

Iptables pro začátečníky

Vladislav Kurz, M.Sc.

`vladislav.kurz@webstep.net`

Malá anketa na úvod

- Kdo neví co je firewall?
- Kdo nikdy nenastavoval firewall?
- Kdo nastavoval FW jenom v GUI? (Linux, Windows, „krabičky“)
- Kdo nastavoval FW přes příkazový řádek?
- Kdo opravdu věděl co přesně nastavuje?

Co lze s iptables dělat

- Filtrování paketů – omezení přístupu na základě IP adres a portů
- Překlad adres – sdílení IP adres, přesměrování dat na jiný port/IP
- Sledování provozu – logování paketů, počítadla dat
- Specialitky – změny v paketech, routování, QoS, . . .

Malé opakování TCP/IP

IP (RFC 791) adresace, fragmentace

ICMP (RFC 792) chybová hlášení, ping

TCP (RFC 793) zaručený transport, multiplex, navazuje spojení, řeší ztráty a duplikace paketů, zachovává pořadí

UDP (RFC 768) jednoduché datagramy, multiplex, nízká režie

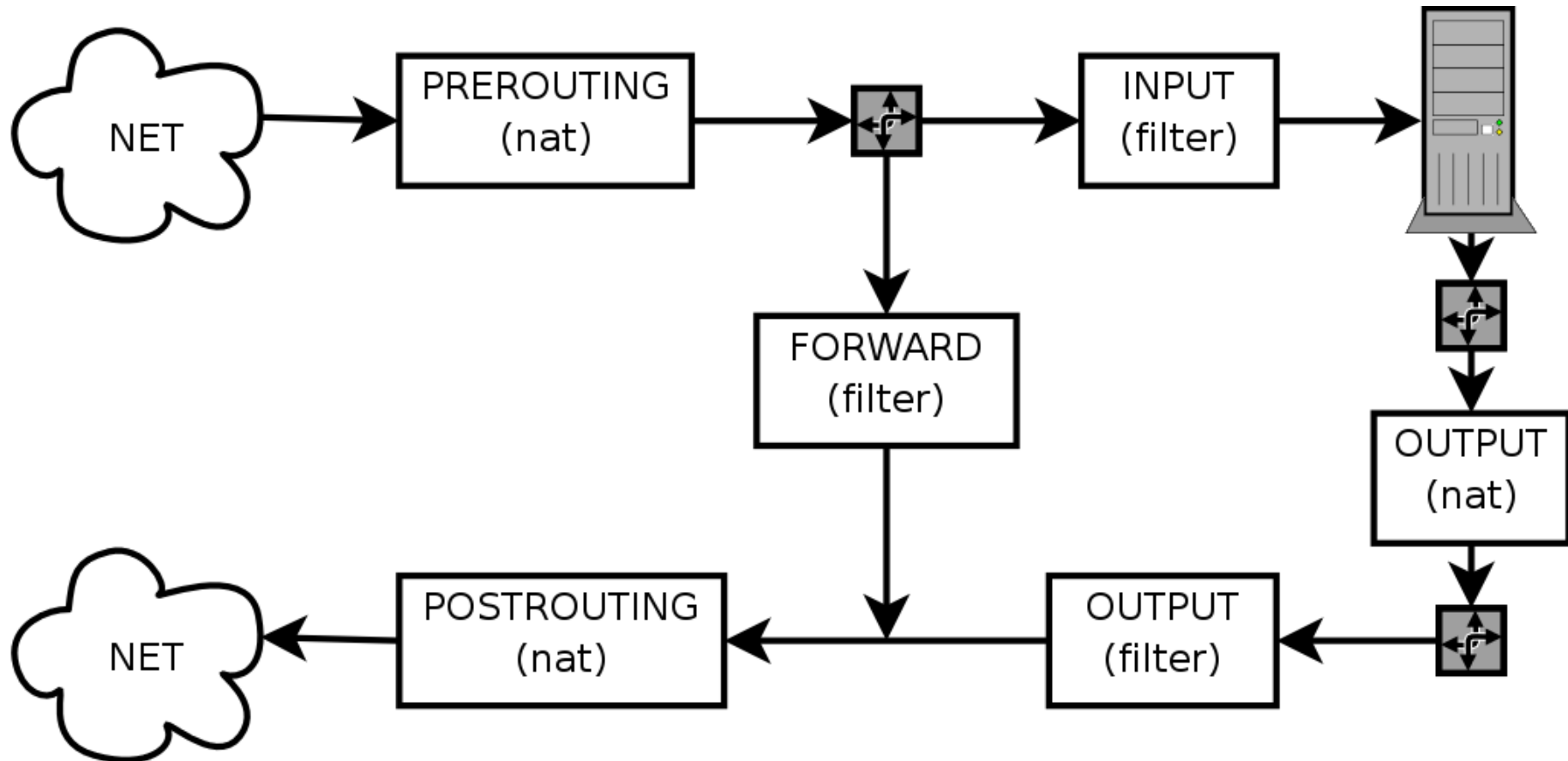
Příklad UDP – DNS dotaz

- `src=10.1.2.3, dst=10.0.0.1, proto=UDP; sport=1234, dport=53`
- `src=10.0.0.1, dst=10.1.2.3, proto=UDP; sport=53, dport=1234`
- Dotaz má náhodný zdrojový port, odpověď má porty prohozeny
- Ztrátu dat řeší aplikace – po čase zkusí jiný server nebo ohlásí chybu. Chybný checksum znamená zahození dat.
- Dotaz na neexistující cíl (port, IP) vrátí ICMP unreachable

Příklad TCP – 3-way handshake

- src=A, dst=B, proto=TCP; sport=X, dport=80, flags=SYN
- src=B, dst=A, proto=TCP; sport=80, dport=X, flags=SYN,ACK
- src=A, dst=B, proto=TCP; sport=X, dport=80, flags=ACK
