Administrace Unixu a sítí

```
inet6-adr: fe80::210:a4ff:fee1:9e5d/64 Rozsah:Linka
AKTIVOVÁNO VŠESMĚROVÉ_VYSÍLÁNÍ BĚŽÍ MULTICAST  MTU:1500  Metrika:1
RX packets:66690 errors:0 dropped:0 overruns:0 frame:0
TX packets:100149 errors:0 dropped:0 overruns:0 carrier:0
kolizí:0 délka odchozí fronty:0
RX bytes:21490419 (20.4 MiB)  TX bytes:10545763 (10.0 MiB)
```

7. Zabezpečení služeb, firewall

Miroslav Spousta

2006

```
bug:/home/qiq# getent passwd
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
Debian-exim:x:102:102::/var/spool/exim4:/bin/false
qiq:x:1000:1000:Miroslav Spousta,,,:/home/qiq:/bin/bash
sshd:x:100:65534::/var/run/sshd:/bin/false
identd:x:101:65534::/var/run/identd:/bin/false
messagebus:x:103:104::/var/run/dbus:/bin/false
gdm:x:104:105:Gnome Display Manager:/var/lib/gdm:/bin/false
hal:x:106:106:Hardware abstraction layer,,,:/var/run/hal:/bin/false
saned:x:109:109::/home/saned:/bin/false
bind:x:105:110::/var/cache/bind:/bin/false
smmta:x:107:111:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:108:112:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
test:x:1001:1001:Test User,,,:/home/test:/bin/bash
postfix:x:110:115::/var/spool/postfix:/bin/false
```

http://www.ucw.cz/~qiq/vsfs/

Xen

- virtuální servery (http://www.cl.cam.ac.uk/Research/SRG/netos/xen/)
- budete mít rootovská oprávnění ve virtuálním serveru
- přístup je po dobu výuky
- OS: Debian 3.1
- RAM: 32 MB, swap: 128 MB (/dev/sda2), root: 512 MB (/dev/sda1)
- několik síťových karet (eth0, eth1, ...)
- Síťové karty jsou propojeny virtuálními přepínači

Přístup k virtuálním serveru:

- **`ssh login@kozel.vsfs.cz`**
- **`xencons localhost 90xx`**
- xx je číslo serveru (login: **`root`**, heslo: žádné)

Zabezpečení služeb

- síťově dostupné služby mohou být snadno zneužity
  - někým z vnitřní (lokální) sítě i z Internetu
- některé služby nabízejí ověření uživatele, jiné nikoliv
- je možné uhodnout jméno a heslo k účtu na nějakém stroji
- obecné pravidlo říká, že služby by měly být povoleny pouze pro ty uživatele/počítače/sítě, které je budou využívat
- v Unixech je možné nastavovat přístup ke službám na různých úrovních
  - konfigurací služby (na kterých interface a adresách bude naslouchat)
  - pomocí TCP wrappers (knihovna, kterou používají některé programy)
  - firewallem – univerzální a velmi flexibilní řešení
- je vhodné kombinovat několik těchto metod dohromady

TCP/UDP porty

- každý uzel v síti má 65536 TCP portů + 65536 UDP portů
- na portu může čekat (naslouchat) server pro nějakou službu
  - neboli port je otevřený – server bude odpovídat na požadavky o spojení
- nebo je port neobsazený
  - neboli zavřený – TCP stack vrátí na požadavek RST
- 0 – 1023 jsou privilegované porty (může je obsadit jen root)
  - tyto porty patří mezi tzv. dobře známé porty, na kterých běží většina služeb Internetu (např. HTTP: 80, SMTP: 25, SSH: 21)
  - přiřazení portů ke službám můžete najít v /etc/services
- komunikovat se serverem můžeme pomocí programu **`telnet`** nebo **`netcat`**

```
vm1:~# netcat localhost 22
SSH-2.0-OpenSSH_3.8.1p1 Debian-8.sarge.4
```

nmap

- chceme-li zjistit, které služby na daném počítači poslouchají, můžeme všechny najednou zjistit pomocí příkazu nmap
- použití: **`nmap cíl`** (TCP) nebo **`nmap -sU cíl`** (UDP)
- nmap umožňuje použít více druhů scannování (man nmap)
- **`nmap -O`** zapíná OS finger-printing

```
k5-14:~# nmap 192.168.33.1
Interesting ports on 192.168.33.1:
(The 1657 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
7/tcp  open  echo
22/tcp open  ssh
37/tcp open  time
MAC Address: 00:AA:BB:00:00:01 (Unknown)

Nmap run completed -- 1 IP address (1 host up) scanned in 1.937 seconds
```

oscannujte sobě i kolegovi počítač (TCP i UDP porty)

