# Y36SPS: Firewalling laborka

***OBECNĚ***

**firewall = IPTABLES** = program, který hlida sitovou komunikaci
- musi pres nej projit kazdy paket, který jde z nebo do PC
- tzn. umi IN, OUT, FW a NAT (= network address translation)
- obecne jde o spojeni IP + PORT
- každý paket ma zdrojovou a cilovou IP, kod a typ paketu (TCP, UDP a ICMP)

---

zakladni prikazy **iptables**

**`iptables –L`** vypise aktualni nastaveni polozek (se jmeny IP adres)
**`iptables -L -n`** vypise aktualni nastaveni polozek (jen s IP adresama, je to rychlejsi, protože nepreklada adresy)

**`input`** = prichozi pakety
**`output`** = odchozi paktey
**`forward`** = pruchozi pakety

stav **`ACCEPT`** = povoleno, prijmout
**`DROP`** = zakazano, „zahodit“

**iptables ma dve „tabulky“** **- filter** (nastavena implicitne)
**- nat**

prikaz **`iptables –t <filter> –L`**
t = tabulka
filter = nazev filtru, neni nutno psat, pokud nechci nat

---

**prikazy** co se muzou hodit

**`ifconfig`** = zjistim sva sitova rozhrani
**`lo`** (loopback) = virtualni rozhrani „ja sam“
**`ethX`** = IP adresy na jednotlivych eth portech

**`ping`** = „ping“ na jinou IP
**`ping --help`** nebo **`ping -h`**
**`ping -c 5 IP`** = vypise jen zadany počet paketu (standardne 4)
**`ctrl+C`** = ukonceni behu ping

omezeni politiky
**`iptables –P`** **=** změna politiky
**`iptables –A`** = pridat pravidlo na konec
**`iptables –I`** = pridat pravidlo na zacatek

**`-j`** = co se ma nastavit DROP/ACCEPT
**`-D`** = smaze radek
**`-F`** = smaze vse

**-F INPUT** = smaze vse v INPUT
**-s <IP>** = zdrojova adresa (u INPUT)
**-d <IP>** = adresa, na kterou se posila (u OUTPUT)
**-m** = rozsireny match, napr. u ESTABLISHED, RELATED

**whoami** = kdo jsem

**su www-data** = prihlasit se jako www-data:

**modprobe ipt_owner** = load mod

---

---

**LABORATORNI ULOHA**

**Nastavit prostredi – viz soubor „zakladni“.**

---

**Začít bodem 2. !!!**

**2. Výchozí politika: co není povoleno, je zakázáno.**

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

tail -f /var/log/messages
telnet ip port
```

---

**1. Nakonfigurujte paketový filtr iptables tak, aby:**

- **POVOLOVAL ESTABLISHED a RELATED TCP SPOJENÍ, jinak vám nebude chodit na TCP vůbec nic**

**ESTABLISHED** (ustanoveny, zalozeny) = povoli prijem paketu z IP pro které povolim spojeni, komunikaci musim zahajit pouze ja

**RELATED** () = pouze pro FTP spojeni, který pouziva 2 datove toky – port 21 (posilam data) a 20 (potvrzeni)
muze byt **ACTIVE** = 1 spojeni jde od klienta k FTP a druhé jde v aktivnim rezimu z FTP
**PASSIVE** = obema kanaly jdou pouze data od klienta

otevrene porty pro prenos jsou vzdy vetsi nez 1024

prikaz (pro pruchozi, tj. FORWARD zatím nenastavujeme)

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

---

- **(n) byly povoleny odchozí spojení ze všech (imaginárních) strojů (za vašim PC-routrem) na port 80 do internetu**

muj PC je PC-router, tzn. musim na FORWARD povolit ACCEPT pro tcp a udp

```
iptables -A FORWARD -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -p udp --dport 80 -j ACCEPT
```

- **(y) pokud si to chcete zkusit, povolte to na 192.168.9.2 port 3128**
  vyzkouset to neslo

---

- **(n) byly povoleny odchozí spojení na porty 0-1024 do internetu pro váš počítač (router)**

```
iptables -A OUTPUT -p tcp --dport 0:1024 -j ACCEPT
iptables -A OUTPUT -p udp --dport 0:1024 -j ACCEPT
```

---

- **(y) uživatel www-data nemohl iniciovat žádné odchozí spojení mimo loopback interface**

```
iptables -I OUTPUT -m owner --uid-owner www-data -d 127.0.0.1 -j ACCEPT
```

nebo

