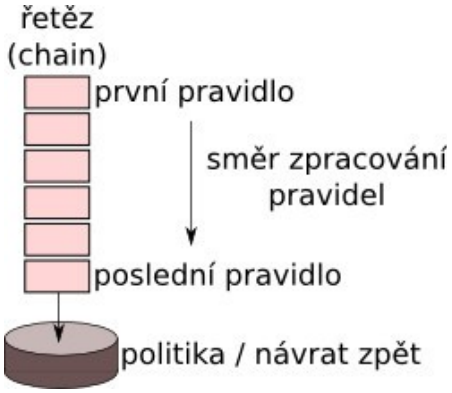
Téma 11: Firewall v CentOS

Nastavení firewallu

Téma 11: Firewall v CentOS

Teoretické znalosti

V této kapitole zjistíte, jak v distribuci CentOS nastavit připojení k síti a firewall.
Firewall v Linuxu je tvořen projektem Netfilter, který pracuje na úrovni jádra a umožňuje filtrovat pakety na základě mnoha kritérií.
Netfilter není pouze firewall, je to především paketový filtr, který umožňuje provádět řadu věcí, z nichž jedna možnost je vytvořit pravidla, která budou zastávat funkci firewallu.
Iptables provádí dynamické filtrování paketů, kterému se říká **stateful packet inspection**, je tak možné sledovat, jestli paket vytváří nové spojení, nebo patří k již existujícímu. Když nechceme nabízet žádné veřejné služby, zablokujeme všechny pokusy o vytvoření spojení zvenčí. Pakety k již existujícím spojení se ale dovnitř dostanou.
Základním nástrojem pro nastavení paketového filtru je řádkový nástroj **iptables**. Pomocí iptables nastavujeme řetězy, kterými prochází jednotlivé pakety:



Paket v řetězu putuje od prvního pravidla k následujícímu. To trvá do té doby, než některému pravidlu vyhoví. Při tom se provede jedna z možných akcí: ACCEPT – přijmout, DROP – zahodit, REJECT odmítnout, nebo předat do jiného řetězu. Jestliže žádný řetěz nevyhoví, aplikuje se na paket výchozí politika řetězu.
Existují tři tabulky: **filter**, ta je výchozí, **NAT** a **mangle**. Nejčastěji používanou je tabulka filter.
Tabulka filter obsahuje tyto řetězy:

- **INPUT** - zpracovává příchozí pakety
- **OUTPUT** - zpracovává odchozí pakety
- **FORWARD** - zpracovává pakety procházející přes zařízení

Tabulku NAT použijeme pro změnu adresy počítače v paketu. Můžeme tak za jednu veřejnou IP adresu schovat celou vnitřní síť. Je taky možné povolit přístup z venkovní sítě na vnitřní počítač s neveřejnou IP adresou.
Tabulka NAT obsahuje tyto řetězy:

