Správa a zabezpečení Linuxového serveru

Ondřej Caletka

cesnet

10. září 2019

Obsah

cesnet

Organizace

9:30	instalace OS, práce s disky
10:30	*přestávka*
11:00	základy systemd, uživatelé a práva
12:30	*oběd*
13:30	síť, OpenSSH
14:30	*přestávka*
15:00	OpenSSH, firewall
16:00	*závěr*
16:30	konec

cesnet

Instalace OS a rozdělení disku

cesnet

Tradiční start systému (legacy boot)

- firmware emuluje chování původního IBM PC
- BIOS načte do paměti první sektor z disku a spustí ho
- rozdělení disku na max. 4 oddíly, max. velikost disku 2 TiB
- vícestupňový bootloader GRUB
  1. stage1 v MBR, načte z pevné pozice stage1.5
  2. stage1.5 rozumí konkrétnímu souborovému systému a načte soubor `/boot/grub/stage2`
  3. stage2 je vlastní zavaděč, který načte a spustí jádro

cesnet

Start v režimu UEFI

- **bez emulací 30+ let starého IBM PC**
- firmware podporuje rozdělení disku pomocí MBR a GPT
- firmware automaticky připojí *EFI System Partition* typu FAT32
- podle konfigurace *EFI variables* je spuštěn konkrétní zavaděč
  - EFI proměnné lze měnit jen při startu v EFI režimu
  - na vyměnitelných médiích se spouští `\EFI\BOOT\BOOTX64.EFI`
- lze zavést přímo jádro Linuxu
- častěji se používá GRUB jako mezivrstva
- možnost použít UEFI secure boot

cesnet

Startujeme v UEFI režimu

- podporované drtivou většinou moderních distribucí – pozná se podle adresáře `EFI` v kořeni instalačního média
- **není třeba vyrábět bootovatelný flash-disk pomocí speciálních nástrojů** – stačí překopírovat soubory na FAT32 flash disk
- na Windows-ready HW bude možná nutné vypnout secure boot
  - některé distribuce používají Microsoftem podepsaný *shim*, který ověřuje podpisy podle klíčů distribuce
- většina instalačních médií podporuje zároveň legacy i UEFI, je třeba správně vybrat
  - EFI režim se pozná za běhu podle `/sys/firmware/efi`

cesnet

Otázky k rozdělení disku

tabulka oddílů	GPT
EFI System Partition	FAT32, 512 MB
swap	alokovat, případně nepoužít

Způsob uložení dat

všechno v jednom oddílu	jednoduché, vhodné pro běžné užití
samostaný oddíl pro data	lepší varianta
mnoho samostaných oddílů	problém s fragmentací volného místa
pokročilý souborový systém	možné dělit za provozu

cesnet

Příklad konzervativního rozdělení disku

- Tabulka oddílů GPT
- EFI System Partition FAT32 512 MB
- Swap?
- kořenový svazek ext4
- Fyzický svazek LVM
  - /var ext4
  - LUKS kontejner
    - /home ext4

cesnet

Instalace OS

xkcd 910 © Randall Munroe, překlad Robert Krátký, CC-BY-NC

cesnet

Pokročilejší práce s disky

cesnet

Vrstvy diskového subsystému

- multipath (vynecháme)
- RAID (nakousneme)
- LVM
- LUKS
- souborový systém
- pokročilý souborový systém (Btrfs, ZFS) – kombinuje více vrstev
- šifrování na úrovni souborů

cesnet

Zarovnání oddílů

- disky mají logické sektory délky 512 B
- moderní disky ale mají často ve skutečnosti větší sektory
- požadavek na přenos sektoru 2, 3, 4 a 5 je mnohem pomalejší než požadavek na sektory 0, 1, 2 a 3
- u flash a SSD disku navíc hraje roli velikost *erase bloku* která je klidně 128 KiB
- je třeba zahodit přístup Válec/Hlava/Sektor a místo toho používat logické adresy bloků
- oddíly musí začínat na *kulatém* násobku bloku (typicky 8192)

cesnet

RAID

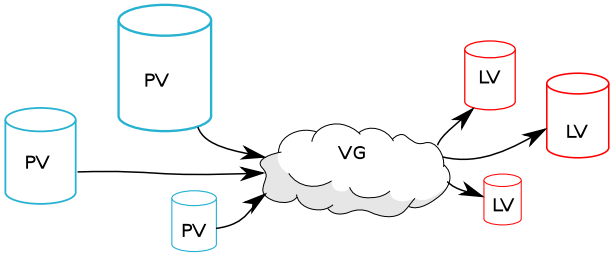
- mdraid – tradiční implementace softwarového RAIDu
  - ovládá se utilitou `mdadm`
  - stav se čte v `/proc/mdstat`
  - různé verze metadat: 0 (na konci, omezení), 1 (na začátku, méně omezení)
- dmraid – implementace pro FakeRAID řadiče v osobních počítačích
  - vhodné pouze pro interoperabilitu s Windows či jinými OS
  - nedoporučováno pro jiná použití
- plnohodnotný HW RAID (např. Dell PERC)
  - pro OS vystupuje pouze jako jeden logický disk
  - baterie pro uchování cache při náhlém výpadku proudu