(x)inetd

- služby, které nemusí běžet stále je možné spouštět pomocí inetd
  - „super internet daemon“
  - konfigurace v /etc/inetd.conf (/etc/xinetd.d/*)
  - definuje, pod jakým uživatelem bude daná služba běžet, zda vůbec, atd.
- služby, které jsou spouštěny inetd/xinetd je možné zjistit v konfiguračním souboru
  - služby, které nejsou potřeba je vhodné zakázat
- typické služby: echo, time, ftp, telnet, talk, tftp, …

zjistěte, které služby máte spuštěné (v /etc/inetd.conf)

  - vyzkoušejte službu echo (případně si ji povolte): nc localhost echo

TCP wrappers

- knihovna, která nabízí centralizovanou správu přístupu různým síťovým aplikacím
  - poštovní server, ftp server, ssh server, talk server
- používá konfigurační soubory /etc/hosts.allow a /etc/hosts.deny
  - obsahují řádky: daemon : klient
  - daemon je síťová služby (např. sendmail), klient je IP adresa/DNS jméno počítače
  - pokud vyhovuje řádek v /etc/hosts.allow, je přístup povolen
  - pokud v /etc/hosts.deny, je přístup odepřen, jinak je povolen
- klient může být také ALL, LOCAL (v DNS bez tečky), KNOWN (má DNS jméno), UNKNOWN (neznámé DNS jméno), PARANOID (pokud se DNS jméno neshoduje s použitou adresou)

```
/etc/hosts.allow:
in.tftpd: LOCAL, .my.domain

/etc/hosts.deny:
ALL: ALL
```

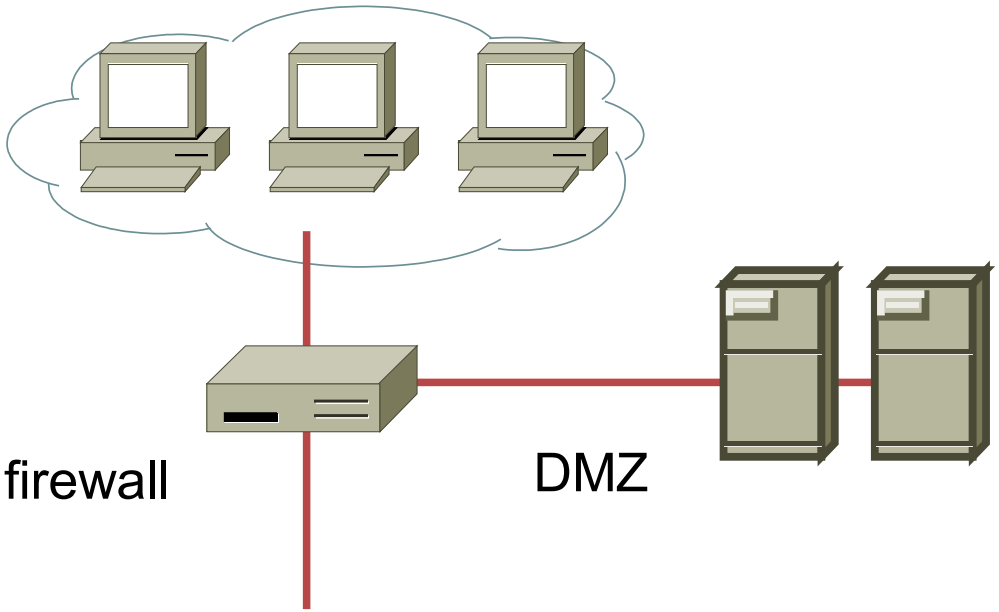
Firewall

- zařízení, které různým způsobem zajišťuje síť před vnějším světem
  - chrání data, přístup na zařízení, ...
- může být řešením na různých úrovních
  - linkové, síťové, aplikační
- HW nebo SW řešení
- paketový filtr
  - pracuje na síťové/transportní úrovni
  - nedívá se do „nákladu“ datagramů/paketů, ale do hlaviček
  - rozlišuje provoz podle spojení, portů, atd.
- aplikační brána
  - překládá provoz mezi vnitřní a vnější sítí
  - pracuje na aplikační úrovní, rozumí datům, které se přenášejí
  - např. HTTP proxy servery

> In construction, a firewall consists of a *windowless*, fireproof wall.
>
> -- http://wikipedia.org

Odbočka: DMZ

- neboli demilitarizovaná zóna
- servery, které jsou přístupny ze světa není dobré umisťovat za firewall
  - stávají se zranitelným místem infrastruktury
- vyhradí se pro ně speciální část sítě oddělená od vnitřní sítě
  - servery (a služby na nich jsou přístupné vnějším uživatelům (v Internetu), ale nejsou fyzicky uvnitř privátní sítě

