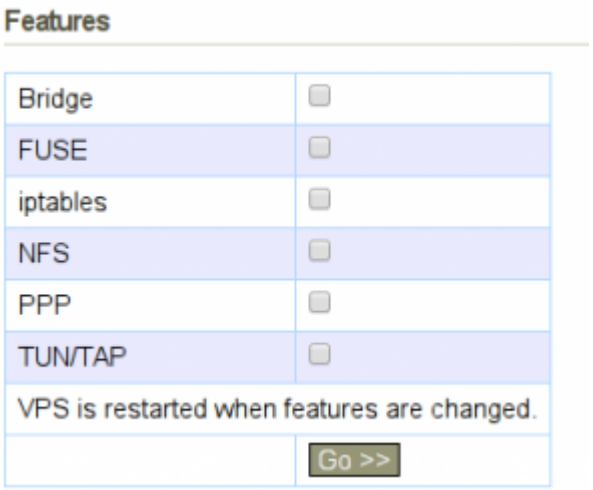
Firewall

Firewall slouží pro filtraci provozu na VPS a z něj. Pomocí něj je možné nastavovat pravidla pro konkrétní pakety, které jsou propouštěny na základe různých parametrů jako jsou zdrojová a cílová IP adresa, zdrojový či cílový port a podobně.

Linuxové jádro obsahuje firewall zvaný Netfilter, který je obsluhován pomocí utility `iptables`. Tento přístup je nízkoúrovňový a umožňuje nastavit velké množství různých parametrů a variant. Pohodlnější variantou je například nadstavba Shorewall, což je sada skriptů, které dovolují pravidla pro Netfilter z jednodušších konfiguračních souborů.

Na našem VPS je v každém případě nejprve potřeba povolit podporu iptables. Pokud to neuděláte, bude firewall fungovat jen v bezestavovém režimu, kdy nebudou fungovat moduly vyžadující conntrack (například NAT a jiné moduly závislé na stavu spojení). V detailu VPS je potřeba zapnout podporu iptables a potvrdit. Dojde k restartu VPS a podpora je zapnutá.



Při prvotních hrátkách s firewallem je možné, že si neopatrným zásahem odříznete cestu k VPS. Nepanikařte a přejděte na návod k opravě rozbité VPS.

IPtables

Současný stav je možné zjistit pomocí

```
iptables -L
```

Výstup bude vypadat pravděpodobně nějak takto

```
  Chain INPUT (policy ACCEPT)
  target     prot opt source               destination

  Chain FORWARD (policy ACCEPT)
  target     prot opt source               destination
```

```
  Chain OUTPUT (policy ACCEPT)
  target     prot opt source               destination
```

Ve výpisu jsou zobrazeny tři řetězce (chain): vstup, průchod a výstup. V nich v současné době nejsou žádná pravidla, která by regulovala průchod paketů. V takovém případě se uplatňuje výchozí politika (policy), která je ve všech třech případech nastavena na „povol“ (ACCEPT). Všechny pakety tedy nyní projdou bez filtrace.

Všechny tři řetězce jsou oddělené, paket prochází vždy jen jedním z nich. Pokud je zdrojem náš server, dostane se paket do řetězce OUTPUT. Pokud je naopak náš server cílem, paket přichází do INPUT. Pokud jsou zdroj i cíl mimo náš stroj a paket tedy jen prochází skrz, dostane se do FORWARD.

Je čas naučit náš firewall nějaká pravidla. Jejich syntaxe je následující:

```
řetězec pravidlo akce
```

Každé pravidlo musí patřit do nějakého řetězce, na procházející paket se pak aplikuje pravidlo a v případě vyhovění se provede akce. Tedy například: do řetězce INPUT přidej pravidlo „pokud je cílem TCP paketu port 80“, akce je „propusť“. Konkrétně:

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

Takto můžete ručně zapsat různá pravidla a sledovat chování. Takto zapsaná pravidla ale **nejsou perzistentní** a po restartu vašeho VPS zmizí. Pokud je potřebujete vysypat cíleně a zbavit se veškerého nastavení, použijte

```
 iptables -F
```

Pokud naopak potřebujete pravidla zachovat, zapište je například do souboru `/etc/iptables.rules`. Na pořadí zapsání záleží! Netfilter pak pravidla prochází sekvenčně a provede jen první akci, u které daný paket vyhověl pravidlu.

