Routování na Mikrotiku

2 ISP v jednom routeru - (verze 1)

Nastavení ukazuje, jak nastavit Mikrotik s dvěma připojení k Internetu. Uvedený příklad posílá různé LAN sítě přes konkretní linku poskytovatele

- ether1 = ISP1 (10.0.0.0/24)
- ether2 = ISP2 (10.10.0.0/24)
- ether5 = LAN1 (192.168.1.0/24)
- ether6 = LAN2 (192.168.101.0/24

Defaultni nastavení je všechno posílat přes ISP1 a pouze LAN2 posílat přes ISP2.

```
/ip route
add distance=1 gateway=10.10.0.1 routing-mark=isp2
add distance=1 gateway=10.0.0.1
/ip route rule
add src-address=10.10.0.0/24 table=isp2
add src-address=192.168.101.0/24 table=isp2
add routing-mark=isp2 table=isp2
/ip firewall nat
add action=src-nat chain=srcnat out-interface=ether1 action=masquerade
add action=src-nat chain=srcnat out-interface=ether2 action=masquerade
```

Záložní konektivita (2 ISP v jednom routeru) - (verze 2)

Tento návod je popisuje i situaci, kdy záložní konektivitu poskytuje nějaký router, který sám provádí NAT, přiděluje DHCP, apod.

I. Příprava záložního rozhraní

V případě ethernetu je port např. ether5 je v nastavení switche potřeba vyřadit ze switche - nastaví se u něj master port: None V případě USB donglu by mělo v Mikrotiku existovat rozhraní např. ppp1

II. přes SSH (i ve winboxu) se zadají tato pravidla:

```
[admin@odien] /ip> /ip route
[admin@odien] /ip route> add dst-address=8.8.8.8 gateway=ether1-gateway
scope=10
[admin@odien] /ip route> add dst-address=8.8.4.4 gateway=ether5 scope=10
[admin@odien] /ip route> add distance=1 gateway=8.8.8.8 routing-mark=ISP1
check-gateway=ping
[admin@odien] /ip route> add distance=2 gateway=8.8.4.4 routing-mark=ISP1
check-gateway=ping
[admin@odien] /ip route> add distance=1 gateway=8.8.4.4 routing-mark=ISP2
check-gateway=ping
```

```
[admin@odien] /ip route> add distance=2 gateway=8.8.8.8 routing-mark=ISP2
check-gateway=ping
```

III. navíc, např. přes klikací rozhraní:

1. na záložním interfacu (ether5 resp. ppp1) se musí pustit DHCP client s vyšší default distance přidaných router, než na ether1-gateway (např. jsem dal 10)
2. na záložním interfacu (ether5 resp. ppp1) se musí ve Firewall sekce NAT zapnout stejný IP Masquerading jako předím na ether1-gateway

Záložní konektivita (2 ISP v jednom routeru) - (verze 3)

- bez kontroly default gw a se skriptem - resi poradi bran.
- ether1 - primarni ISP
- ether5 - sekundarni zalozni ISP
- pro testovani funkcni primarni linky se pouziva sekundarni google dns (8.8.4.4), ktere je dostupne JEN pres primarni konektivitu
- predpoklada se, ze je na kazdem WAN interface pouze jedna ip adresa

```
/ip address add address=10.11.23.10/24 comment=primarni-isp interface=ether1
network=10.11.23.0
/ip address add address=172.16.23.20/24 comment=zalozni-isp interface=ether5
network=172.16.23.0
/ip route add distance=1 gateway=10.11.23.1
/ip route add distance=10 gateway=192.168.8.1
/ip route add distance=1 dst-address=8.8.4.4/32 gateway=10.11.23.1
/ip firewall nat add action=masquerade chain=srcnat comment="primarni-isp:
masquerade" out-interface=ether1
/ip firewall nat add action=masquerade chain=srcnat comment="sekundarni-isp:
masquerade" out-interface=ether5
```

