# OS Linux - Vzdálená správa Firewall - konfigurace, ovládání síťových zařízení. Naplánované úlohy, LOG soubory.

## Vzdálená správa

Vzálenou zprávu linux zařízení nebo jiného zařízení jako mouhou být switch nebo router budeme nejčastěji províďět za pomocí SSH (Secure shell) přez port **22**. Pomocí cmd se lze připojit příkazem **shh** následujícím způsobem **ssh uživatel@adresa\_zarizeni**.

Předchůdce ssh byl telnet port 23 také se používal na vzdálené připojení pouze nemělšifrované spojení. U ssh zajišťuje šifrování TLS. SSH je defaultně nakonfigurované jako domovský shell (můžem přez něj pouštět programy)

## Firewall

Jedná se o filter traficu který vychází nebo vchází do sítě či přímo do zařízení. Může kontrolovat traffic na úrovni paketu (3 vrstave OSI) či segment/datagram (4 vrstva OSI) nebo přímo aplikační vrstvě tedy protokoly jako jsou http, smtp, ftp, ssh (7 vrstva OSI).

Na linuxu je pro kontrolování provozu nejčastěji používán iptables nebo ufw. Jedná se o programy které nám dovolují nastavovat paket filtry.

### Iptables

Pravidla jsou provedena hned po zapsání (pokud by jsme tedy upravovali pravidla vzdáleně a zakázali jsme SSH už jse k počítači nepřipojíme dokud ho někdo nerestartuje).

Pravdila se neukládájí na disk takže jsou po rebootu smazaná. POkud chceme aby byla při startu načítána musíme si to neprogramovat sami (třeba vytvořením skriptu který se bude volat on reboot v cronu).

Prvidla probíhají chrnologicky. A nejnovější pravidlo je zapsáno na konec.

#### Pravidla

* INPUT … na vstupt, počítač dostává
* OUTPUT … na výstupt, počítač odesílá
* FORWARD … průchozí, odkud kam pokud jsou dvě síťovky (s pomocí masquerade můžeme zajistit překlad ipv4 adresy a tím vytvořit NAT)

Obsah obrázku text, snímek obrazovky, Obdélník, displej

Popis byl vytvořen automaticky

#### Politika

* ACCEPT … vše je v pořádku komunikace může proběhnout (je propuštěna)
* DROP … paket je zahozen ale nedá o sobě vědět
* REJECT … paket je zamítnut a pošle odesílateli o tom info

Obsah obrázku text, snímek obrazovky, design

Popis byl vytvořen automaticky

Pokud se vyhodnotí jeden z předchozích stavů tak jse dale v chainu pravidla již nevyhodnocují.

Pokud paket projde až na konec a nebude odpovídat žádenému pravidlu použíje se na něj defaultní politika. Defaultně při staření iptables je defaultní politika nastvena ACCEPT což není úplně dobře.

Problém s značení rozhraní. Pokud by jsme odstranili nebo nějak jinak měnili síťové rozhraná počítače mohlo by dojít k přejmenování.

* integrované síťové karty jsou pojmenovány eno0,  eno1 …
* karty ve slotech PCI Express jsou pojmenovány ens0,  ens1 …
* karty s více konektory přidávají k názvům ještě pozici:  enp1s0
* (volitelné) název odvozený z MAC adresy: enx78e7d1ea46da

Soket obsahuje kombinaci ip adresy a portu

#### Konfigurace

Stateless vs Stateful (conntrack)

* Stateful zná stav každého TCP a UDP připojení k systémů
* Když přijde nový paket, Stateful vý jestli se jedná o část již existujícího připojení
* Nevýhody stateful je že je udržování těchto informací velmi náročné a drahé u velmi zatíženého systému (okolo 1000 paketů výše za sekundu)

##### Add INPUT prvaidla

* iptables -A INPUT -s 1.2.3.4 -j ACCEPT
* iptables -A INPUT -p tcp --dport 22 -j DROP