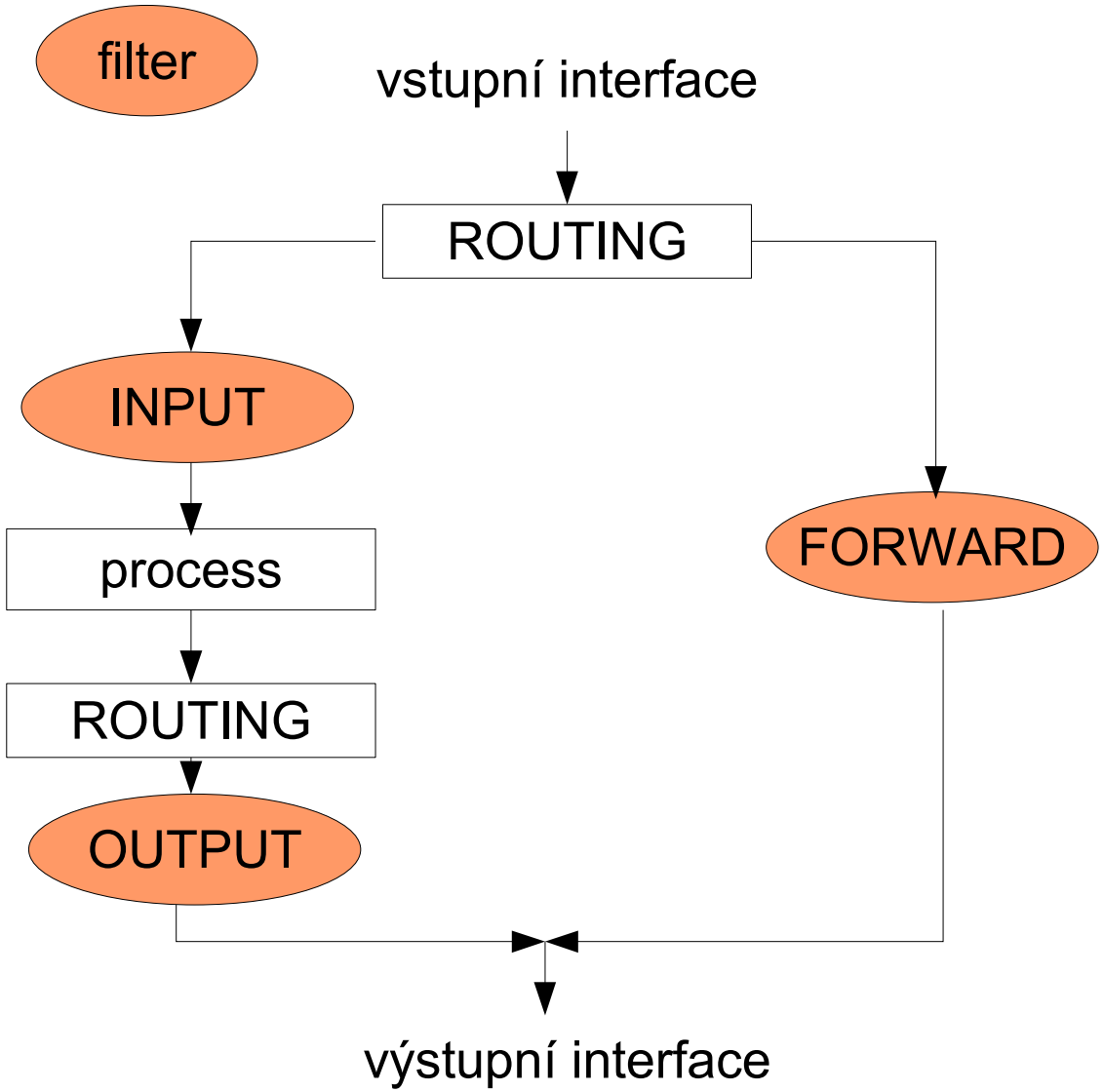


Paketový filtr

- paketový filtr umožňuje filtrovat provoz na základě nejrůznějších informací získaných z datagramu (nebo s ním souvisejících):
- linkové datagramy: MAC adresa
- IP: zdrojová, cílová adresa
- TCP: port a stav spojení
- UDP: port
- ICMP: typ zprávy
- dále můžeme filtrovat např. podle interface, ze kterého datagram přišel, případně na který interface bude datagram poslán, typu datagramu, ToS, ...