- V každém dalším paketu je ACK. Spojení se ukončuje výměnou paketů FIN,ACK. Chyby (např. neotevřený port) hlásí flag RST.
- Každý paket nese sekvenční číslo odesílaných a potvrzených dat

Struktura netfiltru – tables and chains



Tabulky

filter – (default) vhodná pro základní filtrování, logování, počítání

nat – překlad adres (maškaráda, port forwarding), používá se pouze pro první paket spojení

mangle – specialitky. (TOS, MARK) Lze využít při QoS a routování

Zadávání pravidel

```
iptables [-t table] command chain [N] [match options] [-j target]
```

- A add** – přidá pravidlo na konec
- I insert** – přidá pravidlo na pozici N (začátek)
- D delete** – smaže pravidlo číslo N
- R replace** – přepíše pravidlo číslo N
- F flush** – smaže všechny pravidla
- P policy** – nastaví výchozí politiku
- Z zero** – vynuluje počítadla
- L list** – vypíše nastavená pravidla
 - v verbose** – vypíše i počítadla a interfacery
 - n numeric** – vypíše IP adresy, porty jako čísla
 - x exact** – vypíše počítadla v bytech

Podle čeho filtrujeme – match options

```
iptables [-t table] command chain [N] [match options] [-j target]
```

- p **proto** – protokol (TCP, UDP, ICMP, ...)
- s **source** – zdrojová IP adresa
- d **destination** – cílová IP adresa
- i **incoming** – příchozí síťová karta
- o **outgoing** – odchozí síťová karta
- sport** – zdrojový port (TCP/UDP)
- dport** – cílový port (TCP/UDP)
- icmp-type** – druh ICMP zprávy
- tcp-flags** – lze kontrolovat které flagy jsou nastaveny
- syn** – úvodní paket TCP spojení (SYN bez ACK)

Všechny pravidla lze negovat vykřičníkem (!)