Pravidla se pak sice zapisují stejně, ale bez příkazu `iptables` na začátku. Řádek tedy začíná rovnou na -A...

```
*filter

# Pusť veškerý loopback (lo0) provoz a zahoď provoz z 127/8, který nepochází
z rozhraní lo0
-A INPUT -i lo -j ACCEPT
-A INPUT ! -i lo -d 127.0.0.0/8 -j REJECT

# Propusť všechen provoz už navázaných spojení
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Povol veškerý odchozí provoz
-A OUTPUT -j ACCEPT
```

```
# Otevři příchozí porty pro HTTP a HTTPS
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT

# Povol veškerá spojení na SSH
-A INPUT -p tcp -m state --state NEW --dport 22 -j ACCEPT
# Povol spojení jen z mé IP adresy
# -I INPUT -p tcp -s 1.2.3.4 --dport 22 -j ACCEPT

# Zakaž všechno ostatní, co prošlo až sem.
-A INPUT -j REJECT
-A FORWARD -j REJECT

COMMIT
```

Takový konfigurační soubor je možné získat i tak, že uložíte aktuální pravidla zapsaná právě ve firewallu. Pokud jste tedy pravidla zadali ručně z řádky, můžete si je všechna uložit:

```
iptables-save > /etc/iptables.rules
```

Stejně tak je možné obsah souboru znovu načíst do jádra:

```
iptables-restore < /etc/iptables.rules
```

Pokud chcete, aby se tento příkaz spouštěl při startu systému a tedy aby se při startu VPS načetla všechna zadaná pravidla, zapište do souboru `/etc/network/if-pre-up.d/iptables` tyto dva řádky:

```
#!/bin/sh
 /sbin/iptables-restore < /etc/iptables.rules
```

Soubor pak učiňte spustitelným:

```
chmod +x /etc/network/if-pre-up.d/iptables
```

Od této chvíle se při startu serveru aplikují firewallová pravidla podle vašich přání.

Shorewall

Nastavení firewallu se dělá pomocí balíku *shorewall*.

```
apt-get install shorewall
cd /etc/shorewall
```

/etc/shorewall/zones

Nastavení zón ($FW v ostatních souborech se automaticky nahrazuje „fw“).

```
#ZONE   TYPE            OPTIONS         IN                      OUT
#                                       OPTIONS                 OPTIONS
fw      firewall
net     ipv4
vpn     ipv4
```

/etc/shorewall/policy

Tohle je nastaveni implicitních akcí (vyhodnocuje se v zadaném pořadí!).

```
#SOURCE         DEST            POLICY          LOG     LIMIT:    CONNLIMIT:
#                                               LEVEL   BURST           MASK

# povol spojeni "ze serveru na internet"
$FW             net             ACCEPT

# zahod vsechno "z internetu na server"
net             all             DROP            info

# odmitni vsechno "z vpn na internet" (aby si vpn klienti nebrouzdali pres
server)
vpn             net             REJECT          info

# povol vsechno ostatni "z vpn"
vpn             all             ACCEPT

# The FOLLOWING POLICY MUST BE LAST
all             all             REJECT          info
```

/etc/shorewall/interfaces

```
FORMAT 2
###############################################################################
###
#ZONE           INTERFACE               OPTIONS
net             venet0                  tcpflags,logmartians,nosmurfs
vpn             tun0
```

/ets/shorewall/rules

Tady se nastavují jednotlivá pravidla, kterými se pak firewall řídí. V následujícím souboru jsou v komentářích vysvětleny jendotlivé příklady.

```
#ACTION   SOURCE            DEST         PROTO  DEST    SOURCE    ORIGINAL
RATE    USER/   MARK   CONNLIMIT   TIME    HEADERS     SWITCH
#                                           PORT    PORT(S)   DEST
LIMIT   GROUP
#SECTION ALL
#SECTION ESTABLISHED
#SECTION RELATED
SECTION NEW
```

```
# povoleni SSH sluzby pro klienty z internetu

ACCEPT    net               $FW          tcp    ssh
# - pro urcitou IP adresu
#ACCEPT    net:78.80.8.27    $FW          tcp    ssh
# - pro skupinu IP adres (subnet)
#ACCEPT    net:81.25.21.0/24 $FW          tcp    ssh

# OpenVPN
ACCEPT    net               $FW          udp    1194
ACCEPT    $FW               net          udp    -       1194

# WEB
ACCEPT    all               all          tcp    80
ACCEPT    all               all          tcp    443
```

/etc/default/shorewall

V tomto souboru je potřeba Shorewallu říci, že má začít pracovat a načítat konfigurační soubory.

```
startup=1
```

Poté můžeme nechat Shorewall zkontrolovat konfiguraci:

```
shorewall check
```

Pokud potvrdí `Configuration Validated` alebo `Shorewall configuration verified`, můžeme ho spustit

```
shorewall start|stop|restart|...
```

Další užitečné příkazy:

```
shorewall status
shorewall show
shorevall safe-start
shorewall safe-restart
```

Odkazy: shorewall.net/ , wiki.debian.org/HowTo/shorewall

FirewallD

Starší verze firewalld potřebují, aby existovaly soubory modulů jádra, i když jsou načtené.

Vyrobit jde lze jednoduše pár příkazy:

```
mkdir /□lib/□modules/□$(uname -□r)
touch /□lib/□modules/□$(uname -□r)/□modules.{builtin,order}
for i in /sys/module/*; do echo kernel/${i##**/}.ko; done >>
```

```
/lib/modules/$(uname -r)/modules.builtin
depmod -a
```