

Fritz.Box hinter dem SP-H
optimiertes Setup
ohne SIP-PROXY und mit IPv6 Prefix
Delegation
und vieles mehr.

Achtung, funktioniert nur mit Firmware <= V2 ..012.

**ANWENDUNG AUF EIGENE GEFAHR!
Ich übernehme keine Verantwortung falls etwas schief geht!**

danXde

Vorab möchte ich noch anmerken:

- danke an ***stricted*** für das Einrichten des Forum
- danke an ***eMKay77, stricted, hefr54*** für die vielen Tool's und Fortschritte für das Tunning des SP-H
- danke für die weiteren Tipps und Tricks und Feedback bei allen, die sonst dazu beigetragen haben ...sorry sind zu viele, dass ich diese nicht namentlich erwähne.
- Ebenfalls danke an ***Mithcair*** für die Vorarbeit des Dokuments „DSL+LTE Tunnel brechen“, welches ich als Grundlage für die Doku genutzt habe

1 Einführung

Ihr solltet die Anleitung erst mit Sorgfalt durchlesen (ja, bis zum Schluss), da ihr sonst schwerwiegende Fehler verursachen könnt. Außerdem solltet ihr euch über jede Sache (Befehle/Quelltext) informieren der hier geschrieben wird, man weiß halt dann wirklich was man macht und lernt was dabei.

Ich verlinke hier außerdem nochmal alle Foren-Threads die ich dazu genutzt habe um diese Anleitung zu schreiben. Die am besten ebenfalls durchlesen. Alle nicht erwähnten Themen findet ihr ebenfalls im Forum. Einfach mal die Suche benutzen.

Allgemeine Threads über die jeweiligen Schritte:

1. [tools zum ent-/verschlüsseln der config](#)
2. [Telnet aktivieren](#)
3. [Schreibzugriff auf rootfs](#)
4. [SSH dropbear einrichten](#)
5. [Bootstrap - Erweiterungen / Änderungen auf USB auslagern](#)
6. [Spielplatz-Extrem -- oder BusyBox-MIPS in der Vollversion und offiziell](#)

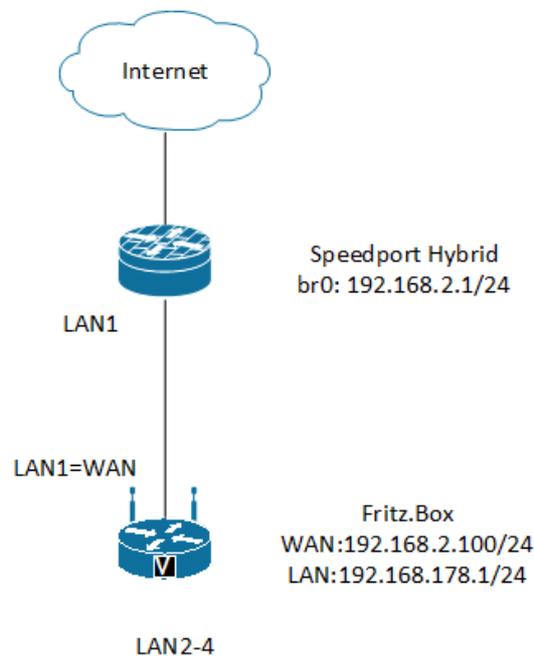
Der Umfang des erklärten Setups ist so, das der SP-H nur

- DSL
- LTE
- Hybrid
- DynDNS

erledigt. Sämtliche anderen Funktionen werden in der Fritz.Box realisiert. Ich setze die AVM 7490 ein, diese macht:

- VoIP Telekom
- VoIP Sipgate
- DECT für Fritz!Phone
- SIP-Server für Fritz!Fon (Softclient - APP)
- DHCP für die Clients
- WLAN
- LAN
- Kindersicherung
- VPN Client-Access für die Handy's von Aussen
- Optional HTTPS-Zugriff auf die Fritz.Box von Aussen

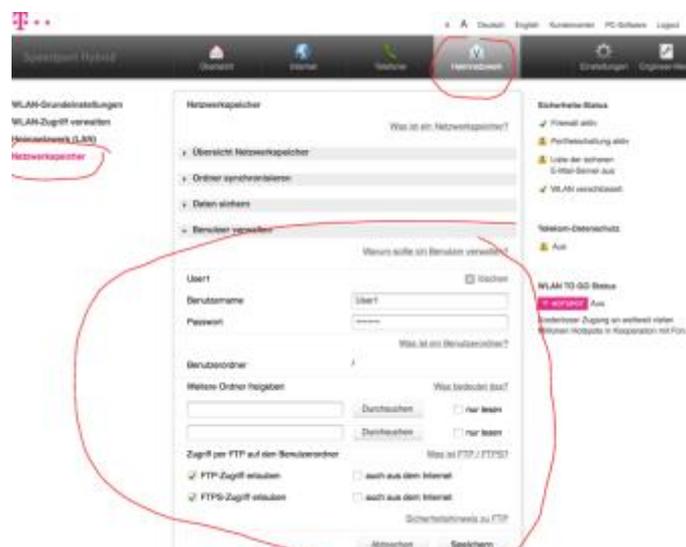
2 Netzwerkübersicht



3 FTP vorbereiten

Vorab bereiten wir noch den FTP-Access vor:

1. Einen USB-Stick in den SP-H stecken
2. Dann einen User im Menü unter Heimnetzwerk nach Wahl anlegen und den FTP- und FTPS-Zugriff aus dem internen Netz erlauben:

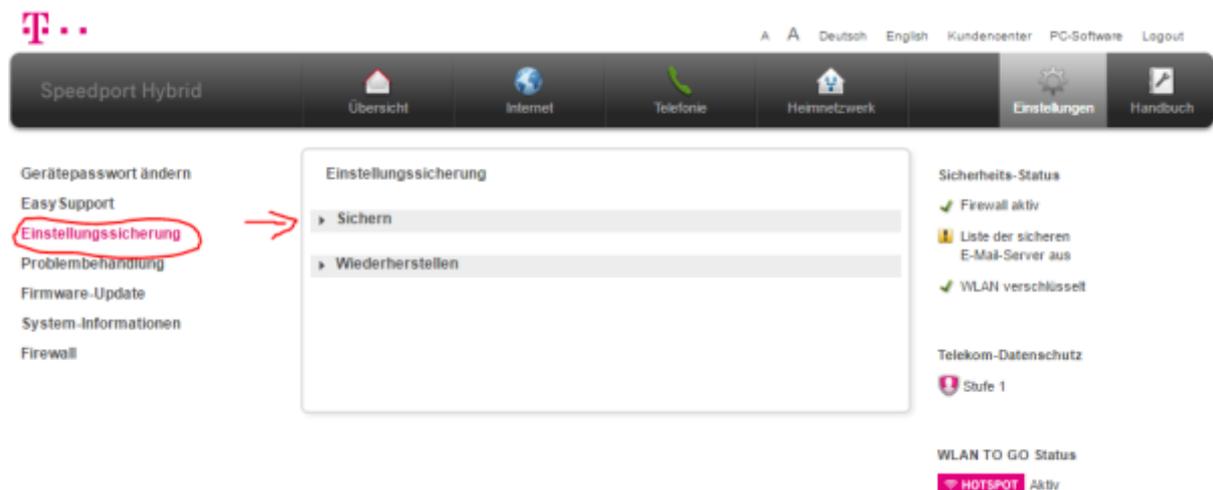
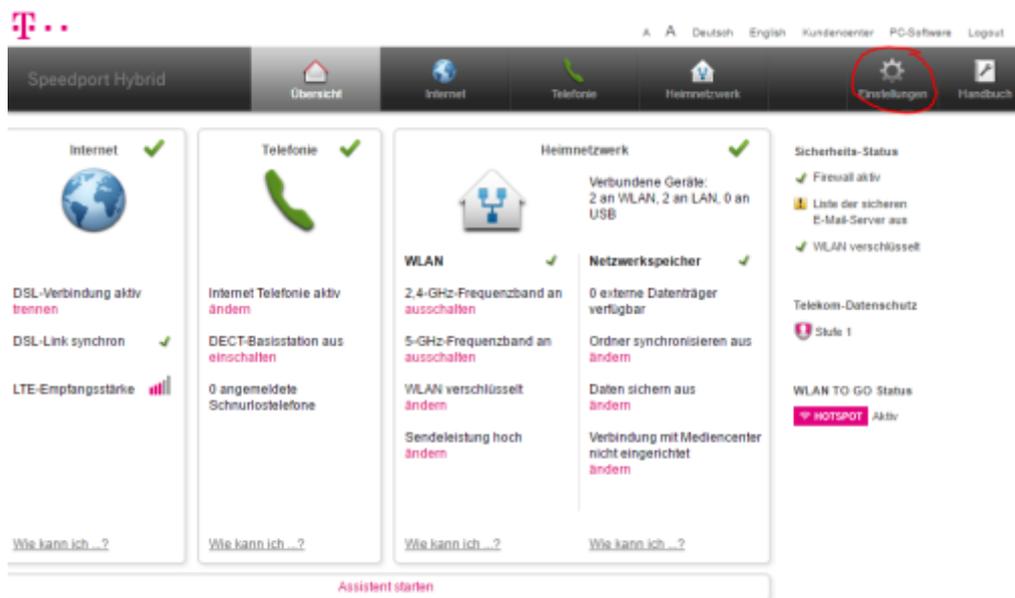


4 Telnet aktivieren

4.1 Wiederherstellungssicherung auf dem PC sichern

Die Einstellungssicherung des Speedport Hybrid (SP-H) runterladen unter Einstellungen (Oben rechts in der WebUI (Erreichbar unter 192.168.2.1). Diese dient zum einen als Sicherheit, zum anderen als Ausgangsdatei für die weiteren Schritte.