Téma 11: Firewall v CentOS

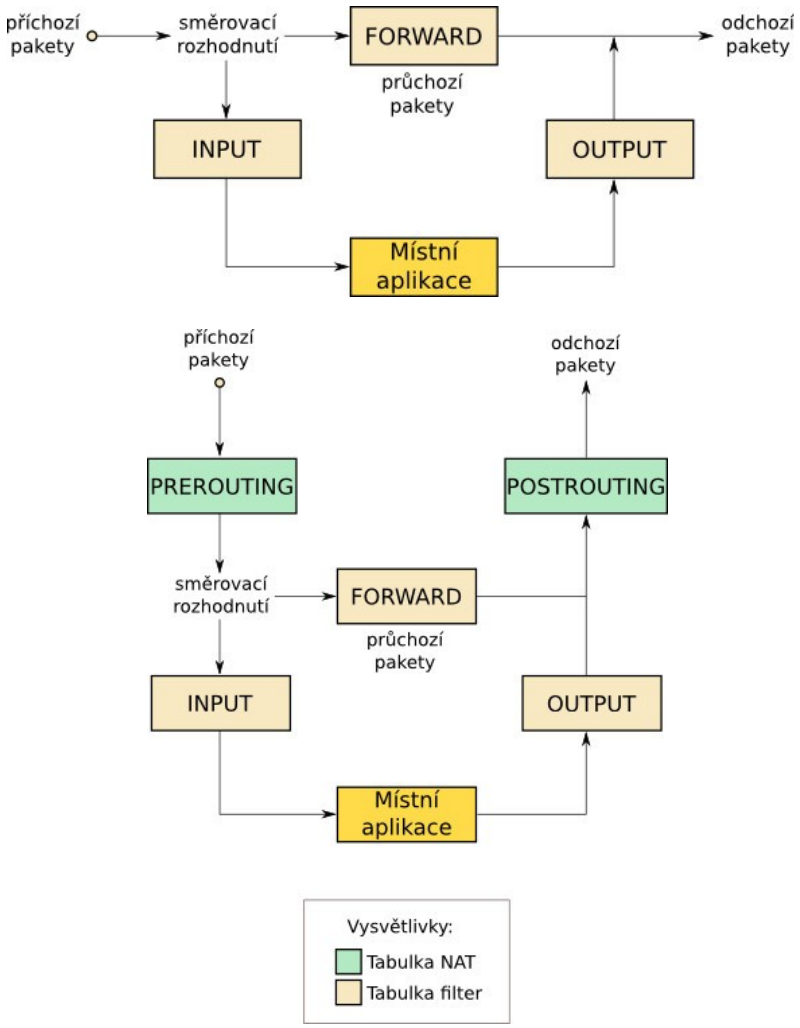
- **PREROUTING** – upravuje příchozí pakety před jejich směrováním
- **OUTPUT** - upravuje lokálně vytvořené pakety před jejich směrováním
- **POSTROUTING** – upravuje pakety po směrování

Pomocí tabulky mangle můžeme upravovat hlavičky paketů, to se hodí například pro úpravu TOS (pro QoS), nebo omezování rychlosti.

Tabulka mangle má tyto řetězce:

- **PREROUTING** – upravuje příchozí pakety před jejich směrováním
- **OUTPUT** - upravuje lokálně vytvořené pakety před jejich směrováním
- **INPUT** - upravuje pakety pro místní počítače
- **FORWARD** – zpracovává pakety směrované přes zařízení
- **POSTROUTING** – upravuje pakety po směrování

Pakety přicházející do sítě postupně procházejí jednotlivými tabulkami v následujícím pořadí: mangle → nat → filter.



