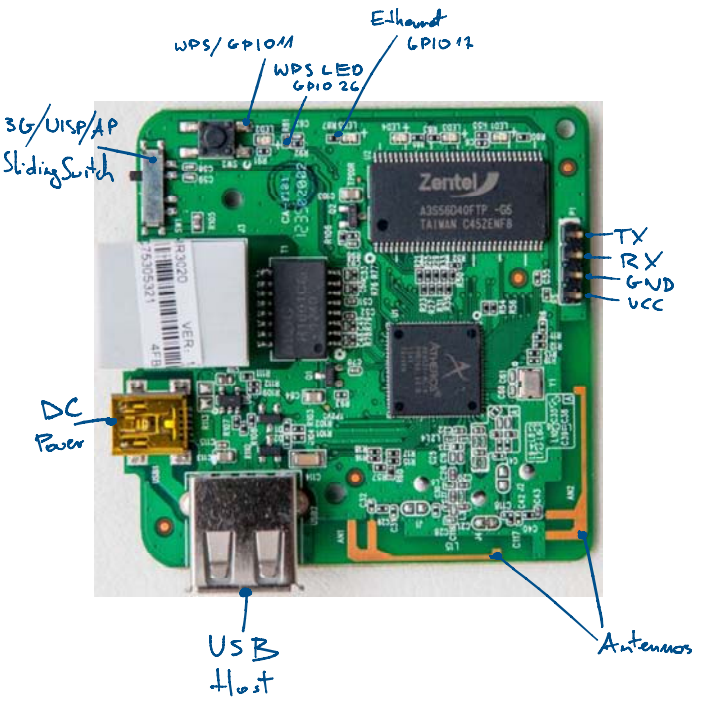
Installing OpenVPN on mini router TP-Link MR3020

Author: Karol Bonenberg (c) 2013



not finished

```
openwrt-ar71xx-generic-tl-mr3020-v1-squashfs-factory.bin
TPLINK TL-MR3020 HW V1.0
```

-1.

Po połączeniu kablem routera z komputerem, router nadaje nam za pomocą DHCP adres. Od tego momentu możemy zalogować się przez przeglądarke na panel administracyjny pod adresem 192.168.1.1.

Zmieniamy hasło administratora, konfigurujemy sieć. Przy interfejsie ustawiamy DHCP Client, wyłączamy serwer DHCP, wyłączamy tryb bridge między interfejsami.

0.

Zacznę od tego, że instalując pakiety zawiesiłem sobie router - błędy typu
“No space left on device” przy operacjach na katalogach
"Invalid argument" przy mount
"Segmentation fault" przy próbie instalowaina softu

żeby się ich pozbyć wyczściłem flasha i skonfigurowałem od początku:

```
rm -r /overlay/*
```

1.

Zaczynamy od usunięcia zbędnych pakietów (za mało miejsca w urządzeniu, instalowanie zablokuje sprzęt)

```
/etc/init.d/uhttpd stop
```

opkg remove --force-depends luci-core luci-uci luci-theme-openwrt luci-web luci-sgi-cgi luci-http luci-app-firewall luci-admin-core luci-sys luci-uvl luci-lmo luci-app-initmgr luci-ipkg luci-cbi luci-admin-mini luci-admin-full luci-ipkg luci-admin-core luci-cbi luci-uvl luci-i18n-english luci-uci uhttpd

2.

Instalujemy paczki do obsługi pamięci usb (za http://wiki.openwrt.org/doc/howto/usb.storage)

```
opkg install kmod-usb-storage
opkg install kmod-fs-ext4
opkg install block-mount
```

3.

Rebootujemy urządzenie

```
reboot
```

4.

Na innym systemie dzielimy pamięć usb na partycje - ok 64 mb typu swap, reszta ext4, formatujemy, i podłączamy do routera. Sprawdzamy czy działa:

```
dmesg | tail
```

5.

Włączamy swap na pamięci usb i montujemy resztę dysku. Polecenie mount jest dokładne, chociaż teoretycznie wystarczyłoby uproszczone (podać powód)

```
mkswap /dev/sda1
swapon /dev/sda1
mkdir -p /mnt/sda2
mount -t ext4 /dev/sda2 /mnt/sda2 -o rw,sync
```

6.