Daher solltet Ihr Euch eine weitere Kopie zur Sicherheit nochmal in einen anderen Ordner ablegen.



Einstellungssicherung

▼ Sichern

Sie können die Einstellungen des Routers auf einem Datenträger (z.B. Festplatte, USB-Stick) sichern um den Router jederzeit wieder auf den Stand zum Zeitpunkt der Sicherung zu bringen.

Klicken Sie dazu auf „Sichern“ und wählen Sie einen Ort aus, an dem Sie die Einstellungen sichern wollen.

Sichern

4.2 Wiederherstellungssicherung entschlüsseln

4.2.1 Windows

1. Das Programm SPHDecode runterladen unter

<https://github.com/Stricted/SPHDecode/releases>

2. Das Programm auspacken und in einen Ordner speichern

3. Dann das Programm SPHDecode.exe öffnen und

- als „Source File“ die heruntergeladene Wiederherstellungssicherung auswählen,
- dann mit „Save File“ auswählen wo die entschlüsselte Datei gespeichert werden soll.
- Dann klickt ihr auf „decrypt“.
Nun befindet sich die Datei im vorher angegebenen Speicherort.

4. Diese müsst ihr nun entweder mit Notepad++ oder ganz einfach WordPad öffnen. Dort ändert ihr den Eintrag:

```
<X_ServiceManage TelnetEnable="0" TelnetPort="23" KeyEquipMode="0"/>
```

Um in:

```
<X_ServiceManage TelnetEnable="1" TelnetPort="23" KeyEquipMode="0"/>
```

Damit aktiviert ihr den Telnet Service vom Router. Dann müsst ihr noch den **User für Telnet** bearbeiten:

```
<X_Cli>
<UserInfo NumberOfInstances="2">
<UserInfoInstance InstanceID="1" Username="!!Huawei" Userpassword="@HuaweiHgw"
Userlevel="0" Timestamp="0000-00-00 00:00:00" Size="0"/>
<UserInfoInstance InstanceID="2" Username="user" Userpassword="user" Userlevel="0"
Timestamp="0000-00-00 00:00:00" Size="0"/>
</UserInfo>
</X_Cli>
```

Um in:

```
<X_Cli>
<UserInfo NumberOfInstances="2">
<UserInfoInstance InstanceID="1" Username="mein_USERNAME" Userpassword="mein_PASSWORT"
Userlevel="0" Timestamp="0000-00-00 00:00:00" Size="0"/>
<UserInfoInstance InstanceID="2" Username="user" Userpassword="user" Userlevel="0"
Timestamp="0000-00-00 00:00:00" Size="0"/>
</UserInfo>
</X_Cli>
```

Den Username und das Userpassword müsst ihr euch selbst was einfallen lassen. Ändern müsst ihr nur die erste InstanceID. Mit den selbst angegebenen Username und Userpassword meldet ihr euch später via z.B. putty auf euren Router an.

Nun noch der Nebenkriegsschauplatz FTP. Der Benutzer „User1“ wurde bereits in Kapitel 3 angelegt, für diesen wird den Pfad von **/mnt** in **/** geändert:

```
<UserAccount NumberOfInstances="2">
<ObjExtention MaxInstanceNum="2"/>
<UserAccountInstance InstanceID="1" Enable="0" Username="GUEST" Password="9876543AB."
AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0" X_AllowFTPInternetAccess="0"
X_AllowFTPSInternetAccess="0" X_AllPath="0" X_Permissions="1" X_UserFolder=""/>
<UserAccountInstance InstanceID="2" Enable="1" Username="USER1" Password="mein_PASSWORT"
AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0" X_AllowFTPInternetAccess="0"
X_AllowFTPSInternetAccess="0" X_AllPath="0" X_Permissions="1" X_UserFolder="/mnt"/>
</UserAccount>
```

ändern in:

```
<UserAccount NumberOfInstances="2">
<ObjExtention MaxInstanceNum="2"/>
<UserAccountInstance InstanceID="1" Enable="0" Username="GUEST" Password="9876543AB."
AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0" X_AllowFTPInternetAccess="0"
X_AllowFTPSInternetAccess="0" X_AllPath="0" X_Permissions="1" X_UserFolder=""/>
<UserAccountInstance InstanceID="2" Enable="1" Username="USER1" Password="mein_PASSWORT"
AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0" X_AllowFTPInternetAccess="0"
X_AllowFTPSInternetAccess="0" X_AllPath="0" X_Permissions="1" X_UserFolder=""/>
</UserAccount>
```

