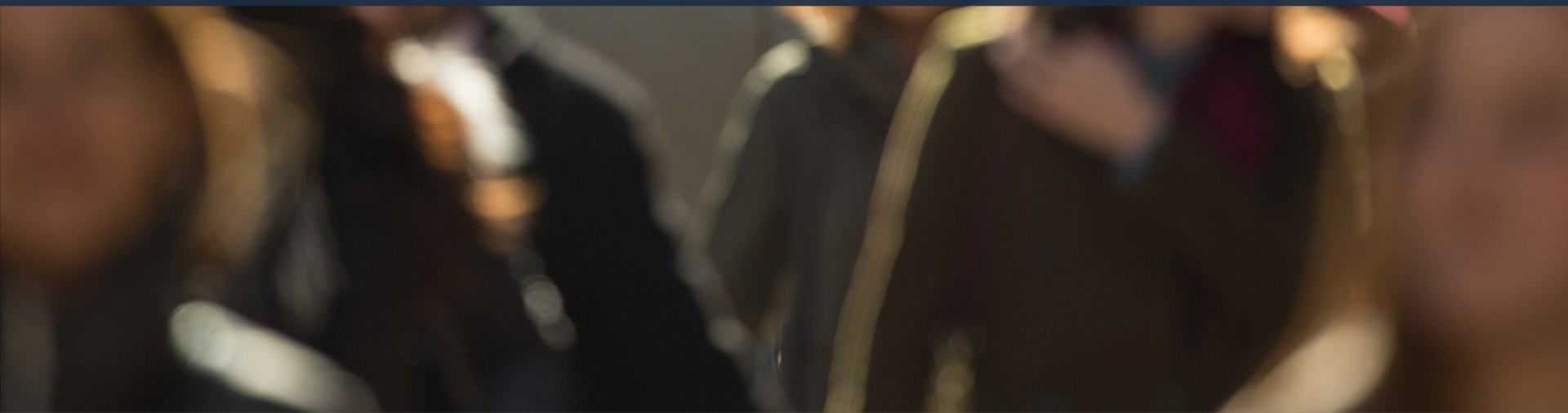
ownCloud auf dem Raspberry Pi 2



Daniel Laczi für Offn is!
Schwabach, 03.10.2015

Über mich

Daniel Laczi
Technical Consultant für Security & IT Solutions
E-Mail: daniell1 (at) t-online.de

Ziel

- **Synchronisation** von
  - Kalender und
  - Kontakten zwischen
  - allen Endgeräten (hier: Linux und Android)
- **Sharing** von Dateien, wie z. B.
  - Bilder mit Freunden
- Möglichst **einfache oder automatische Administration**
- **Low-Budget-Lösung** (geringe Anschaffungs- und Betriebskosten)

Ausgangssituation

- Kalender- und Kontaktsynchronisation bei E-Maildiensten meist nur mittels **Active Sync**
- **Schlechte Unterstützung** von Active Sync unter **Linux**
- **Kalendersharing zwischen Benutzern** häufig **kostenpflichtig**

Lösung

- **Raspberry Pi 2**
  - Preiswerter Minicomputer
  - Geringe Betriebskosten (abgesehen von der Installation und Administration)
- **ownCloud**
  - Offene Protokolle zur Kalender- und Kontaktsynchronisation
    - CalDAV
    - CardDAV
  - Sharing von Dateien
  - Open Source Software (keine Anschaffunskosten, jedoch Administrationsaufwand)

Wichtige Anmerkungen

- Bei der im Vortrag gezeigten Konfiguration befindet sich die **ownCloud im Internet**! Der Server
  - kann gehackt werden und
  - benötigt Administration und Pflege
- Jeder ist **selbst** für seinen Server und Internetanschluss **verantwortlich**!
- Der Vortrag behandelt **keine Lösung für Firmen** (Supportverträge, professionelle Dienstleister etc. unbedingt notwendig!)

Los geht's!

Agenda

- Auswahl einer Distribution
- Installation der ownCloud
- Konfiguration der ownCloud
- SSL Aktivierung
- Applikationen zur Synchronisation
- Externe Erreichbarkeit
- Automatisches Update
- Backup
- Hardening

Kommandos

- **Kommandos für die selbstständige Installation zu Hause**
- **local#** *<Befehl für Desktoprechner/Laptop>*
- **owncloud#** *<Befehl für ownCloud>*
- Statt des verwendeten Editors **vi** kann auch z. B. nano oder jeder andere Edior verwendet werden
  - **ownlcoud#** *apt-get install nano*

Auswahl einer Distribution

- **Sjoerd Simons' Debianimage**
  - Bis auf das Paket flash-kernel sind alle Pakete aus dem Debian Repository
  - ownCloud aus dem Debian stable Repository ist getestet und maintained
  - Automatische Updates möglich
  - Support im Debian IRC Channel *#debian*
- **Natürlich können auch andere Distributionen verwendet werden!**

Auswahl einer Distribution

- Image herunterladen
- Image auf SD Karte kopieren
  - **local#** *dd if=/path/to/image of=/dev/sdX*

  Anmerkung
  Das Image enthält keinen “Installationsmechanismus”, weshalb nachstehende Aufgaben durchgeführt werden sollten (siehe Kommentare von nadu auf der Downloadseite; getestet mit Imageversion jessie-rpi2-20150705)
- SD Karte neu einlesen und zweite Partition vergrößern
  - z. B. mittels **local#** *gparted*
- Per SSH verbinden mit Nutzer **root** und Passwort **debian**
  - **local#** *ssh root@<IP-Adresse oder hostname=jessie-rpi>*
- Rootpasswort ändern
  - **jessie-rpi#** *passwd*