```
iptables -I OUTPUT -m owner --uid-owner www-data ! -d 127.0.0.1 -j DROP
```

prepojeni do uzivatele www-data `su www-data`
kontrola zda jsem tam `whoami`
navrat `exit`

zjisteni id uzivatele `id -u www-data`
kontrola funkcnosti
- uid-owner nelze otestovat pres ping, lze pouze necim, co funguje pres TCP pakety
- zkusili jsme nainstalovat traceroute, ale stejne to nefungovalo

---

- **(n) uživatel tvrdik měl povoleno iniciovat pouze spojení na port 80 do internetu**

nejprve pridat uzivatele tvrdik `adduser tvrdik`

nyní nastavime pozadavek – musí byt jako první prikaz

```
iptables -I OUTPUT -m owner --uid-owner tvrdik -p tcp --dport 80 -j ACCEPT
iptables -I OUTPUT -m owner --uid-owner tvrdik -p udp --dport 80 -j ACCEPT
```

- **(y) pokud si to chcete zkusit, povolte to na lokální (localhost) port 631**

  test `telnet localhost 631`

---

- **(y) bylo povoleno spojení na port 22 z počítače cvičícího 192.168.9.100, nejvýše však 2 za minutu**

```
iptables -A INPUT -s 192.168.9.100 -p udp --dport 22 -m limit
--limit 2/m -j ACCEPT

iptables -A INPUT -s 192.168.9.100 -p tcp --dport 22 -m limit
--limit 2/m -j ACCEPT
```

---

- **(n) fungovalo aktivní FTP pro odchozí i příchozí spojení z/do celého internetu**

dle bodu 4. je internet 0/0, takze prikaz pak bude takto
```
iptables -A INPUT -p tcp --dport 21 -s 0/0 -j ACCEPT
```

jiz povoleno, takze je zbytecne uvadet znovu
```
iptables -A OUTPUT -p tcp --dport 21 -s 0/0 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 20 -s 0/0 -j ACCEPT
```

v OUTPUT musí byt povoleno na port 20 a 21
prikazy v OUTPUT jsou nadbytecne, protože je povolen port 0-1024 – viz vyse

- **(y) šlo by ověřit z lokální sítě**
  nepodarilo se nam odzkouset

---

- **(y) logovaly se odchozí spojení na port 12345**

```
iptables -A OUTPUT -p tcp --dport 12345 -j LOG
iptables -A OUTPUT -p udp --dport 12345 -j LOG
```

vypis se uklada do souboru var/log/messages - lze sledovat zde
```
tail -f /var/log/messages
```

---

- **(y) logovaly se příchozí spojení na port 54321 z počítače vašeho kamaráda v místní síti**

```
iptables -A INPUT -p tcp --dport 54321 -s 192.168.9.200 -j LOG
iptables -A INPUT -p udp --dport 54321 -s 192.168.9.200 -j LOG
```

test = povolit INPUT a OUTPUT ve vychozi politice
telnet <IP port> `telnet 192.168.9.200 54321`

na PC teto IP se vypisuji logy – lze sledovat pomoci prikazu
```
tail -f /var/log/messages
```

---

- **(y blbě, ale šlo by) alespoň triviálním způsobem bránil SYN floodu**

**SYN flood** = počet prichozich tcp spojeni
trivialni reseni = omezime je na 3 za sekundu
bude POUZE pro tcp protokol

```
iptables -A INPUT -p tcp --syn -m limit --limit 1/s --limit-burst 3
-j RETURN
```

**RETURN** = konec, vic nebude prochazet

---

- **(n) fungoval překlad adres do světa (NAT)**

```
iptables -t NAT -A POSTROUTING -j MASQUERADE
```

vypis **`iptables -t -L`**

ještě se musí povolit echo1 = zapise do souboru ip_forward jednicku = povoleni
```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

---

**3. Vymyslete, jak spolehlivě otestovat funkčnost filtru.**

**filtr** = iptables

tzn. otestovat nejakym zpusobem to, co jsem napsala
kaslem na to, protože jedine co funguje je DROP v zakladni politice

---

**4. Internetem se rozumí rozsah 0/0**

- pravidla označené (y) se dají ověřit přímo v laborce
- pravidla označené (n) se nedají ověřit přímo v laborce
- cvičící si nepotrpí na konkrétní hodnoty, tzn. pokud místo 80 záměrně povolíte 3128 aby vám to fungovalo, jen do toho
- odevzdáváte funkční pravidla ve firewallu (iptables -L -n)