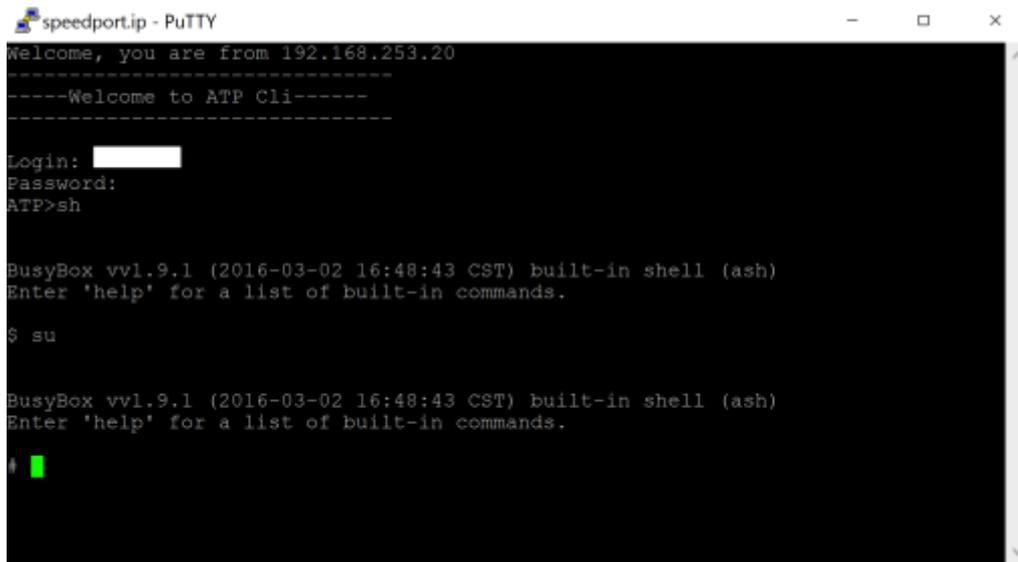
5. Die Datei speichern (bei Notepad++ darauf achten, das die Datei im Linux-Format gespeichert wird) bei Wordpad sollte das so passen, wenn Ihr es im TXT abspeichert.-.
6. Nun geht ihr wieder in das SPHDecode Programm und macht nun das Gleiche wie bei Schritt 3. Und klickt ihr statt „decrypt“ auf „encrypt“.
7. Die nun gespeicherte Datei ladet ihr wieder auf euren Router mit dem Reiter „Wiederherstellen“ in den Einstellungen.

Hinweis: Wenn sich die Datei nicht einlesen lässt, könnte das an den falschen Text-Umbrüchen liegen.



Jetzt sollte der Router neustarten und damit die Config übernehmen.

8. Nun könnt ihr euch mit „putty“ auf Telnet anmelden (192.168.2.1). Dort gebt ihr dann die Anmeldedaten ein die ihr vorher in der Config geändert habt.



```
speedport.ip - PuTTY
Welcome, you are from 192.168.253.20
-----Welcome to ATP Cli-----
Login: 
Password: 
ATP>sh

BusyBox v1.9.1 (2016-03-02 16:48:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

$ su

BusyBox v1.9.1 (2016-03-02 16:48:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
```

Um nun in der Shell vom Router zu landen, gebt ihr einfach „sh“ in die Konsole ein, damit ihr Änderungen durchführen könnt müsst ihr euch noch zu „root“ machen, indem ihr per „su“ zu diesem Userkontext wechselt.

4.2.2 Linux (mit Ubuntu-Installation 16.04 durchgeführt)

1. Das Skript runterladen unter

<http://pastebin.com/SRvMGk1n>

2. Vereinfachung für's leichtere Handling:

```
ln -s speedport_hybrid_configtool.py.py sph.py
```

3. Ausführbar machen

```
chmod 777 sph.py
```

4. Entschlüsseln

```
sph.py decrypt Speedport_Hybrid_050124.02.00.012_xx.xx.xxxx_xxxxxx.config SPH-DECRYPT.TXT
```

5. Jetzt in einem Editor Eurer Wahl öffnen und editieren:

```
<X_ServiceManage TelnetEnable="0" TelnetPort="23" KeyEquipMode="0"/>
```

Um in:

```
<X_ServiceManage TelnetEnable="1" TelnetPort="23" KeyEquipMode="0"/>
```

Damit aktiviert ihr den Telnet Service vom Router. Dann müsst ihr noch den User für Telnet bearbeiten:

```
<X_Cli>
<UserInfo NumberOfInstances="2">
<UserInfoInstance InstanceID="1" Username="!!Huawei" Userpassword="@HuaweiHgw"
Userlevel="0" Timestamp="0000-00-00 00:00:00" Size="0"/>
<UserInfoInstance InstanceID="2" Username="user" Userpassword="user"
Userlevel="0" Timestamp="0000-00-00 00:00:00" Size="0"/>
</UserInfo>
</X_Cli>
```