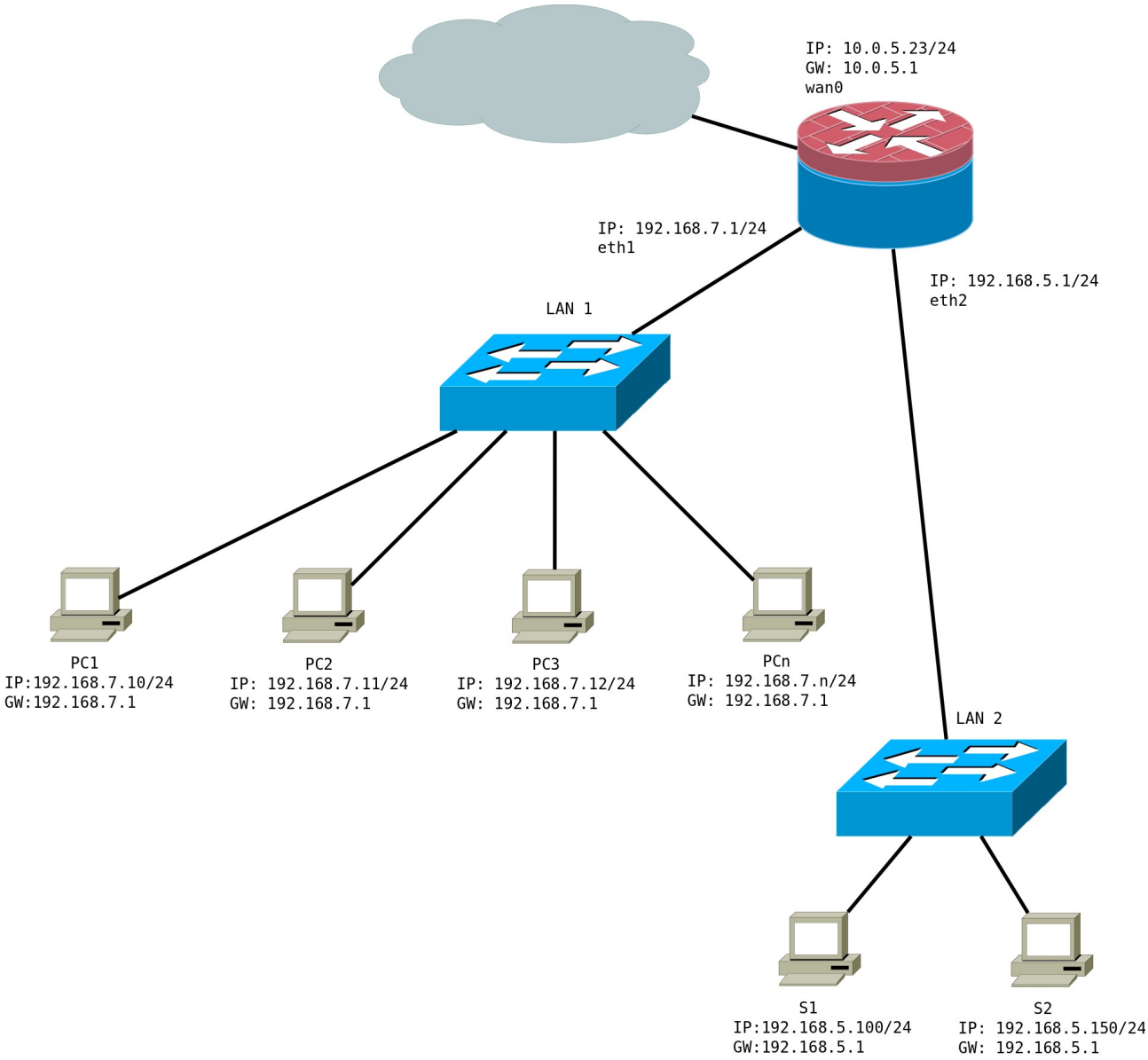
V jednotlivých řetězech je možné definovat vlastní pravidla pro filtrování paketů.

Téma 11: Firewall v CentOS

Zadání cvičení

1. Síťová karta připojená do Internetu se bude jmenovat „wan0“
2. Router bude připojen síťovou kartou „wan0“ do Internetu a „eth1“ LAN1 a „eth2“ do LAN2
3. Všechny karty budou mít staticky nastavené IPv4 adresy
4. Smazat všechna nastavení firewallu
5. Router bude dělat NAT pro LAN
6. Veškerý provoz z LAN do Internetu a na router bude povolen, z Internetu do LAN a na router zakázán
7. Již navázaná spojení povolit
8. ICMP bude povolené
9. Loopback rozhraní „lo“ bude povolené
10. Z Internetu se půjde připojit na SSH (TCP 22) na routeru
11. Na PC1 bude z Internetu dostupný web server. (PC1: TCP 80; router TCP 8080)
12. Na PC3 bude z Internetu dostupná vzdálená plocha VNC. (TCP 5900 až 5906)
13. Na S1 bude web server dostupný z Internetu, LAN1 a LAN2. (S1: TCP 80; router TCP 80)
14. Na S2 bude samba, dostupná pro LAN1 a LAN2, ale nebude dostupná pro počítač PC2. (TCP a UDP 137, 138, 139)
15. Neplatné pakety zahodit
16. Vše ostatní odmítnout s ICMP zprávou „13: Communication Administratively Prohibited“
17. Zajistit automatické spuštění firewallu po startu systému

