# Linux router - poznámky

## nf_conntrack - nastavení conntracku na linux routeru

Conntrack se na routeru používá při zapnutí stavového firewallu nebo při nastavení NATu. Při natování většího poštu IP adres není často defaultní nastavení conntracku dostatečné a proto je potřeba ho upravit. V krajním případě jste na neoptimální nastavení dokonce upozorněni hláškou `nf_conntrack: table full, dropping packet` v syslogu.

Optimalizaci provádíme změnou hodnot parametrů **nf_conntrack_max** a conntrack hashsize reprezentovanou parametrem **nf_conntrack_buckets**. Aktuální hodnoty získáte pomocí příkazů:

```
$ cat /proc/sys/net/netfilter/nf_conntrack_max
65536
$ cat /proc/sys/net/netfilter/nf_conntrack_buckets
16384
```

Hodnotu nf_conntrack_max je vhodné nastavit podle velikosti dostupné operační paměti. nejprve zjistíme velikost jednoho záznamu v conntrack tabulce:

```
cat /proc/slabinfo
slabinfo - version: 2.1
# name            <active_objs> <num_objs> <objsize> <objperslab>
<pagesperslab> : tunables <limit> <batchcount> <sharedfactor> : slabdata
<active_slabs> <num_slabs> <sharedavail>
...
...
nf_conntrack_3         0      0    240   17    1 : tunables  120   60    8 :
slabdata      0      0      0
nf_conntrack_2         0      0    240   17    1 : tunables  120   60    8 :
slabdata      0      0      0
nf_conntrack_1         0      0    240   17    1 : tunables  120   60    8 :
slabdata      0      0      0
nf_conntrack_expect      0      0    184   22    1 : tunables  120   60    8
: slabdata      0      0      0
```

Hodnota **240** je v mém případě hodnota jednoho záznamu v tabulce conntrack. Dejme tomu, že můj router má 2GB RAM, pak pro conntrack tabulku použiju max. 1GB RAM. Použiju vzorec **velikost RAM v bytech / 240 = nf_conntrack_max** (1073741824 / 240 = 4473924,26667) tj. 4473924. Hashsize se potom vypočítá jako nf_conntrack_max / 8 tj. 4473924 / 8 = 559240

Nastavení nových parametrů pro nf_conntrack

/etc/sysctl.conf

```
..
..
#ipconntrack
```

```
net.ipv4.netfilter.ip_conntrack_max=4473924
net.ipv4.netfilter.ip_conntrack_tcp_timeout_established=7200
```

a

/etc/modprobe.d/nf_conntrack.conf

```
options nf_conntrack hashsize=559240
```

obojí je možné změnit za chodu bez restartu

```
echo 559240 > /sys/module/nf_conntrack/parameters/hashsize
sysctl -p
```

# Pokud na routeru mám pouze stavový firewall

V takovém případě je možné conntrack úplně vypnout, zvlášt pokud mám firewall jen na INPUT Do skriptu pro firewall, volaný přes iptables-restore přidáme následující řádky

```
*raw
:PREROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A PREROUTING -d 1.2.3.4/32 -j ACCEPT
-A PREROUTING -d 5.6.7.8/32 -j ACCEPT
-A PREROUTING -j CT --notrack
-A OUTPUT -s 1.2.3.4/32 -j ACCEPT
-A OUTPUT -s 5.6.7.8/32 -j ACCEPT
-A OUTPUT -j CT --notrack
COMMIT
```

Pravidla s IP adresou je potřeba vyjmenovat pro všechny IP adresy na interfacech routeru. **Pokud nepouzivame firewall na OUTPUTu, je mozne v tomto pripade pro notrack v outputu uplne vynechat**

V pripade, ze mate na routeru vetsi mnozstvi interfacu, je mozne pouzit ipset:

```
-A PREROUTING -m set --match-set local4 src -j ACCEPT
-A PREROUTING -m set --match-set local4 dst -j ACCEPT
-A PREROUTING -j CT --notrack
```

a v prislusnem ipsetu (v nasem pripade **local4**) vyjmenovat adresy na vsech lokalnim interfacech pomoci skriptu

```
#!/bin/bash

ipset flush local4
```

```
ipset flush local6

ipset -exist create local4 hash:ip comment timeout 0
ipset -exist create local6 hash:ip comment timeout 0 family inet6

for i in `ip a s | grep -o "inet [0-9\.]*" | cut -d ' ' -f 2 | sort -u`; do
        ipset -exist add local4 $i
done

for i in `ip a s | grep -o "inet6 [0-9a-f\:]*" | cut -d ' ' -f 2 | sort -u`;
do
        ipset -exist add local6 $i
done
```

## NAT a tunelované VPN

Pokud nám zlobí některé služby např. PPTP, je potřeba na routeru přidat automatické zavádění modulů:

```
ip_conntrack
ip_conntrack_ftp
ip_conntrack_pptp
ip_gre
ip6_gre
gre
ip_nat
ip_nat_ftp
ip_nat_pptp
nf_conntrack
nf_conntrack_ipv4
nf_conntrack_ipv6
nf_conntrack_proto_gre
nf_nat_proto_gre
ppp_mppe
pptp
pppoe
nf_nat_pptp
```

V kernelu od verze 4 je ještě potřeba zapnout přes sysctl používání nat helper, které je defaultně vypnuté:

```
net.netfilter.nf_conntrack_helper=1
```

## Optimalizace

Optimalizace pro 10G karty

```
# 10GB/54MB (56623104)
net.core.rmem_max = 56623104
net.core.wmem_max = 56623104
net.core.rmem_default = 56623104
net.core.wmem_default = 56623104
net.core.optmem_max = 40960
net.ipv4.tcp_rmem = 4096 87380 56623104
net.ipv4.tcp_wmem = 4096 65536 56623104
```

Optimalizace poctu sousedu a ARP cache

```
#   For IPv4
net.ipv4.neigh.default.gc_thresh1=8192
net.ipv4.neigh.default.gc_thresh2=12228
net.ipv4.neigh.default.gc_thresh3=24456
# For IPv6
net.ipv6.neigh.default.gc_thresh1=8192
net.ipv6.neigh.default.gc_thresh2=12228
net.ipv6.neigh.default.gc_thresh3=24456
```

Optimalizace souvisejici s firewallem

```
#number of incoming connections
net.core.somaxconn = 2048

#Maximum number of remembered connection requests
net.ipv4.tcp_max_syn_backlog = 30000

# Increase the tcp-time-wait buckets pool size to prevent simple DoS attacks
net.ipv4.tcp_max_tw_buckets = 2000000

# Decrease TIME_WAIT seconds
net.ipv4.tcp_fin_timeout = 10
```