Po priprave tohoto nastaveni je potreba vlozit tento skript do scheduleru: system > scheduler, skript vlozte sem a nastavte, aby se vykonaval treba kazdych 10 sekund

```
# Please fill the WAN interface names
:local InterfaceISP1 ether1
:local InterfaceISP2 ether5

# Please fill the ping check host - currently: secondary dns google
:local PingTarget 8.8.4.4

# Please fill how many ping failures are allowed before fail-over happends
:local FailTreshold 3

# Editing the script after this point may break it
# -------------- stop editing here --------------

# Declare the global variables
:global PingFailCountISP1
:global PingFailCountISP2
```

```
# This inicializes the PingFailCount variables, in case this is the 1st time
the script has ran
:if ([:typeof $PingFailCountISP1] = "nothing") do={:set PingFailCountISP1 0}
:if ([:typeof $PingFailCountISP2] = "nothing") do={:set PingFailCountISP2 0}

# This variable will be used to keep results of individual ping attempts
:local PingResult



# Check ISP1
:set PingResult [ping $PingTarget count=1 interface=$InterfaceISP1]
:put $PingResult

:if ($PingResult = 0) do={
    :if ($PingFailCountISP1 < ($FailTreshold+2)) do={
        :set PingFailCountISP1 ($PingFailCountISP1 + 1)
        :if ($PingFailCountISP1 = $FailTreshold) do={
            :log warning "ISP1 has a problem en route to $PingTarget -
increasing distance of routes."
            /ip route set 0 distance=10
            /ip route set 1 distance=1
                        /ip firewall connection {remove [find]}
            :log warning "isp 2 ACTIVE."
        }
    }
}
:if ($PingResult = 1) do={
    :if ($PingFailCountISP1 > 0) do={
        :set PingFailCountISP1 ($PingFailCountISP1 - 1)
        :if ($PingFailCountISP1 = ($FailTreshold -1)) do={
            :log warning "ISP1 can reach $PingTarget again - bringing back
original distance of routes."
            /ip route set 0 distance=1
            /ip route set 1 distance=10
                        /ip firewall connection {remove [find]}
            :log warning "spoje-net ACTIVE, disable backup isp."
        }
    }
}
```

V pripade nedostupnosti 8.8.4.4 se provede prohozeni priorit u vychozi brany tak aby se uprednostinila zalozni konektivita a nasledne se vymaze connection tabulka (jinak se zmena neprojevi pro jiz navazana spojeni !!!) V pripade, ze primarni konektivita opet naskoci, provede to stejne v obracenem poradi.

- *prevzato a upraveno pro vlastni potrebu z https://wiki.mikrotik.com/wiki/Failover_Scripting*

VPN v Mikrotiku

- V mikrotiku funguje korektně pouze od firmwaru 6.23
- Je nutno použít nešifrovanou OpenVPN, kterou si rozjedete např. na portu 443

1. Vygenerovaný certifikát nahrajeme přes scp do mikrotiku. (musíme nahrát ca.crt a pak uživatelský uzivatel.crt a uzivatel.key)
2. Importujeme nahrané certifikáty v pořadí ca.crt, uzivatel.crt, uzivatel.key. Passphrase nezadavame.

```
certificate import file-name=<zvoleny_soubor>
```

3. Nastavíme OpenVPN klienta. <uzivatel> nastavuje common name v importovanem certifikatu ! Za `aa.bb.cc.dd` dosadíme IP adresu našeho OpenVPN serveru

```
ppp profile add name=ovpn use-compression=no use-encryption=no use-mpls=no
use-vj-compression=no

interface ovpn-client add certificate=cert_2 cipher=null connect-
to=aa.bb.cc.dd mode=ethernet name=ovpn-out1 port=443 profile=ovpn
user=<uzivatel>
```

4. Manuálně si budeme muset přidat routy, které chceme posílat primárně přes VPN. Bohužel u Mikrotiku blbě funguje předávání rout, proto si musíte zjistit, jaká se používá brána ve VPN a tu nastavit manuálně do routovací tabulky

```
/ip route
add distance=1 dst-address=10.1.0.0/16 gateway=172.16.1.1
add distance=1 dst-address=10.2.0.0/16 gateway=172.16.1.1
add distance=1 dst-address=10.3.0.0/16 gateway=172.16.1.1
```

Internet přes OpenVPN v Mikrotiku s mobilním připojení

Pokud potřebujeme Internet tunelovat přes OpenVPN v Mikrotiku (např. cenzura internetu, chceme vlastní konektivitu etc.) postupujeme následujícím způsobem:

1. Přidáme rozhraní ppp-out1 (pro případ, že máme Internet přes mobilní připojení)

```
/interface ppp-client
add add-default-route=no apn=internet disabled=no name=ppp-out1 port=usb1 \
    use-peer-dns=no
```

2. Přidáme maškarádu na odchozí OpenVPN rozhraní (pokud používáme NAT a nechceme routovat vnitřní adresy)

```
/ip firewall nat
```

```
add action=masquerade chain=srcnat comment="default configuration" \
    out-interface=ppp-out1
add action=masquerade chain=srcnat comment="default configuration" \
    out-interface=ovpn-out1
```

3. Přidáme default routu přes OpenVPN - Musím přidat vyjímku na adresu OpenVPN serveru, který musíme poslat přes původní konektivitu

```
/ip route
add distance=1 gateway=172.16.1.1
add distance=1 dst-address=aa.bb.cc.dd/32 gateway=ppp-out1
```