Auswahl einer Distribution

- Hostname ändern
  - Hosname in /etc/hosname und /etc/hosts ändern (alternativer Editor zu vi: nano)
    - **jessie-rpi#** *vi /etc/hostname*
    - **jessie-rpi#** *vi /etc/hosts*
  - exim4 neukonfigurieren (Hostname ändern, ansonsten Standardeinstellungen verwenden)
    - **jessie-rpi#** *dpkg-reconfigure exim4-config*
  - Überprüfen, welche weiteren Dateien ggf. noch geändert werden müssen
    - **jessie-rpi#** *grep -R jessie-rpi /etc/**
  - System neustarten
    - **jessie-rpi#** *reboot*

Auswahl einer Distribution

- Mittels Apt-Pinning verhindern, dass das Paket flash-kernel aus dem offiziellen Repo aktualisiert wird
  - Neue Datei anlegen (owncloud = ownCloud)
    - **owncloud#** *touch /etc/apt/preferences.d/flash-kernel*
  - Folgende Zeilen in die Datei einfügen
    - **owncloud#** *vi /etc/apt/preferences.d/flash-kernel*

      Package: flash-kernel

      Pin: origin repositories.collabora.co.uk

      Pin-Priority: 1000
  - Überprüfen
    - **owncloud#** *apt-cache policy flash-kernel*

Auswahl einer Distribution

- Neue SSH Keys erzeugen
  - Alte Schlüssel entfernen
    - **owncloud#** *rm /etc/ssh/ssh_host_*_key**
  - Neue Schlüssel erzeugen
    - **owncloud#** *dpkg-reconfigure openssh-server*

  Hinweis
  Bei erneuter Verbindung mittels SSH zur ownCloud bekommt man einen Fehler. Hierfür auf dem Desktoprechner den alten Eintrag aus der Meldung in /home/nutzername/.ssh/kown_hosts löschen.

- Sprache und Region, sowie Zeitzone auf Deutsch stellen (den Anweisungen folgen)
  - **owncloud#** *dpkg-reconfigure locales*
  - **owncloud#** *dpkg-reconfigure tzdata*

Auswahl einer Distribution

- Repositories bearbeiten
  - Editor öffenen und nachstehende Zeile **entfernen**
    - **owncloud#** *vi /etc/apt/sources.list*

      deb [trusted=yes] https://repositories.collabora.co.uk/debian/ jessie rpi2

- Zusätzliche Repositories aktivieren
  - Editor öffnen und nachstehende Zeilen einfügen
    - **owncloud#** *vi /etc/apt/sources.list*

      deb https://repositories.collabora.co.uk/debian/ jessie rpi2

      deb http://ftp.debian.org/debian/ jessie contrib non-free

      deb http://security.debian.org/ jessie/updates main contrib non-free

      deb http://ftp.debian.org/debian/ jessie-updates main contrib non-free

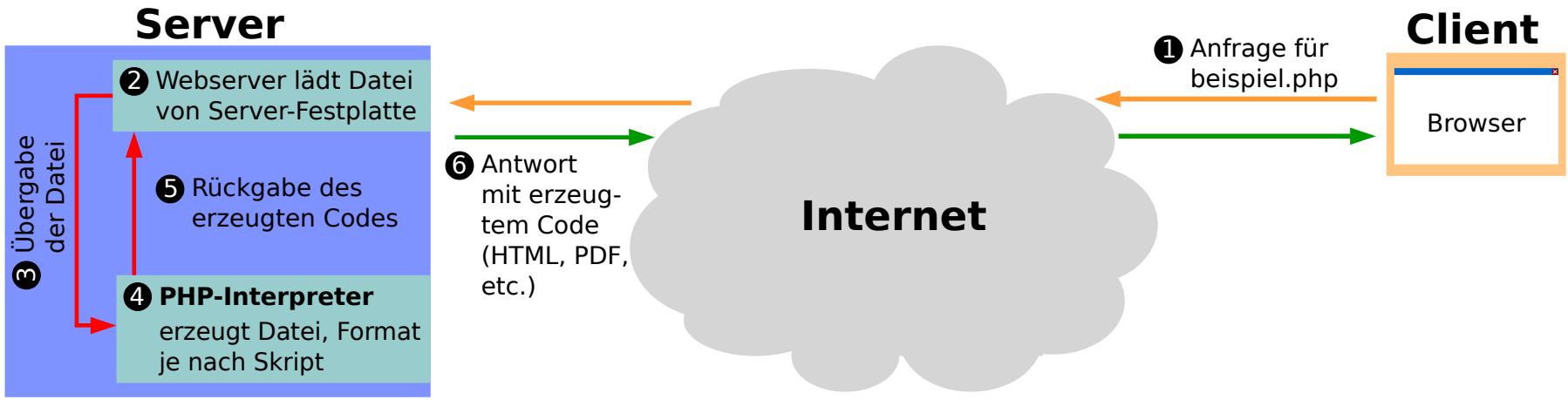
- Nun kann ein Update des Systems durchgeführt werden
  - **owncloud#** *apt-get update*
  - **owncloud#** *apt-get dist-upgrade*

Installation der ownCloud

- **Benötigte Komponenten**
  - Webserver (Standard: Apache)
  - Datenbank (Standard: MySQL)
  - PHP
- Der **Webserver** und die **Datenbank** müssen **vor den ownCloud Paketen installiert** werden