Téma 11: Firewall v CentOS



Řešení

Síťová karta připojená do Internetu se bude jmenovat „wan0“

Nejdřív si zjistíme, jaké máme síťové karty pomocí příkazu **ifconfig -a**:

```
[root@localhost /]# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:BD:A5:AE
          inet addr:192.168.1.65  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:febd:a5ae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:710 errors:0 dropped:0 overruns:0 frame:0
          TX packets:374 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:105702 (103.2 KiB)  TX bytes:30650 (29.9 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:E5:DA:8E
          inet addr:10.0.3.15  Bcast:10.0.3.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee5:da8e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2493 (2.4 KiB)  TX bytes:4719 (4.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:480 (480.0 b)  TX bytes:480 (480.0 b)

[root@localhost /]#
```

Můžeme se také příkazem **lspci | grep "Network\|Ethernet"** podívat, jaký síťový hardware je v PC:

```
[root@localhost /]# lspci | grep "Network\|Ethernet"
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
00:08.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controller (rev 02)
[root@localhost /]#
```

Samotné přejmenování síťové karty se nastavuje v souboru **/etc/udev/rules.d/70-persistent-net.rules** :

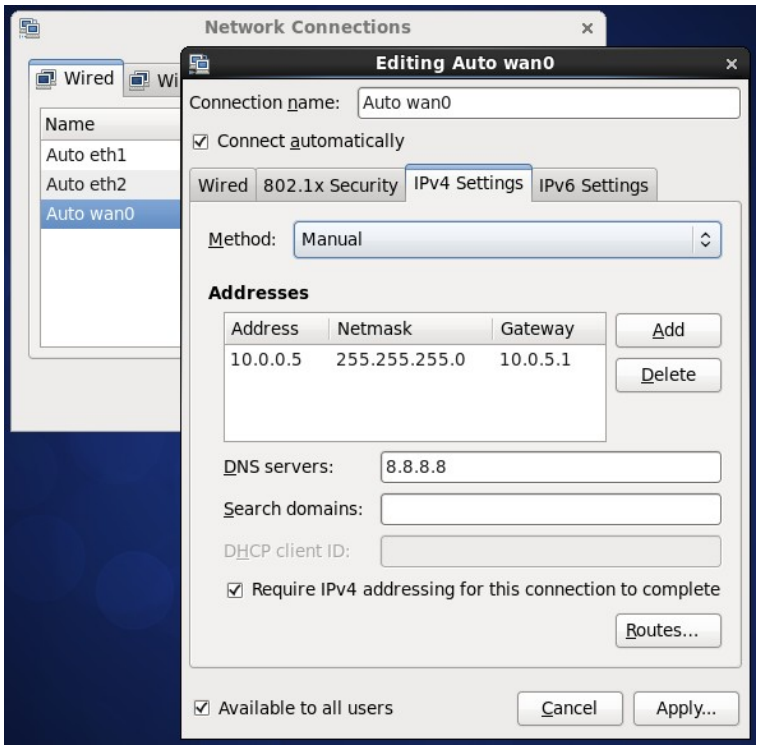
```
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:bd:a5:ae",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"

# PCI device 0x8086:0x100e (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="08:00:27:e5:da:8e",
ATTR{type}=="1", KERNEL=="eth*", NAME="wan0"
```

Podle MAC adresy si najdeme požadovanou kartu a upravíme parametr NAME.

Po restartování počítače se už bude síťová karta jmenovat wan0 – to si můžeme ověřit opět příkazem **ifconfig -a** .