##### List pravidel

* iptables -L -v
* iptables -n -L -v --line-numbers
* iptables -L INPUT -n -v
* iptables -L OUTPUT -n -v --line-numbers
* iptables -L INPUT --line-numbers

##### Mazani přidaného pravidla

* # přidání pravidla
* iptables -A INPUT -i eth0 -s 10.0.1.5 -p tcp --dport 22 -j ACCEPT
* # odebrání pravidla
* iptables -D INPUT -i eth0 -s 10.0.1.5 -p tcp --dport 22 -j ACCEPT
* #smazeni všech
* iptables -F – flush

##### Výchozí politika

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

##### Další

* iptables -A INPUT -s 1.2.3.4 -j DROP
* iptables -A INPUT -s 192.168.0.0/24 -j DROP
* iptables -A INPUT -p tcp --dport 80 -j DROP
* iptables -A INPUT -i eth1 -p tcp --dport 80 -j DROP
* iptables -A OUTPUT -d 1.2.3.4 -j DROP
* iptables -A OUTPUT -d 192.168.1.0/24 -j DROP
* iptables -A OUTPUT -o eth1 -d 192.168.1.0/24 -j DROP

##### Stateful využívá čtyři příkazy

* **NEW** - paket vytváří nové spojení nebo se vztahuje ke spojení, kde dosud neproběhla obousměrná komunikace
* **ESTABLISHED** - paket se vztahuje ke spojení, kde probíhá obousměrná komunikace (tedy již k vytvořenému spojení)
* **RELATED** - paket vytváří nové spojení, ale vztahuje se k některému z existujících (např. u FTP)
* **INVALID** - paket se nevztahuje k žádnému známému spojení (obvykle je to paket, který tu nemá co dělat, a proto je dobré ho rovnou zahazovat)

###### Použití

* iptables -A INPUT **-m state --state NEW** -m tcp -p tcp --dport 22 -j ACCEPT
* iptables -A INPUT **-m state --state NEW** -p udp --dport 53 -j ACCEPT
* iptables -A INPUT **-m state --state NEW** -p tcp --dport 80 -j ACCEPT

##### Logování

* iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j LOG --log-prefix "IP\_SPOOF A: "
* iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP

### UFW

* Sada skriptů pro IPtables
* Pomocí ufw lze nastavit:
  + pravidla pro běžící služby
  + pravidla pro aplikace
  + pravidla pro konkrétní porty a rozsahy portů

#### Konfigurace

* sudo ufw enable
* sudo ufw disable
* sudo ufw status
* ufw status verbose

##### Použití

* ufw allow in ssh
* Porty a protokoly
* sudo ufw allow 53
* sudo ufw deny 23/tcp
* sudo ufw deny ssh

##### Logování

* ufw logging on              --> zapne (úroveň low)
* ufw logging off              --> vypne
* ufw logging LEVEL        --> zapne na určitou úroveň

## Ovládání síťových zařízení

### Permanentně ip adresu Debian

sudo nano /etc/network/interfaces

#### from:

allow-hotplug enp0s5

iface enp0s5 inet dhcp

#### to:

# The loopback network interface

auto lo

iface lo inet loopback

# The primary network interface

auto enp0s5

iface enp0s5 inet static

address 192.168.2.236

netmask 255.255.255.0

gateway 192.168.2.254

dns-domain sweet.home

dns-nameservers 192.168.2.254 1.1.1.1 8.8.8.8

### Dočasná ip adresa

sudo ip addr add 127.255.255.255/16 dev eth0

sudo ip route add default via <gateway\_IP>

sudo ip route add <network/mask> via <gateway\_IP>

* ifconfig, ip, route … všchny tyto příkazi se používají pro nastavení sítí
* ifconfig .. nastavuje ip již zastaralý
* route .. na routovaci tabulku již zasatralý
* ip .. nahrazuje ty predchozi ale puvodni stale funguji

po rebootu to neprezije musim editovat nejaky soubor v debianu /etc/network/interfaces a jinak