Installation der ownCloud

- **Exkurs: Funktionsweise PHP**
  - ownCloud Seiten werden dynamisch vom **PHP-Interpreter** erzeugt
  - **Sicherheitslücken** im PHP Interpreter können es Angreifern ermöglichen Code auf dem Server auszuführen (siehe Hardening)



Quelle: https://de.wikipedia.org/wiki/Datei:PHP_funktionsweise.svg

Installation der ownCloud

- SQL-Server und ownCloud installieren (installiert Apache und PHP automatisch mit)
  - **owncloud#** *apt-get install mysql-server*
  - **owncloud#** *apt-get install owncloud*
- SQL Benutzer für ownCloud anlegen
  - **owncloud#** *mysql -h localhost -u root -p*
    - **mysql>** *CREATE USER 'benutzername'@'localhost' IDENTIFIED BY 'Passwort';*
    - **mysql>** *GRANT ALL ON owncloud.* TO 'benutzername'@'localhost';*
    - **mysql>** *SHOW GRANTS FOR 'benutzername'@'localhost';*
    - ggf. **mysql>** *REVOKE ALL PRIVILEGES ON *.* FROM 'benutzername'@'localhost';*
- ownCloud über Weboberfläche konfigurieren
  - IP-Adresse der ownCloud im Webbrowser (z. B. Firefox) eingeben
    - Data folder: /usr/share/owncloud/data (Standard)
    - Database name: owncloud (Name der Datenbank)
    - Database host: localhost

Konfiguration der ownCloud

- **Konfigurationsmöglichkeiten auf der Weboberfläche als Administrator**
  - Anlegen weiterer Benutzer und Gruppen
  - Festlegen von Quotas
  - Sharing
    - Zwischen Benutzern
    - Server-to-Server
    - Externe Links
  - Enforce HTTPS
  - E-Mailserver

Demo

Konfiguration der ownCloud

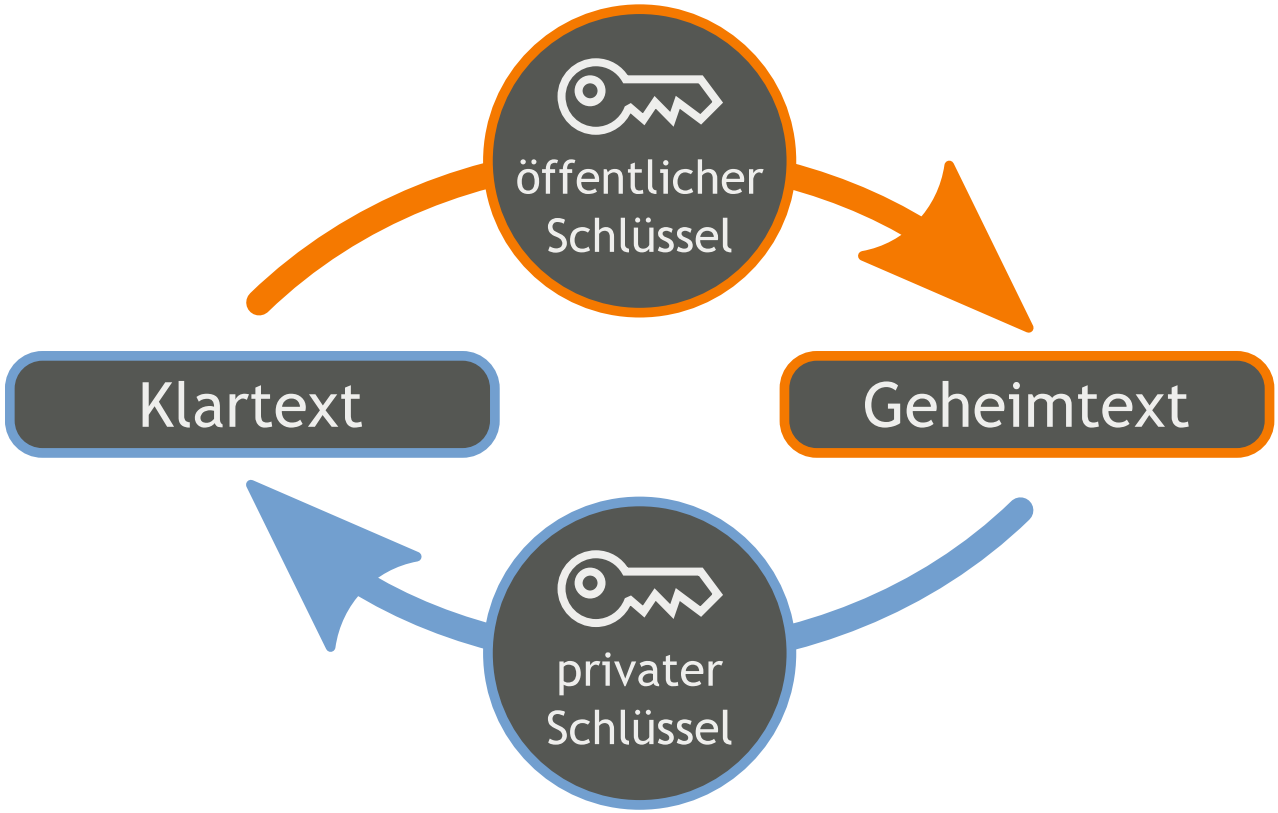
- Generell: **nur benötigte Funktionen aktivieren**
- Für die **normale Verwendung** empfiehlt es sich **Nutzer** zu verwenden, die **keine Administrationsrechte** besitzen
- **Quotas** sind sinnvoll, damit der Speicher nicht vollläuft (siehe auch Backup)
- **Enforce HTTPS** sollte bei externer Erreichbarkeit unbedingt eingeschaltet werden
- Der **E-Mailserver** wird benötigt, damit z. B. externe Links versendet werden können