Všechny karty budou mít staticky nastavené IPv4 adresy

IP adresy můžeme nastavit z grafiky: System → Preference → Network Connections:



nebo pomocí textových souborů. Ty musí být uloženy v adresáři **/etc/sysconfig/network-scripts/** a musí mít jméno **ifcfg-JmenoSitoveKarty**, tedy např. i**fcfg-wan0** .

```
[root@localhost /]# cat /etc/sysconfig/network-scripts/ifcfg-wan0
DEVICE=wan0
BOOTPROTO=none
IPADDR=10.0.5.23
PREFIX=24
GATEWAY=10.0.5.1
ONBOOT=yes
DNS1=8.8.8.8
DNS2=8.8.4.4
[root@localhost /]# cat /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
BOOTPROTO=none
IPADDR=192.168.7.1
PREFIX=24
ONBOOT=yes
[root@localhost /]# cat /etc/sysconfig/network-scripts/ifcfg-eth2
DEVICE=eth2
BOOTPROTO=none
IPADDR=192.168.5.1
PREFIX=24
ONBOOT=yes
[root@localhost /]#
```

- **DEVICE** – jméno fyzického zařízení
- **BOOTPROTO** – **dhcp** = získat nastavení z DHCP, **none** = ruční nastavení

- **IPADDR** – IP adresa
- **PREFIX** – místo prefixu je možné použít „**NETMASK=255.255.255.0**“
- **GATEWAY** – brána
- **ONBOOT** – aktivovat zařízení při bootování
- **DNS1**, **DNS2** – adresy DNS serverů

Na adrese https://www.centos.org/docs/2/rhl-rg-en-7.2/ch-networkscripts.html je k dispozici podrobný popis.

Jestliže použijeme nataveni pomocí **ifcfg** souborů, zakážeme **NetworkManager** a povolíme **network**:

```
[root@localhost /]# chkconfig NetworkManager off
[root@localhost /]# chkconfig network on
[root@localhost /]#
```

Firewall

Ještě před samotným nastavováním je potřeba smazat všechna předchozí pravidla:

```
[root@localhost /]# iptables -F
[root@localhost /]# iptables -t nat -F
[root@localhost /]# iptables -t mangle -F
[root@localhost /]# iptables -X
[root@localhost /]#
```

Smažeme všechna pravidla v tabulce filter (1), nat (2) a mangle (3) a nakonec (4) všechna uživatelská pravidla.

Aby bylo možné použít počítač jako router, je nutné povolit předávání paketů. V souboru **/etc/sysctl.conf** změníme net.ipv4.ip_forward=0 na **net.ipv4.ip_forward=1** . A znovu načteme nastavení příkazem **sysctl -p** .

Syntaxe iptables je následující:

```
iptables -A FORWARD -i eth1 -o eth2 -p udp -m multiport --dport 137,138,139 -j ACCEPT
```

- **-A** – přidáme nové pravidlo na konec řetězu
- **-i** – vstupní síťová karta
- **-o** – výstupní síťová karta
- **-p** – protokol
- **-m** – načtení modulu

Téma 11: Firewall v CentOS

- **--dport** – cílový port
- **-j** – akce, která se vykoná
- **-s** – zdrojová adresa
- **-d** – cílová adresa

Odebrání pravidla je možné provést jeho přesným opsáním, ale zaměněním **-A** za **-D** :

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -D INPUT -p tcp --dport 22 -j ACCEPT
```

! je negace. Podrobnou nápovědu získáme příkazem **man iptables** .

Nejdříve smaže všechna pravidla ze všech tabulek:

```
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X
```

Poté nastavíme výchozí politiku. Vstupní a procházející pakety zahodíme, odchozí povolíme:

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

Spojení z LAN do Internetu a na router povolíme:

```
iptables -A FORWARD -o wan0 -i eth1 -j ACCEPT
iptables -A FORWARD -o wan0 -i eth2 -j ACCEPT
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A INPUT -i eth2 -j ACCEPT
```

Již navázaná spojení z Internetu do LAN, nebo na router povolíme:

```
iptables -A FORWARD -i wan0 -o eth1 -m state --state=RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i wan0 -o eth2 -m state --state=RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -i wan0 -m state --state=RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Povolíme místní smyčku:

Téma 11: Firewall v CentOS

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

Na routeru bude z Internetu dostupné SSH:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Povolíme ICMP:

```
iptables -A INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT
```

NAT – počítače v LAN1 a LAN2 budou moci přistupovat na Internet:

```
iptables -t nat -A POSTROUTING -o wan0 -j MASQUERADE
```

(paketům, které jdou do Internetu, změní zdrojovou IP na IP adresu wan0)

Požadavky z Internetu na port 8080 na routeru přesměrujeme do LAN1 na PC1:

```
iptables -t nat -A PREROUTING -i wan0 -p tcp --dport 8080 -j DNAT --to 192.168.7.10:80
iptables -A FORWARD -i wan0 -o eth1 -d 192.168.7.10 -p tcp --dport 80 -j ACCEPT
```

(První pravidlo změní cílovou adresu a port paketů příchozích z Internetu na wan0 na IP a port 192.168.7.10:8080 (PC1). Poté paket projde do řetězu FORWARD. Druhé pravidlo zajistí jeho průchod.)

Požadavky z Internetu na vzdálenou plochu VNC (porty 5900 až 5906) na routeru přesměrujeme do LAN1 na PC3:

```
iptables -t nat -A PREROUTING -i wan0 -p tcp --dport 5900:5906 -j DNAT --to 192.168.7.12
iptables -A FORWARD -i wan0 -o eth1 -d 192.168.7.12 -p tcp -j ACCEPT
```

Na serveru S1 bude z Internetu a LAN1 dostupný webserver:

```
iptables -t nat -A PREROUTING -i wan0 -p tcp --dport 80 -j DNAT --to 192.168.5.100
iptables -A FORWARD -i wan0 -o eth2 -d 192.168.5.100 -p tcp -j ACCEPT
iptables -A FORWARD -i eth1 -o eth2 -p tcp --dport 80 -j ACCEPT
```

Na S2 bude Samba (TCP a UDP 137, 138, 139), dostupná pro LAN1 a LAN2, ale nebude dostupná pro počítač PC2:

```
iptables -A FORWARD -i eth1 -s 192.168.7.11 -d 192.168.5.100 -j REJECT
```

Téma 11: Firewall v CentOS

```
iptables -A FORWARD -i eth1 -o eth2 -p tcp -m multiport --dport 137,138,139 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth2 -p udp -m multiport --dport 137,138,139 -j ACCEPT
```

Neplatné pakety zahodíme:

```
iptables -A INPUT -m state --state INVALID -j DROP
```

Všechny ostatní, které nevyhoví žádnému pravidlu odmítneme s ICMP zprávou „13: Communication Administratively Prohibited“ :

```
iptables -A INPUT -j REJECT --reject-with icmp-admin-prohibited
```

Aby správně fungovalo předávání paketů mezi jednotlivými sítěmi, musíme upravit routovací tabulku:

```
ip route flush table main
route add -net 192.168.7.0/24 gw 192.168.7.1 eth1
route add -net 192.168.5.0/24 gw 192.168.5.1 eth2
route add -net 10.0.5.0/24 gw 10.0.5.23 wan0
route add 0.0.0.0 gw 10.0.5.23 wan0
```

Nejdřív smažeme všechny záznamy. Poté přidáme záznamy pro obě LAN sítě a nakonec výchozí cestu:

```
[root@localhost pokus]# ip route flush table main
[root@localhost pokus]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
[root@localhost pokus]# route add -net 192.168.7.0/24 gw 192.168.7.1 eth1
[root@localhost pokus]# route add -net 192.168.5.0/24 gw 192.168.5.1 eth2
[root@localhost pokus]# route add -net 10.0.5.0/24 gw 10.0.5.23 wan0
[root@localhost pokus]# route add -net 0.0.0.0 gw 10.0.5.23 wan0
[root@localhost pokus]#
[root@localhost pokus]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.7.0     192.168.7.1     255.255.255.0   UG    0      0        0 eth1
10.0.5.0        10.0.5.23       255.255.255.0   UG    0      0        0 wan0
192.168.5.0     192.168.5.1     255.255.255.0   UG    0      0        0 eth2
0.0.0.0         10.0.5.23       0.0.0.0         UG    0      0        0 wan0
[root@localhost pokus]#
```