Przenosimy cały system na usb, inaczej external root (aka pivot root). Za https://samhobbs.co.uk/2013/11/more-space-for-packages-with-extroot-on-your-openwrt-router

```
mkdir -p /tmp/cproot
mount --bind / /tmp/cproot
tar -C /tmp/cproot -cvf - . | tar -C /mnt/sda2 -xf -
umount /tmp/cproot
```

7.

Konfigurujemy fstab

```
vim /etc/config/fstab
```

i dodajemy linie

```
config mount
        option target       /
        option device       /dev/sda2
        option fstype       ext4
        option options      rw,sync
        option enabled      1
        option enabled_fsck 0
```

8.

Sprawdzamy urządzenia i wolne miejsce, rebootujemy

```
mount
df -h
reboot && exit
```

9.

Instalujemy OpenVPN

```
opkg update; opkg install openvpn-openssl
```

10.

Generujemy certyfikaty PKI

```
opkg install openvpn-easy-rsa
source /etc/easy-rsa/vars     #Zignorować ostrzeżenia
clean-all
pkitool --initca

pkitool --server my-server
pkitool         my-client
build-dh
```

11.

Sprawdzamy czy są pliki index.txt, serial, i po parze .crt/.key dla każdej strony komunikacji. Do tego trochę innych plików.

```
ls $KEY_DIR
```

12.

Żeby przetransferować klucze z serwera do klientów przydałby się bezpieczny ftp.

```
opkg install vsftpd-tls
/etc/init.d/vsftpd enable
```

13.

Dodajemy użytkownika, ale do tego potrzeba nam kilka nowych pakietów

```
opkg update
opkg install shadow-useradd
opkg install sudo
useradd kot
passwd kot
mkdir -p /home/kot
chown kot:kot /home/kot
vi /etc/passwd
```

i zmieniamy ścieżkę z nazwą nowego użytkownika na:

```
kot:x:1000:1000:kot,ftp:/home/kot:/bin/ash
```

dodajemy mu uprawnienia do sudo:

```
visudo
```

a tam odkomentowujemy dwie linijki:

```
Defaults targetpw  # Ask for the password of the target user
ALL ALL=(ALL) ALL  # WARNING: only use this together with 'Defaults
targetpw'
```

14.

Resetujemy urządzenie i możemy zalogować się jako nowy użytkownik. Od teraz wszystkie polecenia administratora wykonywane będą za pomocą sudo. Wiąże się to z pewnymi ograniczeniami których nie będziemy póki co usuwać. Na przykład uprawnienia do sudo ma edytor vi, ale już nie vim.

15.

Konfigurujemy TLS. Najpierw generujemy certyfikaty:
https://www.digitalocean.com/community/tutorials/how-to-configure-vsftpd-to-use-ssl-tls-on-an-ubuntu-vps

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout
/etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
sudo vi /etc/vsftpd.conf
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
ssl_enable=YES
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
ssl_ciphers=HIGH
```

16.

Resetujemy usługę:

```
sudo /etc/init.d/vsftpd restart
```

17.

Łączymy się za pomocą FileZilla. Ustawienia protokołu to FTP, oraz "Require explicit FTP over TLS". Akceptujemy certyfikat, podajemy hasło, musi działać.

18.

Kopiujemy pliki klienta OpenVPN do bezpiecznego katalogu u siebie:

```
ca.crt
zacha-client.*
```

Discussion
For security reasons, you need to think long and hard about where you backup your PKI files, and especially the .key files:
* ca.key should be moved to a place that is not accessible from the Internet; it needed only when 'doing' CA stuff, such as creating certificates (but not using those certificates)
* the other .key files should be kept 'private', that is, stored only on the 'owning' system
* all .key files should never be distributed in an insecure manner - it is well known that copying .key files across the Internet is a leading cause of male-pattern baldness!

19.

Konfiguracja ustawień sieciowych

Dalej postępujemy zgodnie ze standardową procedurą zabezpieczania routerów na linuxie.

20.

Testy

* Firewall średni lub wysoki, 192.168.1.2 jako DMZ
* Firewall niski