SSL Aktivierung

- **Voraussetzungen** für eine verschlüsselte Verbindung zur ownCloud
  - **SSL-Zertifikat**, ggf. **mit externem Hostnamen**
  - Aktivierung des SSL-Apachemoduls
  - Import des Zertifikats auf allen Endgeräten (hier: Firefox und Android)
- **Signierung des SSL-Zertifikats**
  - Durch öffentliche Root CA (normalerweise kostenpflichtig)
  - Selbstsignierung (kostenlos)

SSL Aktivierung

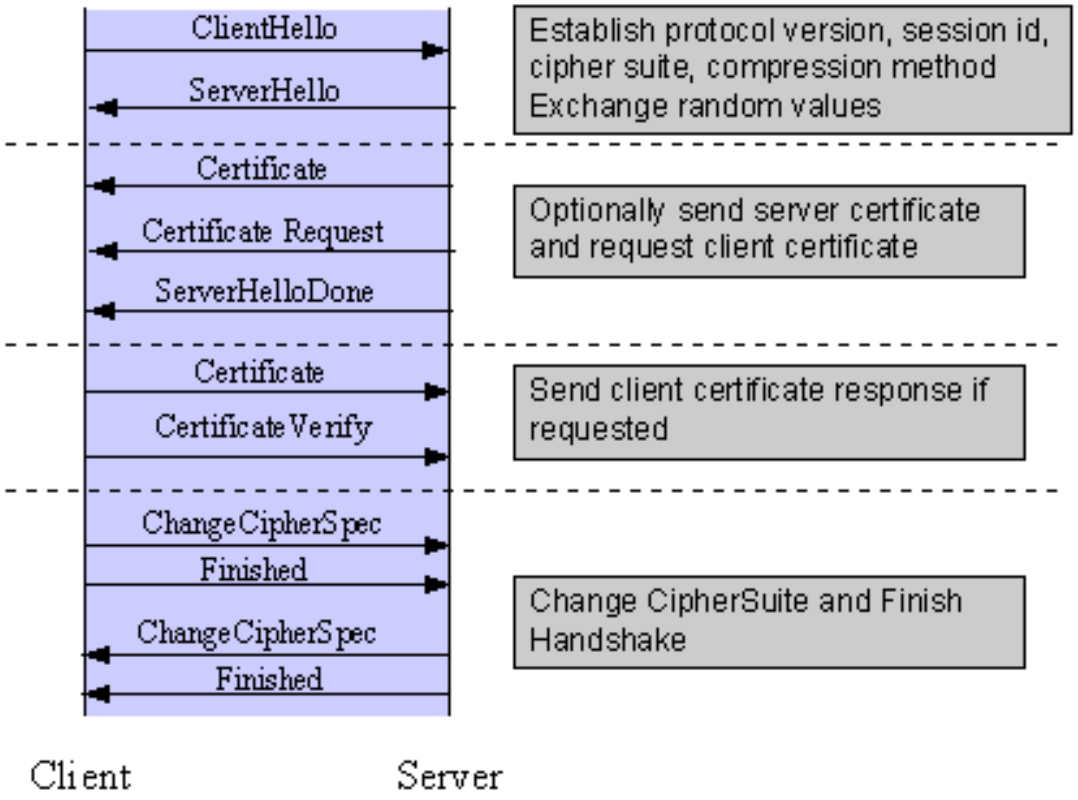
- **Exkurs: Asymmetrische Verschlüsselung**



Quelle:
https://upload.wikimedia.org/wikipedia/commons/a/a2/Orange_blue_public_key_cryptography_de.svg

SSL Aktivierung

- **Exkurs: SSL Handshake** (stark vereinfacht)



Quelle: http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html

SSL Aktivierung

- In diesem Vortrag wird ein **selbstsigniertes Zertifikat** verwendet

Achtung
Dieses Zertifikat enthält **keine Sperrinformationen**, d. h. Zertifikate können nicht zurückgezogen oder gesperrt werden (z. B. bei Diebstahl)!
Die **Gültigkeitsdauer** des SSL-Zertifikat sollte deshalb **nicht zu hoch** sein (hier 180 Tage = ½ Jahr)

SSL Aktivierung

- Nach **Ablauf der Gültigkeit** muss das **Zertifikat erneuert oder** ein **neues Zertifikat** ausgestellt werden
- **Android** erlaubt nur den **Import von Root CA Zertifikaten** in den vertrauenswürdigen Zeritifkatsspeicher
- Daher wird ein zusätzliches **Root CA Zertifikat** ausgestellt, mit dem dann das SSL Zertifikat der ownCloud signiert wird

SSL Aktivierung

- Privaten Schlüssel und Signaturanfrage (CSR) für den öffentlichen Schlüssel erstellen
  - **owncloud#** *openssl genrsa -out /etc/ssl/private/owncloud-key.pem 4096*
  - **owncloud#** *mkdir /etc/ssl/localcerts*
  - **owncloud#** *openssl req -new -key /etc/ssl/private/owncloud-key.pem -out /etc/ssl/localcerts/owncloud.csr -sha512*

  Hinweis
  Beim Erstellen des CSR für den öffentlichen Schlüssel werden weitere Parameter, wie Organisation und Hostname, abgefragt. Standardmäßig kann unter “Common Name (e.g. server FQDN or YOUR name) []:” nur ein Hostname eingetragen werden. Benötigt man mehrere Hostnamen (z. B. internen und externen Namen) muss man diese Namen OpenSSL in einer Konfigurationsdatei übergeben (siehe Weiterführende Informationen).