Cesta paketu

- datagramy vstupují na fyzickém interface
  - např. eth0
- každý paket projde jedním z filtrů
  - INPUT (pakety určené pro tuto stanici)
  - OUTPUT (pakety odeslané z této stanice)
  - FORWARD (pakety směrované)
- v těchto filtrech na něj útočí různá pravidla a můžou daný paket zastavit (např. úplně zničit), nebo propustit



iptables

- v Linuxu se nastavují pravidla filtru pomocí příkazu **`iptables`**
- každé pravidlo je uloženo v určité tabulce (table) v některém řetízku (chain)
- *tabulky* jsou pro jednotlivé druhy operací (filter, nat, mangle)
- *řetízky* jsou v rámci tabulek buď vestavěné (např. pro tabulku filter řetízky INPUT, OUTPUT, FORWARD), nebo uživatelsky definované
- výpis tabulky: **`iptables [-t filter] -L`**
- přidání pravidla do řetízku: **`iptables -A INPUT -p icmp -j ACCEPT`**
  - pravidlo se přidá do INPUT řetízku
  - můžeme specifikovat omezení, které paket musí splňovat, aby pravidlu vyhovoval
    - zde pouze omezení na icmp datagram
  - definujeme, co se má s paketem provést, pokud vyhovuje omezením
    - má se skončit zpracování v tomto řetízku a datagram se akceptuje
    - další možnosti: DROP, REJECT, LOG, SNAT, DNAT, MASQUERADE, ..., případně jméno jiného řetízku

iptables

- vestavěné řetízky mají tzv. default policy
  - co se aplikuje na datagram, pokud nevyhovuje ani jednomu pravidlu (většinou DROP)
  - **`iptables -P INPUT DROP`**
- pravidla mohou specifikovat mnoho různých omezení
- obecná omezení (platí pro všechny IP datagramy):
  - **`-s 195.113.31.123`** zdrojová adresa, **`-d 195.113.31.123`** cílová adresa
  - **`-i eth0`** zdrojový interface, **`-o eth1`** cílový interface
- protokol TCP: **`-p tcp`**
  - **`--sport 80`** zdrojový port paketu, **`--dport 10:12`** cílový port paketu
  - **`--tcp-flags SYN,FIN,ACK SYN`** které příznaky (druhý argument) z kterých (první argument) musí být nastaveny
- protokol UDP: **`-p udp`**
  - **`--sport 80`** zdrojový port paketu, **`--dport 10:12`** cílový port paketu

iptables

- protokol ICMP: **`-p icmp`**
  - **`--icmp-type 8`** typ icmp zprávy
- **`-m mac`**: **`--mac-source 00:00:00:00:00:11`**
- u pravidel je možné uvést vykřičník, jako negaci, např.: **`! --dport 80`**
- **`iptables -F INPUT`**
  - flush neboli vyprázdní řetízek (pozor, nemění default policy(!))

zakažte přístup na službu daytime na svém počítači

- *případně jen pro adresy různé od localhost*

zakažte přístup zvenku na všechny porty pro svůj počítač

- *a teď zkuste spojení (třeba ssh) ven*

iptables

- potřebujeme rozeznávat, které pakety patří kterému spojení (a v jakém je dané spojení právě stavu)
- paketový filtr v Linuxu je stavový
  - každý paket je v jedné z těchto kategorií: NEW, ESTABLISHED, RELATED, INVALID
  - NEW: paket patří spojení, které jsme ještě neviděli (typicky SYN pro TCP)
  - ESTABLISHED: paket patří spojení, které bylo navázáno (SYN ACK)
  - RELATED: paket byl vyvolán spojením, které bylo navázáno (např. ICMP k navázanému spojení)
  - INVALID: nepatří k žádnému spojení a je nějakým způsobem divný (např. má neplatnou hlavičku)
- stavy se pamatují i pro UDP datagramy (a ICMP)
- pro každý protokol a stav existuje timeout po kterém se spojení prohlásí za přerušené

iptables

- v pravidlech se můžeme odvolávat také na stav spojení, ke kterému daný paket patří (NEW, ESTABLISHED, RELATED, INVALID)
- **`-m state --state RELATED,ESTABLISHED`**
  - pravidlo bude vyhovovat jen paketům, které patří k již navázaným spojením
- stav aktuálních spojení můžete zjistit v /proc/net/ip_conntrack

zakažte zvenku (eth0) všechny porty pro svůj počítač

- *povolte spojení dovnitř pro* **`RELATED,ESTABLISHED`**
- *a teď zkuste spojení (třeba ssh) ven – mělo by fungovat*

*nastavte si logování pro nedovolené pakety (**`-j LOG`**)*