Um in:

```
<X_Cli>
<UserInfo NumberOfInstances="2">
<UserInfoInstance InstanceID="1" Username="mein_USERNAME"
Userpassword="mein_PASSWORT" Userlevel="0" Timestamp="0000-00-00 00:00:00"
Size="0"/>
<UserInfoInstance InstanceID="2" Username="user" Userpassword="user"
Userlevel="0" Timestamp="0000-00-00 00:00:00" Size="0"/>
</UserInfo>
</X_Cli>
```

Den Username und das Userpassword müsst ihr euch selbst was einfallen lassen. Ändern müsst ihr nur die erste InstanceID. Mit den selbst angegebenen Username und Userpassword meldet ihr euch später via z.B. putty auf euren Router an.

Nun noch der Nebenkriegsschauplatz FTP. Der Benutzer „User1“ wurde bereits in Kapitel 3 angelegt, für diesen wird den Pfad von /mnt in / geändert:

```
<UserAccount NumberOfInstances="2">
<ObjExtention MaxInstanceNum="2"/>
<UserAccountInstance InstanceID="1" Enable="0" Username="GUEST" Password="9876543AB."
AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0"
X_AllowFTPInternetAccess="0" X_AllowFTPSInternetAccess="0" X_AllPath="0"
X_Permissions="1" X_UserFolder=""/>
<UserAccountInstance InstanceID="2" Enable="1" Username="USER1"
Password="mein_PASSWORT" AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0"
X_AllowFTPInternetAccess="0" X_AllowFTPSInternetAccess="0" X_AllPath="0"
X_Permissions="1" X_UserFolder="/mnt"/>
</UserAccount>
```

Ändern in:

```
<UserAccount NumberOfInstances="2">
<ObjExtention MaxInstanceNum="2"/>
<UserAccountInstance InstanceID="1" Enable="0" Username="GUEST" Password="9876543AB."
AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0"
X_AllowFTPInternetAccess="0" X_AllowFTPSInternetAccess="0" X_AllPath="0"
X_Permissions="1" X_UserFolder=""/>
<UserAccountInstance InstanceID="2" Enable="1" Username="USER1"
Password="mein_PASSWORT" AllowFTPAccess="1" X_AllowFTPSAccess="1" AllowHTTPAccess="0"
X_AllowFTPInternetAccess="0" X_AllowFTPSInternetAccess="0" X_AllPath="0"
X_Permissions="1" X_UserFolder="/" />
</UserAccount>
```

6. Die Datei speichern

7. Verschlüsseln

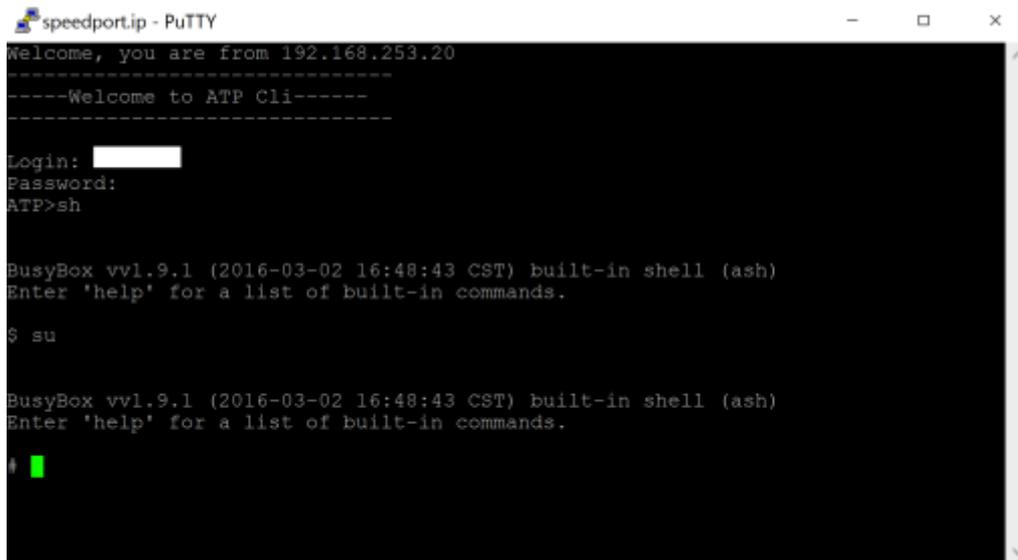
sph.py encrypt SPH-DECRYPT.TXT Speedport_Hybrid_050124.02.00.012.new.config

8. Die nun gespeicherte Datei ladet ihr wieder auf euren Router mit dem Reiter „Wiederherstellen“ in den Einstellungen.



Jetzt sollte der Router neustarten und damit die Config übernehmen.