- Neues Terminalfenster öffnen und CSR auf den Desktoprechner kopieren
  - **local#** *scp root@owncloud:/etc/ssl/localcerts/owncloud.csr .*

- Root CA Zertifikat auf Desktoprechner erstellen (privaten Schlüssel, dann öffentlichen Schlüssel)
  - **local#** *openssl genrsa -aes256 -out rootca-key.pem 4096*
  - **local#** *openssl req -x509 -new -extensions v3_ca -key rootca-key.pem -days 1095 -out rootca-pub.pem -sha512*

SSL Aktivierung

- SSL-Zertifikat für die ownCloud mit dem privaten Schlüssel des Root Zertifikats signieren
  - **local#** *openssl x509 -req -in owncloud.csr -CA rootca-pub.pem -CAkey rootca-key.pem -CAcreateserial -out owncloud-pub.pem -days 180 -sha512*
- Signierten öffentlichen Schlüssel des SSL-Zertifikats auf ownCloud kopieren (vom Desktoprechener ausführen)
  - **local#** *scp ./owncloud-pub.pem root@owncloud:/etc/ssl/localcerts/owncloud-pub.pem*
- Zugriffsberechtigungen auf das Zertifikat einschränken
  - **owncloud#** *chmod 600 /etc/ssl/private/owncloud-key.pem*
  - **owncloud#** *chmod 600 /etc/ssl/localcerts/owncloud-pub.pem*
- SSL-Modul einschalten
  - **owncloud#** *a2enmod ssl*

SSL Aktivierung

- Apache Konfiguration erstellen (Beispielkonfigurationsdatei kopieren)
  - **owncloud#** *cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/owncloud.conf*
  - Konfigurationsdatei öffnen und Pfade der Schlüssel anpassen
    - **owncloud#** *vi /etc/apache2/sites-available/owncloud.conf*

      <VirtualHost *:443>

      SSLCertificateFile /etc/ssl/localcerts/owncloud-pub.pem

      SSLCertificateKeyFile /etc/ssl/private/owncloud-key.pem
  - Konfiguration aktivieren
    - **owncloud#** *a2ensite owncloud.conf*
- Webserver neustarten
  - **owncloud#** *service apache2 restart*
- Root Zertifikat (rootca-pub.pem) in Firefox/Thunderbird/Android/etc. importieren

Applikationen zur Synchronisation

- **Thunderbird**
  - Kalender (CalDAV): Lightning
  - Aufgaben (CalDAV): Lightning
  - Adressbuch (CardDAV): SOGo Connector
- **Android**
  - Kalender (CalDAV): DAVdroid
  - Aufgaben (CalDAV): DAVdroid und Mirakel
  - Adressbuch (CardDAV): DAVdroid

Applikationen zur Synchronisation

- **Lighting** kann bei fast allen Linuxdistributionen über den **Paketmanager** installiert werden
- Die **Android Apps** sind z. B. im **AppStore F-Droid** verfügbar und werden dort auch mit Updates versorgt
- Die jeweilgen **CalDAV und CardDAV URLs** sind in der **ownCloud Weboberfläche** unter der jeweiligen Funktion (App) zu finden

Demo

Externe Erreichbarkeit

- **Ziel: Erreichbarkeit der ownCloud über das Internet mittels eines festen Namens**
- **Voraussetzungen**
  - DNS Name im öffentlichen (Public) DNS
  - Router mit Dynamic DNS
  - Router mit Portforwarding

Externe Erreichbarkeit

- Der **DNS Name** löst die öffentliche IP-Adresse des Routers auf
- Bei **Dynamic DNS** teilt der Router dem DNS Provider regelmäßig seine öffentliche IP-Adresse mit (z. B. das kostenlose MyFritz bei FRITZ!Boxen)
- Empfehlung: ownCloud auf **Highport** (z. B. 54676) laufen lassen (keine Angriffe am Standardport 443)

