup Lösung eingesetzt. Die Sicherung relevanter Daten soll hierbei einem möglichen Defekt oder Inkompatibilität durch Updates o.ä. entgegenwirken.

'FauBackup' und 'gitbac' bieten hierbei eine Lösung mittels externer Programme an. Debian bietet jedoch bereits standardmäßig einige Aufrufe welche zur Anlegung von Backups verwendet werden können. Diese wurden im Folgenden in Ubuntu anhand eines Bash Skripts zur Automatisierung getestet und implementiert.

3.1.1 dd

Als erste Ausführung wurde der Aufruf 'dd' verwendet. Hierbei wird der Inhalt der gesamten SD Karte als Image Datei abgespeichert.

1.

Übersicht der vorhandenen Dateisysteme mittels des Terminalaufrufs 'df'

```
bastian@bastian-VirtualBox:~$ df
Dateisystem    1K-Blöcke   Benutzt Verfügbar Verw% Eingehängt auf
udev             497668         4    497664    1% /dev
tmpfs            101768       900    100868    1% /run
/dev/sda1       9156984   5217368   3451424   61% /
none                  4         0         4    0% /sys/fs/cgroup
none               5120         0      5120    0% /run/lock
none             508832        76    508756    1% /run/shm
none             102400        56    102344    1% /run/user
none          455712764 373044796  82667968   82% /media/sf_OrdnerWindows
bastian@bastian-VirtualBox:~$
```

2.

Terminalaufruf für den Backupprozess

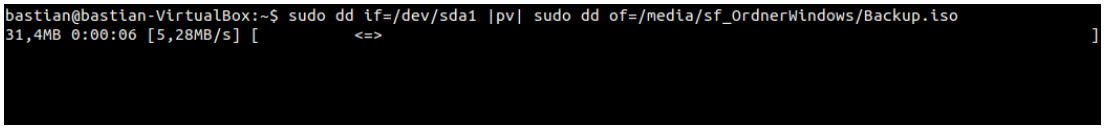
```
sudo dd if=INPUTPARTITION of=OUTPUTFILE
```

sudo	->	Backupprozess benötigt root-Rechte
dd	->	bit-genaues Kopieren der Dateien
if=FILE	->	Die Datei oder Partition welche integriert wird
of=FILE	->	Die Output Datei welche angelegt wird

3.

Optionale Nutzung von Terminalaufruf 'pv' um den Fortschritt des Backup Prozesses zu sehen Die mögliche Restzeit lässt sich nur durch das Hinterlegen der Größe der Partition anzeigen.

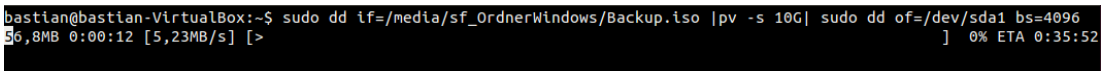
```
sudo dd if=INPUTPARTITION |pv| sudo dd of=OUTPUTFILE
```



4.

Wiederherstellen eines hinterlegten Backups läuft ähnlich wie der ursprüngliche Prozess ab.

```
sudo dd if=OUTPUTFILE |pv| of=INPUTPARTITION
```



3.1.2 rsync

Da der Vorgang mittels 'dd' als suboptimal angesehen wird, wurde alternativ der Aufruf 'rsync' verwendet. Dieser bietet die Möglichkeit eines inkrementellen Backups wodurch die Dauer des Prozesses erheblich reduziert werden kann. Hierbei werden die Größe und die Änderungszeit der Dateien in Quelle und Ziel miteinander verglichen. Eine Aktualisierung findet demnach nur statt, wenn Unterschiede vorzufinden sind.

```
rsync -aAXv --delete --exclude={"/dev/*","/proc/*","/sys
    /*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found
    "} / /path/to/backup/folder
```

rsync	->	Kopieren der Dateien
-aAX	->	Übertragung im Archiv Modus wodurch alle symbolischen Verweise beibehalten werden
–delete	->	Dateien die im Ursprungsverzeichnis nicht mehr existieren werden im Zielverzeichnis ebenfalls gelöscht
–exclude	->	Dateien werden ausgelassen

Wiederherstellen des Rsync Backups durch folgenden Befehl:

```
rsync -aAXv /path/to/backup/location/* /mount/point/of/
    new/install/ --exclude={/dev/*,/proc/*,/sys/*,/tmp/*,/
    run/*,/mnt/*,/media/*,/lost+found,/home/*}
```

3.1.3 tar

Eine weitere Anwendungsmöglichkeit bietet die 'tar' Archivierung. Vorteil dieses Aufrufs ist, dass durch Angabe von Parametern die Berechtigungen aller zu sichernden Daten ebenfalls beibehalten werden und die Archivierung Speicherplatz spart.
Wechsel in das Backupverzeichnis dann:

```
tar -cpzf Backup.tar ORDNER
```

tar	-> Archivieren von Daten
-c	-> Archiv wird erzeugt (create)
-p	-> Berechtigungen beibehalten (privilige)
-z	-> Zusätzliche Komprimierung mit gzip
-f	-> Archiv in Datei schreiben (finish)

3.1.4 Automatisierung mittels Bash-Skript

Damit der Nutzer die Aufrufe nicht händisch zu bestimmten Zeiten ausführen muss, wurden zwei Bash-Skripte zur Automatisierung geschrieben. Es gibt ein monatliches Backup mittels (tar) und wöchentliche inkrementelle Backups (rsync) auf die zurückgegangen werden kann.