Co s paketem uděláme – target

```
iptables [-t table] command chain [N] [match options] [-j target]
```

ACCEPT – povolíme (a dalších pravidel v chainu se neptáme)

DROP – zahodíme (a tváříme se jakoby nic)

REJECT – zahodíme, ale pošleme zpět ICMP unreachable

LOG – zapíšeme info z hlavičky do logu a pokračujeme dál

DNAT – změníme cílovou adresu nebo port (nebo obojí)

SNAT – změníme zdrojovou adresu nebo port (nebo obojí)

REDIRECT – speciální DNAT, pošle na lokální stroj

MASQUERADE – speciální SNAT, přepíše adresu na vlastní

MARK – přiřadí paketu číslo, které lze použít pro další zpracování

Pravidlo nemusí mít target, v tom případě se pokračuje dál a pouze se zvedne počítadlo.

Konečně příklad

Domácí PC s webserverem. Ven povolit vše, příchozí pouze HTTP.

```
iptables -F
```

```
iptables -P INPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -A INPUT -p tcp --dport=80 -j ACCEPT
```

Pojede to? Kde je chyba?

Opravený příklad

Domácí PC s webserverem. Ven povolit vše, příchozí pouze HTTP.

```
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -p tcp ! --syn -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p udp --dport 1024: -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
```

Pořád to není ideální.

Stavový firewall – connection tracking

- Defragmentace paketů
- Probíhá v chainu PREROUTING a OUTPUT
- Rozeznává stavy: NEW, ESTABLISHED, RELATED, INVALID
- Některé protokoly potřebují speciální podporu: `ip_conntrack_*`
- Podobná podpora je potřebná i pro NAT: `ip_nat_*`

Stejný příklad potřetí

Domácí PC s webserverem. Ven povolit vše, příchozí pouze HTTP.

```
iptables -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
```