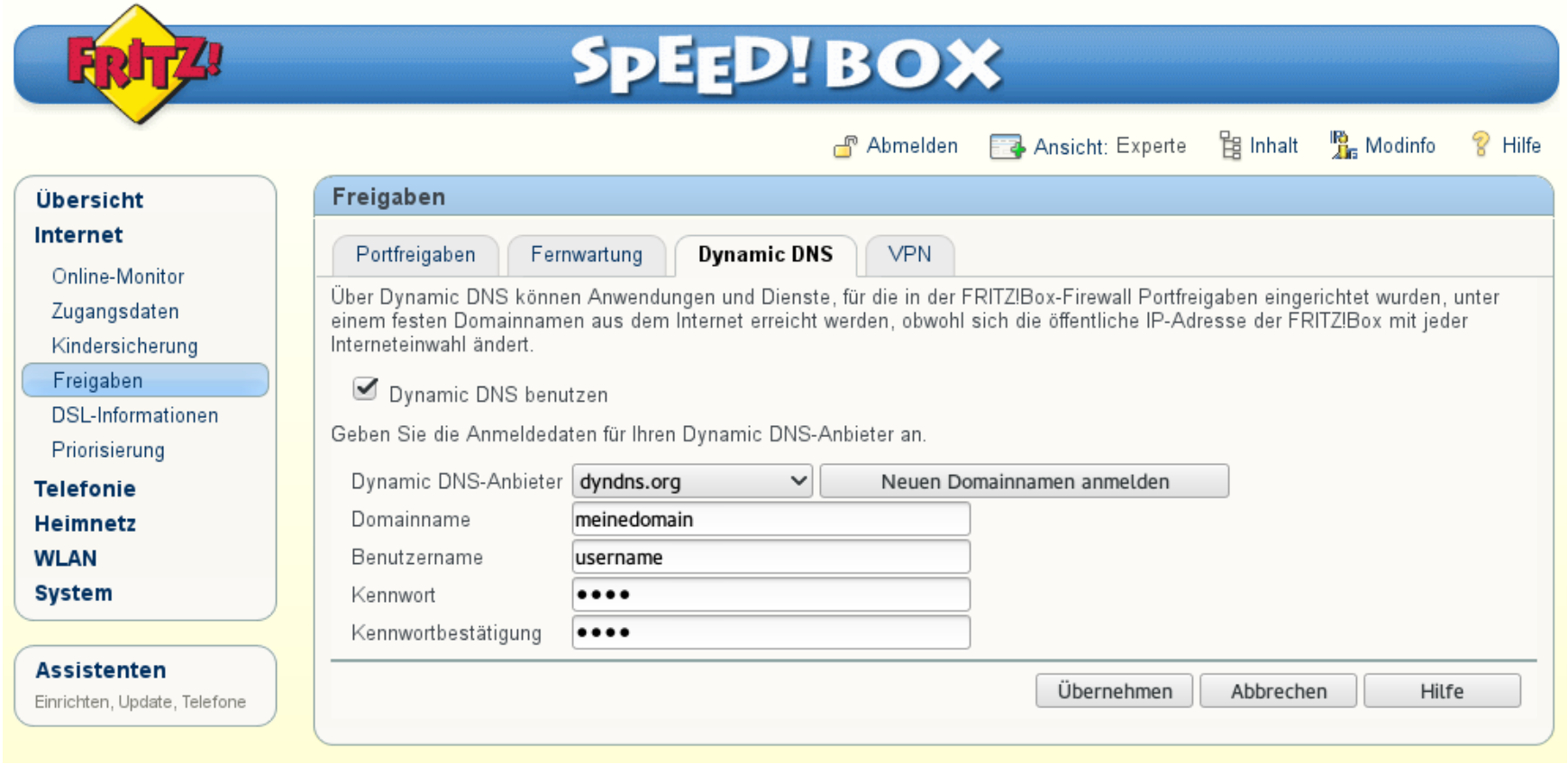
Externe Erreichbarkeit

- DNS Adresse für Dynamic DNS besorgen
- Dynamic DNS im Router konfigurieren
- Portforwarding im Router konfigurieren
- Konfiguration der ownCloud
  - config.php öffnen und im Array trusted_domains den externen DNS Namen hinzufügen
    - **owncloud#** *vi /usr/share/owncloud/config/config.php*

```
'trusted_domains' =>
array (
  0 => '<IP Adresse>',
  1 => 'owncloud',
  2 => '<dyndnsname.dyndns.org>',
  ),
```