9. Nun könnt ihr euch mit „putty“ auf Telnet anmelden (192.168.2.1). Dort gebt ihr dann die Anmeldedaten ein die ihr vorher in der Config geändert habt.



```
speedport.ip - PuTTY
Welcome, you are from 192.168.253.20
-----Welcome to ATP Cli-----
Login: 
Password:
ATP>sh

BusyBox v1.9.1 (2016-03-02 16:48:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

$ su

BusyBox v1.9.1 (2016-03-02 16:48:43 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
```

Um nun in der Shell vom Router zu landen, gebt ihr einfach „sh“ in die Konsole ein, damit ihr Änderungen durchführen könnt müsst ihr euch noch zu „root“ machen, indem ihr per „su“ zu diesem Userkontext wechselt.

5 USB-Stick vorbereiten

Die Größe des USB-Stick ist Euch überlassen, das Minimum ist 512MB.

5.1 Formatieren des Sticks

5.1.1 Windows

Nutzt dazu bitte das Tool und die Erklärung von hier:

<https://code-bude.net/2013/02/12/ext4-partition-unter-windows-erstellen/>

Bitte legt eine EXT3-Partition auf dem Stick an. Im Bild ist es die untere Festplatte, achtet bitte drauf, das Ihr nicht ausversehen die Windows oder eine andere Festplatte mit Daten erwischt!!!



5.1.2 Linux

1. Terminal starten
2. `fdisk` starten
3. `sudo fdisk /dev/<Stickbezeichnung z.B. sda1>`
4. in `fdisk` `d` eingeben, um existierende Partitionen zu löschen, ggf. `c` eingeben, um den DOS-compatible mode auszuschalten und `u`, um display/entry units zu Sektoren (sectors) zu ändern.
5. in `fdisk` `n` eingeben um eine neue Partition zu erstellen.
6. in `fdisk` `p`, um die neue Partition zu einer primären (primary) Partition zu machen.
7. Partitionsnummer eingeben, z.B. `1` eingeben.
8. in `fdisk` `w` eingeben, um die Partitionstabelle auf das Speichermedium zu schreiben.
9. ENTER, um `fdisk` zu beenden.
10. USB-Drive mit dem Dateisystem `ext3` formatieren (Auch wieder im Terminal)
`sudo mkfs.ext3 -L usbdrive /dev/<Stickbezeichnung z.B. sda1>`

5.2 Archiv auf den Stick kopieren

Um nun die Tools auf dem USB-STICK auf dem Router zum Laufen zu bekommen, müssen wir erstmal ein paar Dinge vorbereiten. Dazu bitte den USB-Stick an den SP-H stecken. Per FTP die Archiv-Datei `bootstrap.tgz` und die ausgepackte Datei `busybox-mips` (im Binary-Mode) auf den im SP-H gesteckten Stick kopieren.

Dann auf dem Router per Telnet einloggen und in die Shell (`sh`) in den Usermodus „root“ (`su`) wechseln, dann folgende Schritte im SP-H auf der CLI ausführen:

```
cd /mnt/<Name_des_USBSTICK>/
cp busybox-mips /tmp/busybox-mips
chmod 777 /tmp/busybox-mips
/tmp/busybox-mips tar xvzf _bootstrap_.tgz
```

5.3 Bootstrap im SP-H konfigurieren

Um nun die Tools auf dem USB-STICK auf dem Router zum Laufen zu bekommen, müssen wir erstmal ein paar Dinge vorbereiten. Dazu bitte den USB-Stick an den SP-H stecken (für die Linuxer) und sich auf dem Router per Telnet einloggen und in die Shell (`sh`) in den Usermodus „root“ (`su`) wechseln

Hier habt Ihr 2 Möglichkeiten:

- Variante 1- Manipulation des originalen Filesystems
 - Zugriff funktioniert mit telnet und ssh,
 - allerdings kein Fallback, wenn der Stick abgezogen wird)
- Variante 2 - Remount des `/bin` Verzeichnisses
 - Zugriff funktioniert nach dem Boot nur per ssh
 - telnet funzt nicht mehr
 - fallback, wenn USB-Stick abgezogen, funktioniert er „wie original“

5.3.1 Automatische Installation Variante 1:

Achtung, wenn das ausgeführt wurde, dann könnt Ihr den SP-H nicht mehr ohne den Stick nutzen, wenn Ihr das allerdings nicht macht, dann können die Dienste wieder gestartet werden und bringen das Setup durcheinander (Z.B. wenn Telekom eine neue Filterliste schickt...).

```
cp /mnt/USB-STICK_NAME/_bootstrap_/bin/install_system.sh /tmp/.
cd /tmp
chmod 777 install_system.sh
install_usb.sh
```

Weiter in 5.4....