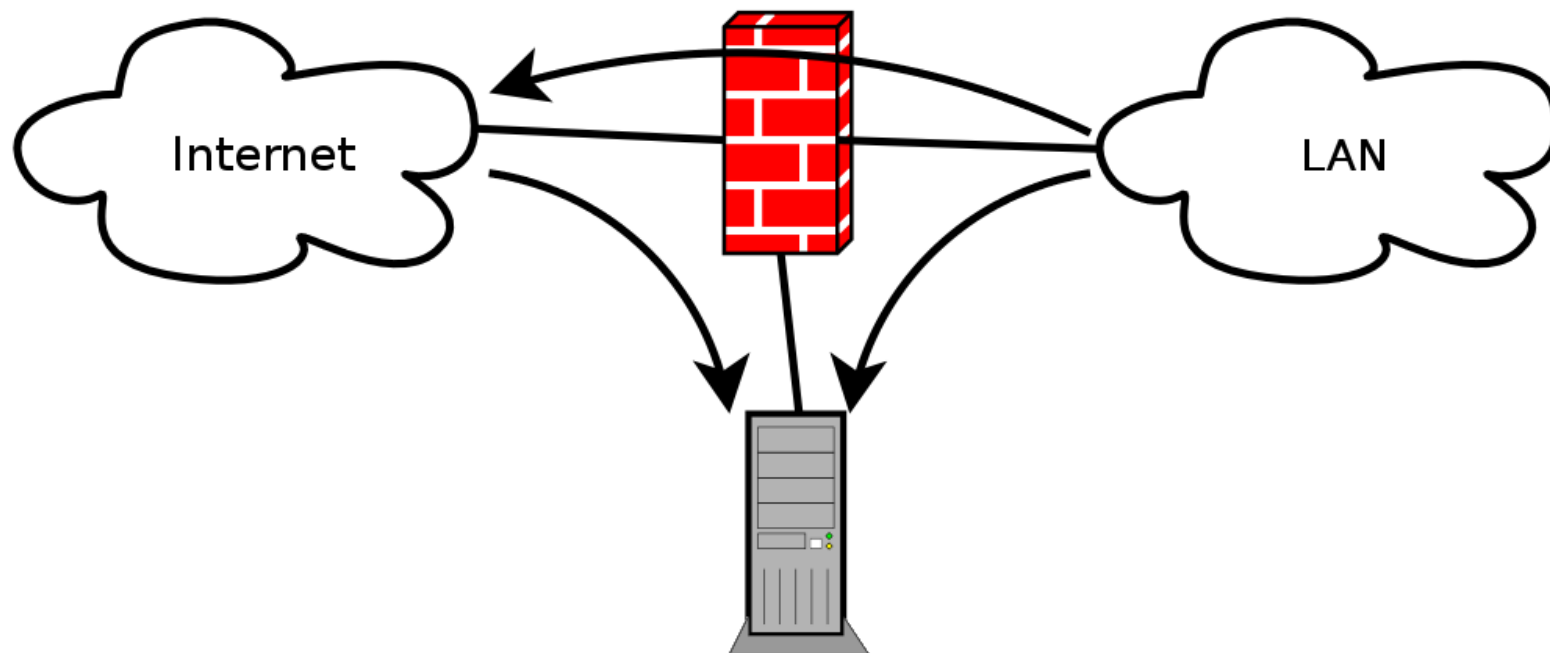
Překlad adres

Máme jednu IP adresu pro celou síť, uvnitř je navíc server.

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT \
    --to-destination 10.1.1.1
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -d 10.1.1.1 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth1 -j ACCEPT
iptables -P FORWARD DROP
```

eth0 je internet, eth1 je lokální síť, flush a policy accept jsem vynechal

Demilitarizovaná zóna – DMZ



I když se někomu podaří průnik do DMZ tak z ní stále nemůže útočit na LAN.

Dotazy a odkazy

- man iptables
- <http://www.netfilter.org/documentation/>

Bonus: User chains – zpřehlednění FW

- Chainy se prochází sekvenčně. Pomocí chainů lze optimalizovat složitý firewall s mnoha pravidly.
- Opakované kombinace akcí lze dát do chainu. Lze je pak volat ze všech chainů ve stejné tabulce.
- **-N** nový chain, **-X** smazat chain, **-E** přejmenovat chain
- **-j RETURN** vrátí se za pravidlo které zavolalo tento chain.

Bonus: Příklady user chainů

```
iptables -N LOGDROP
```

```
iptables -A LOGDROP -j LOG
```

```
iptables -A LOGDROP -j DROP
```

```
iptables -N DMZ
```

```
iptables -A DMZ -p tcp --dport 25 -j ACCEPT
```

```
iptables -A DMZ -p tcp --dport 80 -j ACCEPT
```

```
iptables -A DMZ -j DROP
```

```
iptables -A FORWARD -d 10.1.1.1 -j DMZ
```

Bonus: Užitečná rozšíření – extensions

- m **mac** –**mac-source** **xx:xx:xx:xx:xx:xx** kontroluje zdrojovou MAC adresu, lze použít k jednoduché kontrole zda uživatelé používají přidělenou IP adresu.
- m **limit** –**limit** **X/t** omezí pravidlo na X krát za t (s,m,h,d). Vhodné pro omezení logování, nevhodné pro omezování rychlosti (počítá pakety, ne data)

Rozšíření (extensions) je spousta. RTFM man iptables.

Bonus: Aktivní a pasivní FTP

- FTP navazuje dvě TCP spojení: příkazy a data.
- Příkazové spojení začíná klient, destination port=21
- Klient řekne serveru jestli chce pasiv nebo aktiv.
- Pasivní: server otevře náhodný port a klient se na něj připojí.
- Aktivní: **klient** otevře náhodný port a server se na něj připojí. Spojení ze serveru má **source** port=20 !