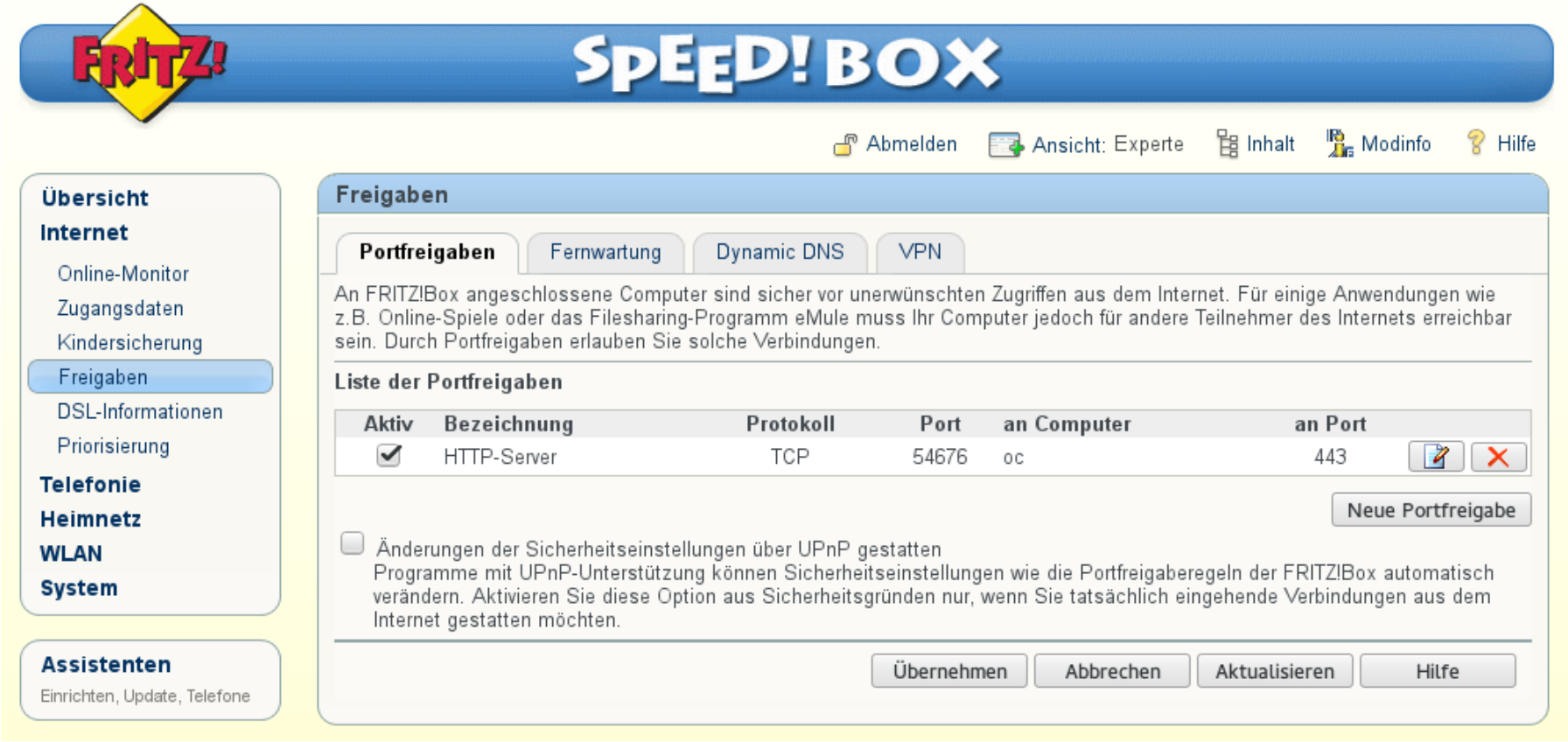
Externe Erreichbarkeit

- Beispiel: FRITZ!Box
  - Dynamic DNS



Externe Erreichbarkeit

- Beispiel: FRITZ!Box
  - Portforwarding



Automatisches Update

- **Sinnvoll zur**
  - **Reduzierung** des **Administrationsaufwands**
  - **Erhöhung der Sicherheit** (Security Updates werden automatisch installiert, auch wenn man z. B. nicht zu Hause ist)
- Mittels **Unattended Upgrades** möglich
- Information: **Unattended Upgrade** macht ein ***apt-get update***, kein *apt-get dist-upgrade* (muss also hin und wieder manuell durchgeführt werden)

Automatisches Update

- Paket unattended-upgrades installieren
  - **owncloud#** *apt-get install unattended-upgrades*
- Konfigurationsdatei öffenen und Nachstehendes einfügen, um das automatische Update für alle Repositories zu aktivieren
  - **owncloud#** *vi /etc/apt/apt.conf.d/50unattended-upgrades*

```
Unattended-Upgrade::Origins-Pattern {
        "o=Debian,a=stable";
        "o=Debian,a=stable-updates";
        "o=Debian,a=proposed-updates";
        "origin=Debian,codename=${distro_codename},label=Debian-Security";
        "o=Collabora,c=rpi2";
};
```

- Automatische Update einschalten
  - **owncloud#** *dpkg-reconfigure unattended-upgrades*
- Test durchführen
  - **owncloud#** *unattended-upgrades --dry-run*

Backup

- **Zu sichernde Daten**
  - Konfiguration und Dateiordner
  - Datenbank
- **Empfehlung**
  - Ein lokales Backup auf der ownCloud selbst
  - Zusätzliches Backup auf einen Netzwerkspeicher
  - Erstellung des Backups automatisiert und regelmäßig (am besten täglich) mittels Cronjob
  - Komplette Sicherung der SD Karte nach großen Konfigurations-änderungen

Backup

- **Bei lokalen Backups beachten**
  - Quotas anlegen, damit für das Backup ausreichend Speicherplatz zur Verfügung steht
  - Lokale Backups regelmäßig löschen

Backup

- Backupordner anlegen
  - **owncloud#** *mkdir /backup*
- Konfiguration und Dateiordner lokal sichern
  - **owncloud#** *rsync -Aax /var/lib/owncloud/ /backup/owncloud-dirbkp_`date +"%Y%m%d"`/*
- MySQL Datenbank lokal sichern
  - **owncloud#** *mysqldump --lock-tables -h [server] -u [username] -p[password] [db_name] > /backup/owncloud-sqlbkp_`date +"%Y%m%d"`.bak*
- Gesamte SD Karte sichern (zuvor unmounten)
  - **local#** *umount /dev/mmcblkXp1; umount /dev/mmcblkXp2*
  - **local#** *dd if=/dev/mmcblkX | gzip > /path/to/image.gz*
- Gesamte SD Karte zurückschreiben (Wiederherstellung)
  - **local#** *gzip -dc /path/to/image.gz | dd of=/dev/mmcblkX*

Hardening

- **Server im Internet** sind generell **angreifbar!**
- Dringende **Empfehlung: keine anderen Dienste** (z. B. E-Mailserver) auf der ownCloud betreiben
- Außerdem
  - **Default Website deaktivieren**
  - **Logging** und **Fail2ban** aktivieren (letzteres sperrt IP-Adressen nach einer gewissen Anzahl fehlgeschlagener Anmeldeversuche)
  - **Nicht benötigte Funktionen** in der Weboberfläche **deaktivieren**
    - Server-to-Server Sharing?
    - Öffentliche Freigaben?

Hardening

- **ownCloud Hardening and Security Guide**
  - Installation SELinux (dringend zu empfehlen)
  - Deaktivierung der Bildvorschau
  - Erzwingung von HTTPS
  - Richtige SSL Konfiguration (SSLCipherSuite)
  - u. a.