### DNS dočasně

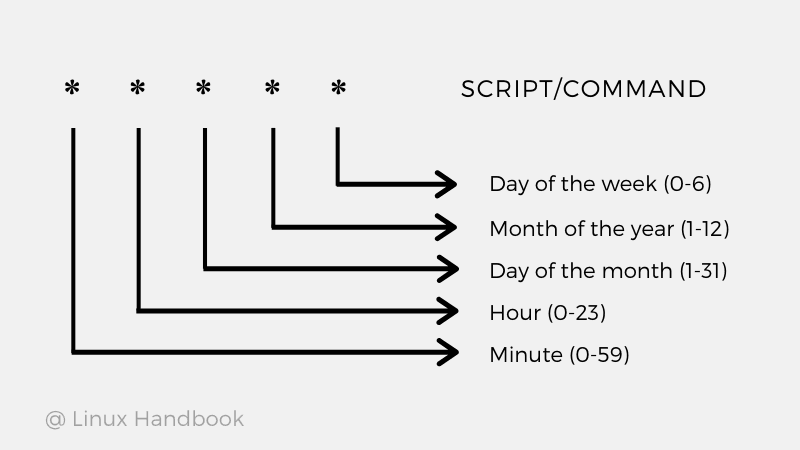
echo "nameserver 1.2.3.4" | sudo tee /etc/resolv.conf

## Naplánované úlohy

Využívá se když chceme spouštět comandy podle rozvrhu na minuty přesně. Crontab stands for „conr table“ protže používá cron plánovač úloh.

crontab [-u {username}] -l … pro vylogování aktuálního stavu crontabu (můžem i podle uživatele)

conrtab -e … pro editaci conrtabu



### Příklad použití

* 00 11, 16 \* \* \* /home/maverick/bin/incremental-backup … 00 = 0th Minute (Top of the hour) 11, 16 = 11 AM and 4 PM \* = Every day \* = Every month \* = Every day
* 00 09-18 \* \* 1-5 /home/maverick/bin/check-db-status … 00 = 0th Minute (Top of the hour) 09-18 = 9 am, 10 am, 11 am, 12 am, 1 pm, 2 pm, 3 pm, 4 pm, 5 pm, 6 pm \* = Every day \* = Every month 1-5 = Mon, Tue, Wed, Thu and Fri (Every Weekday)
* \*/10 \* \* \* \* /home/maverick/check-disk-space … každých 10 minut

### Speciální slova

Keyword Equivalent

* @yearly 0 0 1 1 \*
* @daily 0 0 \* \* \*
* @hourly 0 \* \* \* \*
* @reboot Run at startup.

## Logování do souboru

1. Tee command splits the output of a command so that it can be seen on the display and also be saved in a file.

command | tee log.txt

1. The script command makes a typescript(copy) of everything printed on your terminal:

script -a log.txt

1. This will save output of result into file

Command > log.txt

1. Syslog: Syslog je standardní systémový logovací démon v Linuxu. Můžete použít syslog pro záznam událostí z různých služeb a aplikací do souborů. Syslog konfigurace se nachází v souboru **/etc/syslog.conf** nebo **/etc/rsyslog.conf**, v závislosti na distribuci. Lze nastavit různé úrovně logování (např. debug, info, warning, error) pro různé zařízení nebo služby.
2. Journalctl: Journalctl je nástroj pro prohlížení záznamů v systemd journalu. Systemd je moderní systémový init systém používaný ve většině moderních distribucí Linuxu. Journalctl umožňuje filtrovat a procházet logy podle různých kritérií. Například můžete použít příkaz **journalctl -f** k zobrazování nových událostí v reálném čase.
3. Aplikační logy: Mnoho aplikací a služeb v Linuxu zaznamenává své vlastní logy. Tyto logy se obvykle nacházejí v různých umístěních, jako je **/var/log/**, a mají specifické názvy související s danou aplikací. Například log Apache HTTP serveru se nachází v souboru **/var/log/apache2/error.log**.