5.3.2 Automatische Installation Variante 2:

```
cp /mnt/USB-STICK_NAME/_bootstrap_/bin/install_usb.sh /tmp/.
cd /tmp
chmod 777 install_usb.sh
install_usb.sh
```

Weiter in 5.4....

5.3.3 Manuelle Installation Variante 1:

Achtung, wenn das ausgeführt wurde, dann könnt Ihr den SP-H nicht mehr ohne den Stick nutzen, wenn Ihr das allerdings nicht macht, dann können die Dienste wieder gestartet werden und bringen das Setup durcheinander (Z.B. wenn Telekom eine neue Filterliste schickt...).

```
# USB sauber einrichten
bsbase="$(ls -d /mnt/*/_bootstrap_ 2>/dev/null)"
usbname="$(echo $bsbase | cut -d '/' -f 3 2>/dev/null)"
usbdev="$(mount | grep $usbname | cut -d ' ' -f 1 2>/dev/null)"
mkdir /tmp/bsmnt
mount -o exec $usbdev /tmp/bsmnt

# rootfs vorbereiten
mkdir /tmp/rmnt
mount -t jffs2 /dev/mtdblock0 /tmp/rmnt
mount -o remount,rw /dev/mtdblock0 /tmp/rmnt
mkdir /tmp/rmnt/opt
mkdir /tmp/rmnt/opt/bin
mkdir /tmp/rmnt/usr/local

# bootstrap
cp /tmp/bsmnt/_bootstrap_/install/bootstrap /tmp/rmnt/opt/bin/.
chmod 777 /tmp/rmnt/opt/bin/bootstrap

# profile
mv /tmp/rmnt/etc/profile /tmp/rmnt/etc/profile.o
cp /tmp/bsmnt/_bootstrap_/install/profile /tmp/rmnt/etc/profile
chmod 770 /tmp/rmnt/etc/profile
```

Jetzt müssen wir noch einige Programme anpassen, damit diese in Zukunft nicht mehr gestartet werden können.

```
cd /tmp/rmnt/bin
mv voiper voiper.o
mv sipproxd sipproxd.o
mv dhcpd6s dhcpd6s.o
mv iptables iptabels.o
```

Weiter in 5.4....

5.3.4 Manuelle Installation Variante 2:

```
# USB sauber einrichten
bsbase="$(ls -d /mnt/*/_bootstrap_2>/dev/null)"
usbname="$(echo $bsbase | cut -d '/' -f 3 2>/dev/null)"
usbdev="$(mount | grep $usbname | cut -d ' ' -f 1 2>/dev/null)"
mkdir /tmp/bsmnt
mount -o exec $usbdev /tmp/bsmnt

# rootfs vorbereiten
mkdir /tmp/rmnt
mount -t jffs2 /dev/mtdblock0 /tmp/rmnt
mount -o remount,rw /dev/mtdblock0 /tmp/rmnt
mkdir /tmp/rmnt/opt
mkdir /tmp/rmnt/opt/bin
mkdir /tmp/rmnt/usr/local

# bootstrap
cp /tmp/bsmnt/_bootstrap_/install/bootstrap /tmp/rmnt/opt/bin/.
chmod 777 /tmp/rmnt/opt/bin/bootstrap

# profile
mv /tmp/rmnt/etc/profile /tmp/rmnt/etc/profile.o
cp /tmp/bsmnt/_bootstrap_/install/profile /tmp/rmnt/etc/profile
chmod 770 /tmp/rmnt/etc/profile

# Remount /bin vorbereiten
cd /tmp/bsmnt/_bootstrap_/orig
cp -R /bin ./bin

cd /tmp/bsmnt/_bootstrap_/orig/bin
mv voiper voiper.o
mv siproxd siproxd.o
mv dhcp6s dhcp6s.o
mv iptables iptabels.o

cd /tmp/bsmnt/_bootstrap_
rm -f bootstrap_init.sh
cp bootstrap_init_usb.sh bootstrap_init.sh
```

5.4 SSH konfigurieren (optional)

Ihr könnt erstmal den Benutzer „User1“ mit Passwort „YourPASS“ benutzen. Wenn Ihr wollt, könnt Ihr das auf Eure eigenen Bedürfnisse anpassen, dazu müsst noch die beiden Dateien anpassen und einen eigenen User anlegen.

```
_bootstrap_/etc/dropbear/passwd
_bootstrap_/etc/dropbear/shadow
```

Anmerkung: Mit dem Befehl

```
makepasswd --clearfrom=- --crypt-md5 <<< YourPASS
```

könnt Ihr Euer Passwort auf einem Linux-System verschlüsseln (`sudo apt install makepasswd` wenn noch nicht da) `YourPASS` durch Euer Passwort in den Anführungszeichen ersetzen.