Um das Zeitintervall der Backups einzustellen wird der 'Cron' Dienst verwendet. Hiermit können Skripte und Programme zu festgelegten Zeiten gestartet werden. Wenn ein hinterlegter Job täglich zu einer bestimmten Uhrzeit ausgeführt wird muss allerdings auch der Rechner zu dem Zeitpunkt aktiv sein. Ist dies nicht der Fall, startet der Prozess nicht. Um dies zu umgehen wird 'Anacron' verwendet. Durch ablegen des Skripts in eines der entsprechenden Verzeichnisse wird der Prozess entsprechend ausgeführt.

/etc/cron.hourly/	- Stündlich ausführen
/etc/cron.daily/	- Täglich ausführen
/etc/cron.weekly/	- Wöchentlich ausführen
/etc/cron.monthly/	- Monatlich ausführen

```
Woechentliches Backup wird ausgefuehrt. Diesen Vorgang bitte nicht abbrechen!
sending incremental file list
deleting security_update.log
rsync: symlink "/media/sf_OrdnerWindows/Backup/initrd.img" -> "boot/initrd.img-3.13.0-119-generic" failed: Read-only file system (30)
rsync: symlink "/media/sf_OrdnerWindows/Backup/initrd.img.old" -> "boot/initrd.img-3.13.0-117-generic" failed: Read-only file system (3
0)
rsync: symlink "/media/sf_OrdnerWindows/Backup/vmlinuz" -> "boot/vmlinuz-3.13.0-119-generic" failed: Read-only file system (30)
rsync: symlink "/media/sf_OrdnerWindows/Backup/vmlinuz.old" -> "boot/vmlinuz-3.13.0-117-generic" failed: Read-only file system (30)
./
```

3.2 Update des Banana Pi

Ziel war es das Betriebssystem und alle Programme immer auf dem aktuellsten Stand zu halten. Hierbei kann es jedoch zu Inkompatibilität bestimmter Funktionen oder Konfigurationen kommen. Daher wurde die Umsetzung auf die relevantesten Updates (Sicherheitsupdates) reduziert. Im Normalfall können mittels des Konsolenaufrufs 'apt-get update' die Updateliste und mit 'apt-get upgrade' die Programm Pakete selbst aktualisiert werden. Durch Nutzung des Aufrufs 'unattended-upgrade' wird auf Sicherheitsupdates des Systems überprüft und diese anschließend installiert.

Durch 'unattended-upgrade –dry-run -d' wird auf Verfügbarkeit von Updates geprüft ohne anschließende Installation. Nach jedem Durchgang wird eine Logdatei in /var/log/unattended-upgrades/ angelegt welche genauere Informationen zu den aktualisierten Dateien liefert.

3.3 Implementierung einer Displaystatusanzeige

Zur besseren Übersicht des Netzwerktraffics als auch der Ressourcen des Banana Pi sollte eine Displaystatusanzeige implementiert werden. Der Aufruf 'htop' bietet eine Übersicht aller laufenden Prozesse und deren Ressourcennutzung. 'iftop' zeigt die Netzwerkinterfaces und die eingehende und ausgehende Kommunikationen. Da das Terminal jedoch nur einen der Befehle zu einem Zeitpunkt ausüben kann wird ein Terminal-Multiplexer verwendet. Zur Auswahl stehen hierbei 'Terminator', 'screen', und 'Tmux'. Aufgrund der geringen Einarbeitungszeit und einfachen Anwendbarkeit wurde letzteres zur Implementation ausgewählt. Alle Multiplexer bieten die Möglichkeit Sitzungen zu erstellen. Leider kann dies nicht zur Implementierung der Displaystatusanzeige verwendet werden, da die Sitzung beim Herunterfahren des Betriebssystems gelöscht wird. Daher wird zum Systemstart ein Bash-Skript eingesetzt, welches automatisch die benötigten Fenster zur Überwachung anlegt.

3.4 Routing, VLANs

3.5 WLAN

3.6 Radius

3.7 Mailserver

3.8 Samba

4 Benutzeranleitung

5 Lerneffekt - Eigenbewertung

6 Ausblick

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Stellen der Arbeit die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Furtwangen, den ??.??.2017