Hardening

- Default Website deaktivieren
  - Konfiguration der ownCloud Webseite öffnen und DocumentRoot der beiden VirtualHosts (Port 80 und 443) auf den Ordner /usr/share/owncloud festlegen
    - **owncloud#** *vi /etc/apache2/sites-available/owncloud.conf*
    - Beispielkonfiguration siehe nächste Seite
  - Defaultkonfiguration deaktivieren
    - **owncloud#** *a2dissite 000-default.conf*
  - Prüfen, ob nur owncloud.conf aktiv ist
    - **owncloud#** *ls -al /etc/apache2/sites-enabled/*
  - Webserver neuladen
    - **owncloud#** *service apache2 reload*
  - Ggf. Defaultwebsite entfernen (befindet sich im www Ordner)

Hardening

- Beispiel /etc/apache2/sites-available/owncloud.conf

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  DocumentRoot /usr/share/owncloud
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<IfModule mod_ssl.c>
  <VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /usr/share/owncloud
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile    /etc/ssl/localcerts/owncloud-pub.pem
    SSLCertificateKeyFile /etc/ssl/private/owncloud-key.pem
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>
    BrowserMatch "MSIE [2-6]" \
      nokeepalive ssl-unclean-shutdown \
      downgrade-1.0 force-response-1.0
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
  </VirtualHost>
</IfModule>
```

Hardening

- Fail2ban einschalten
  - Logging der ownCloud aktivieren; Konfigurationsdatei öffnen und nachstehende Zeilen einfügen
    - **owncloud#** *vi /etc/owncloud/config.php*

      'logtimezone' => 'Europe/Berlin',

      'logfile' => '/var/log/owncloud.log',

      'loglevel' => '2',
  - Webserver neustarten
    - **owncloud#** *systemctl restart apache2*
  - Logfile überprüfen
    - **owncloud#** *vi /var/log/owncloud.log*
  - Fail2ban installieren
    - **owncloud#** *apt-get install fail2ban*
  - Filter erstellen
    - **owncloud#** *touch /etc/fail2ban/filter.d/owncloud.conf*

Hardening

- Fail2ban einschalten (Fortsetzung)
  - **owncloud#** *vi /etc/fail2ban/filter.d/owncloud.conf*

    [Definition]

    failregex={"app":"core","message":"Login failed: '.*' \(Remote IP: '<HOST>', X-Forwarded-For: '.*'\)","level":2,"time":".*"}
  - Servicedefinition erstellen
    - **owncloud#** *touch /etc/fail2ban/jail.d/owncloud.local*
    - **owncloud#** *vi /etc/fail2ban/jail.d/owncloud.local*

      [owncloud]

      enabled = true

      filter = owncloud

      port   = https

      logpath = /var/log/owncloud.log

Hardening

- Fail2ban einschalten (Fortsetzung)
  - Fail2ban neustarten
    - **owncloud#** *systemctl restart fail2ban*
  - Test durchführen (falsche Anmeldedaten auf der Website angeben)
    - **owncloud#** *tail /var/log/owncloud.log*

Hinweis
Aktuell kann keines der Loglevel falsche Passworteingaben für geteilte Inhalte (external Sharing) loggen!

Hardening

- Weitere Einstellungen aus dem offiziellen Hardening and Security Guide sind fortgeschrittenes Level (wie z. B. Proper SSL configuration und SELinux). Es ist daher sehr zu empfehlen, sich dort selbst weiter einzuarbeiten. Die URLs zu den Guides sind in den Quellen am Ende aufgelistet.

Quellen

- Präsentationsvorlage
  https://github.com/owncloud/promo/tree/master/Presentation%20materials
- Sjoerd's Debian Image
  http://sjoerd.luon.net/posts/2015/02/debian-jessie-on-rpi2/
- Hostname ändern
  https://wiki.debian.org/HowTo/ChangeHostname
- SSL Konfiguration
  https://wiki.debian.org/Self-Signed_Certificate
- Automatisches Update
  https://wiki.debian.org/UnattendedUpgrades
- Backup
  https://doc.owncloud.org/server/7.0/admin_manual/maintenance/backup.html
  https://doc.owncloud.org/server/7.0/admin_manual/maintenance/restore.html

Quellen

- Secure Owncloud Server - Prevent brute-force password hacks
  http://www.rojtberg.net/711/secure-owncloud-server/
- Hardening and Security Guidance - ownCloud Administrators Manual
  https://doc.owncloud.org/server/7.0/admin_manual/configuration/harden_server.html
- Hardening and Security Guidance - ownCloud Enterprise Edition Administrators Manual
  https://doc.owncloud.com/server/7.0EE/admin_manual/configuration_server/harden_server.html

Weiterführende Informationen

- Asymmetrisches Kryptosystem
  https://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem
- SSL/TLS Strong Encryption: An Introduction
  http://httpd.apache.org/docs/2.4/ssl/ssl_intro.html
- Multiple Names on One Certificate
  http://apetec.com/support/GenerateSAN-CSR.htm
- F-Droid
  https://f-droid.org/
- Cron im Debianwiki
  https://wiki.debian.org/CronAnacronAtBatchSchedulers

Weiterführende Informationen

- Notes on PHP and security - Debian Wiki
  https://wiki.debian.org/PHP/#Notes_on_PHP_and_security
- Mozilla SSL Configuration Generator
  https://mozilla.github.io/server-side-tls/ssl-config-generator/
- SELinux im Debianwiki
  https://wiki.debian.org/SELinux
- Proxy für Owncloud soll Heimnutzung erleichtern
  http://www.golem.de/news/private-cloud-proxy-fuer-owncloud-soll-heimnutzung-erleichtern-1508-116014.html

Vielen Dank für die Aufmerksamkeit!