6 Adresscheck und gegeben falls deren Anpassung (wenn man nicht 192.168.2.0/24 für den SP-H verwenden möchte)

Das Setup ist jetzt darauf ausgelegt. Das der SP-H die 192.168.2.1 hat. Die Fritz.box bekommt per DHCP die .100 zugewiesen.

Wenn Ihr mit festen IP arbeiten wollt, achtet bitte darauf, dass diese **außerhalb des DHCP-Bereiches sich befindet.**

Prüft in der SP-H Gui die Zuordnung des Gerätes.

Wenn Ihr Änderungen am Scope vornehmt, müssen auf jeden Fall in folgenden Dateien Adressen angepasst werden:

```
/opt/bootstrap_init.sh  
/opt/bin/fw.sh  
/opt/http/httpd.conf
```

Wenn nur die Fritz.Box eine andere IP hat, dann nur:

```
/opt/bin/fw.sh
```

In den Dateien dann nach 192.168.2. suchen und die entsprechenden IP's zu ändern.

7 Aufräumen und Neustarten

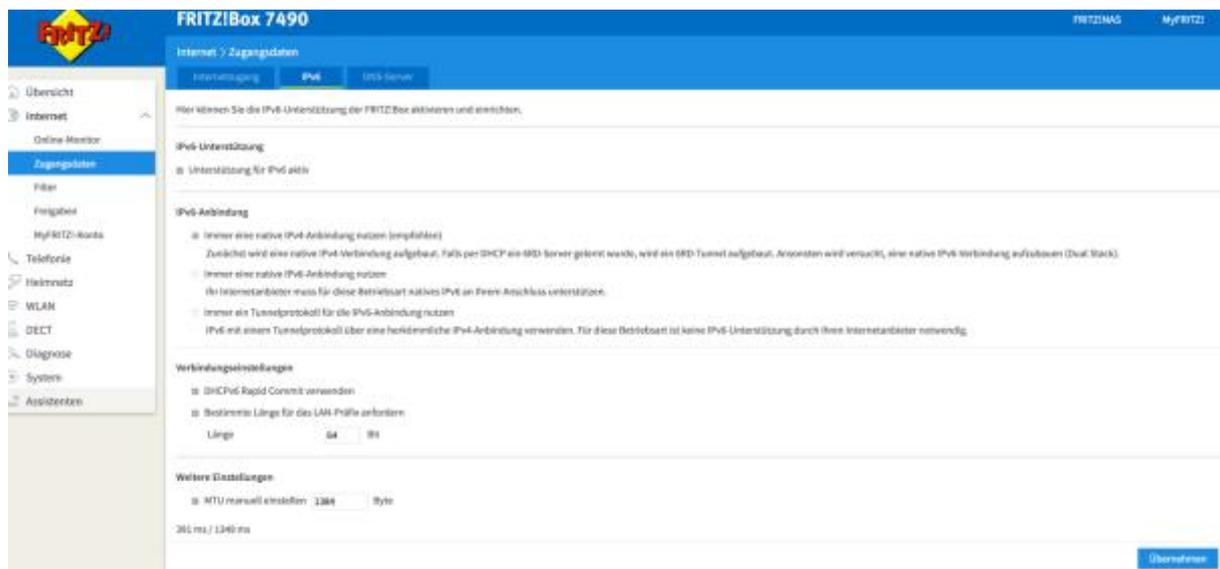
In der CLI (`telnet`, `sh`, `su`) eingeben:

```
cd /  
umount /tmp/rmnt  
reboot
```

8 Fazit:

Jetzt sollte der SP-H mit der neuen Konfiguration booten, ohne SIP-PROXY, mit IPv6 Prefix-Delegation, mit Telnet, FTP und SSH Zugang. Einer zweiten WebGUI auf Port 8080, dem LED-Daemon, etc.

Achtung in der Fritz.Box muss IPv6 nun ebenfalls aktiv sein, damit der Client hinter der Fritz.Box auch IPv6 nutzen kann:



9 Hinweise zu Funktionseinschränkung

Wenn Ihr das Setup erfolgreich konfiguriert habt, dann ist sind leider 2 Dinge nicht mehr in der GUI funktional:

1. Die LTE-Ausnahmen
2. Die Portweiterleitungen

Wenn Ihr dort weitere Änderungen benötigt, müssen diese im Firewall-Script konfiguriert werden:

```
/opt/bin/fw.sh
```

10 Weitere Funktionen:

Ruft dazu einfach mal im Browser die URL <http://speedport.ip:8080> auf.

- XDSLGRAPH (bitte im Forum funktionsweise nachlesen)
- LTEGRAPH (bitte im Forum funktionsweise nachlesen)
- Frequenzsetting per GUI (bitte im Forum funktionsweise nachlesen)
- LEAD-Demon (Wenn ein Tunnel down ist, blinkt die Power-LED rot)