Takto nastavený firewall nepřežije restart počítače. Je nutné ho někam uložit. Při použití **/etc/rc.local** se firewall spustí příliš pozdě a úpravou **/etc/rc.d/init.d/network** riskujeme, že při aktualizaci systému (a tohoto souboru) o nastavení přijdeme. Nejlepší je, napsat vlastní init script.

Téma 11: Firewall v CentOS

Vytvoříme soubor **/etc/rc.d/init.d/firewall** s následujícím obsahem:

```
#!/bin/sh
#
# firewall   Start iptables firewall
#
# chkconfig: 2345 11 99
# description:      Start iptables firewall
#
# config: /usr/local/bin/firewall.sh
#
### BEGIN INIT INFO
# Provides: firewall
# Required-Start: $network
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: start and stop iptables firewall
# Description: Start, stop and save iptables firewall
### END INIT INFO

# Source function library.
. /etc/init.d/functions

# only usable for root
[ $EUID = 0 ] || exit 4

case "$1" in
    start)
       /usr/local/bin/firewall.sh
       ;;
    status)
       echo -e "--------------------------------\nROUTE:\n"
       route -n
       echo -e "\n--------------------------------\nIPTABLES:\n"
       iptables -L -n -v
       ;;
    *)
       echo "Pouzijte 'start', nebo 'status'"
       ;;
esac
exit
```

Téma 11: Firewall v CentOS

A skript s firewallem uložíme do **/usr/local/bin/firewall.sh** :

```
#!/bin/bash

iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X

iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

iptables -A FORWARD -o wan0 -i eth1 -j ACCEPT
iptables -A FORWARD -o wan0 -i eth2 -j ACCEPT
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A INPUT -i eth2 -j ACCEPT

iptables -A FORWARD -i wan0 -o eth1 -m state --state=RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i wan0 -o eth2 -m state --state=RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -i wan0 -m state --state=RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

iptables -A INPUT -p tcp --dport 22 -j ACCEPT

iptables -A INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT

iptables -t nat -A POSTROUTING -o wan0 -j MASQUERADE

iptables -t nat -A PREROUTING -i wan0 -p tcp --dport 8080 -j DNAT --to 192.168.7.10:80
iptables -A FORWARD -i wan0 -o eth1 -d 192.168.7.10 -p tcp --dport 80 -j ACCEPT

iptables -t nat -A PREROUTING -i wan0 -p tcp --dport 5900:5906 -j DNAT --to 192.168.7.12
iptables -A FORWARD -i wan0 -o eth1 -d 192.168.7.12 -p tcp -j ACCEPT

iptables -t nat -A PREROUTING -i wan0 -p tcp --dport 80 -j DNAT --to 192.168.5.100
iptables -A FORWARD -i wan0 -o eth2 -d 192.168.5.100 -p tcp -j ACCEPT
iptables -A FORWARD -i eth1 -o eth2 -p tcp --dport 80 -j ACCEPT

iptables -A FORWARD -i eth1 -s 192.168.7.11 -d 192.168.5.100 -j REJECT
```

Téma 11: Firewall v CentOS

```
iptables -A FORWARD -i eth1 -o eth2 -p tcp -m multiport --dport 137,138,139 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth2 -p udp -m multiport --dport 137,138,139 -j ACCEPT

iptables -A INPUT -m state --state INVALID -j DROP

iptables -A INPUT -j REJECT --reject-with icmp-admin-prohibited

ip route flush table main
route add -net 192.168.7.0/24 gw 192.168.7.1 eth1
route add -net 192.168.5.0/24 gw 192.168.5.1 eth2
route add -net 10.0.5.0/24 gw 10.0.5.23 wan0
route add 0.0.0.0 gw 10.0.5.23 wan0
```

Přidáme právo spuštění pro **/usr/local/bin/firewall.sh** a **/etc/rc.d/init.d/firewall** a zapneme init skript **chkconfig firewall on**:

```
[root@localhost /]# chmod +x /usr/local/bin/firewall.sh
[root@localhost /]# chmod +x /etc/rc.d/init.d/firewall
[root@localhost /]# chkconfig firewall on
[root@localhost /]#
```

Aktuální stav si můžeme ověřit příkazy **route -n** a **iptables -L -n -v** :

```
[root@localhost /]# iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1   295 ACCEPT     all  --  eth1   *       0.0.0.0/0            0.0.0.0/0
    1   295 ACCEPT     all  --  eth2   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  wan0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    4   270 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    2   120 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
    0     0 REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           reject-with icmp-admin-prohibited

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth1   wan0    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth2   wan0    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  wan0   eth1    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  wan0   eth2    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  wan0   eth1    0.0.0.0/0            192.168.7.10        tcp dpt:80
    0     0 ACCEPT     tcp  --  wan0   eth1    0.0.0.0/0            192.168.7.12
    0     0 ACCEPT     tcp  --  wan0   eth2    0.0.0.0/0            192.168.5.100
    0     0 ACCEPT     tcp  --  eth1   eth2    0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 REJECT     all  --  eth1   *       192.168.7.11         192.168.5.100       reject-with icmp-port-unreachable
    0     0 ACCEPT     tcp  --  eth1   eth2    0.0.0.0/0            0.0.0.0/0           multiport dports 137,138,139
    0     0 ACCEPT     udp  --  eth1   eth2    0.0.0.0/0            0.0.0.0/0           multiport dports 137,138,139

Chain OUTPUT (policy ACCEPT 1 packets, 40 bytes)
 pkts bytes target     prot opt in     out     source               destination
    6   390 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
[root@localhost /]#
```

Téma 11: Firewall v CentOS

Nebo jednodušeji využitím init skriptu **/etc/init.d/firewall status** :

```
[root@localhost /]# /etc/init.d/firewall status
------------------------------
ROUTE:

Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.5.23       255.255.255.255 UGH   0      0        0 wan0
192.168.7.0     192.168.7.1     255.255.255.0   UG    0      0        0 eth1
10.0.5.0        10.0.5.23       255.255.255.0   UG    0      0        0 wan0
192.168.5.0     192.168.5.1     255.255.255.0   UG    0      0        0 eth2

------------------------------
IPTABLES:

Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
   12  2875 ACCEPT     all  --  eth1   *       0.0.0.0/0            0.0.0.0/0
   12  2875 ACCEPT     all  --  eth2   *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  wan0   *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    2    80 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    2   120 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0           state INVALID
   12  2863 REJECT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           reject-with icmp-admin-prohibited

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth1   wan0    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth2   wan0    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  wan0   eth1    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  wan0   eth2    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  wan0   eth1    0.0.0.0/0            192.168.7.10        tcp dpt:80
    0     0 ACCEPT     tcp  --  wan0   eth1    0.0.0.0/0            192.168.7.12
    0     0 ACCEPT     tcp  --  wan0   eth2    0.0.0.0/0            192.168.5.100
    0     0 ACCEPT     tcp  --  eth1   eth2    0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 REJECT     all  --  eth1   *       192.168.7.11         192.168.5.100       reject-with icmp-port-unreachable
    0     0 ACCEPT     tcp  --  eth1   eth2    0.0.0.0/0            0.0.0.0/0           multiport dports 137,138,139
    0     0 ACCEPT     udp  --  eth1   eth2    0.0.0.0/0            0.0.0.0/0           multiport dports 137,138,139

Chain OUTPUT (policy ACCEPT 42 packets, 8853 bytes)
 pkts bytes target     prot opt in     out     source               destination
    4   200 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
[root@localhost /]#
```