Nedohledovaný RAID = žádný RAID!

cesnet

Logical Volume Management

- správa logických svazků
- virtualizace blokových zařízení
- implementováno pomocí Device Mapper
- místo disků používáme logické svazky (LV).
- tyto svazky alokujeme ve skupině svazků (VG).
- poskytovatelem fyzického prostoru pro VG jsou fyzické svazky (PV).



cesnet

Vlastnosti LVM

- Výhody:
  - snadná změna velikosti a počtu oddílů.
  - nezávislost na připojovacím rozhraní – disky se indentifikují pomocí UUID.
  - snapshoty – možnost kdykoli získat „hardcopy“ daného LV, zmrazenou v čase.
- Nevýhody:
  - obvykle nezajišťuje redundanci dat.
  - přidává další složitost – problémy se záchranou dat, instrukcí TRIM a root oddílem na LVM.

cesnet

Práce s LVM

- inicializace PV – oddíl s $id = 0x8e$:
  `pvcreate /dev/sdc1`
- inicializace VG – volba velikosti bloku (PE):
  `vgcreate MyVG /dev/sd[cde]1`
- vytvoření LV:
  `lvcreate -n padesatmb -L 50m MyVG`
- zjišťování informací:
  `{pv,vg,lv}s`
  `{pv,vg,lv}display`
  `{pv,vg,lv}display -m` – vypíše mapování PE

cesnet

Správa LVM

- změna velikosti LV:
  `lvrezize -L +10m /dev/MyVG/padesatmb`
  přepínač `-r` provede zároveň změnu velikosti filesystému.
- přesouvání dat mezi PV (za běhu):
  `pvmove -vi 5 /dev/md1 /dev/sdc3`
- de/aktivace – potřeba provést při startu: `vgchange -a [y/n]`

cesnet

Snapshoty

- vytvoření snapshotu:
  `lvcreate -s -L 30m -n snap1 /dev/MyVG/padesatmb`
- velikost může být menší, než velikost původního oddílu, zapisují se jen zmeny.
- snapshot je zapisovatelný.
- snapshotu můžeme měnit velikost prostoru vyhrazeného pro změny.
- snapshot zrušíme stejně jako LV:
  `lvremove /dev/MyVG/snap1`

cesnet

Šifrování disku

- fyzická bezpečnost dat **vypnutého** počítače
- nesupluje šifrování kritických uživatelských dat, např. **privátních klíčů, hesel**
- bez výrazného vlivu na výkon
- na úrovni souborů nebo blokového zařízení
- pouze cenných dat (`/home`) nebo celého disku

cesnet

Scénáře útoku

- **odcizení zařízení**, servisní zásah
  - stačí obyčejné šifrování důležitých souborů
- evil-maid útok
  - úprava nešifrované části systému
  - vyzradí heslo při přístím zadání
- cold-boot útok
  - nelze eliminovat na straně OS

cesnet

Šifrování celého disku

- lepší odolnost proti evil-maid
  - menší plocha viditelná útočníkovi
  - i bez MAC je problematický útok na zašifrovaná data
- vyžaduje zadání hesla při startu
  - nelze nastartovat vzdáleně a odemknout přes síť

cesnet

Tradiční šifrování celého disku

- nešifrovaný oddíl `/boot`, případně ESP
- obsahuje zavaděč, jádro a initramfs
- initramfs se během startu zeptá na heslo a připojí ostatní disky
- chceme-li Secure boot, měl by zavaděč ověřovat podpisy jádra a initramfs

cesnet

Jak nepřijít o výkon

cryptsetup benchmark

```
# Testy jsou počítány jen z práce s pamětí (žádné I/O úložiště).
PBKDF2-sha1      1394382 iterations per second for 256-bit key
PBKDF2-sha256    1635843 iterations per second for 256-bit key
PBKDF2-sha512    1339177 iterations per second for 256-bit key
PBKDF2-ripemd160 1018034 iterations per second for 256-bit key
PBKDF2-whirlpool  777875 iterations per second for 256-bit key
#     Algorithm | Key |  Encryption |  Decryption
        aes-cbc   128b  1112,2 MiB/s  3501,2 MiB/s
    serpent-cbc   128b    93,5 MiB/s   713,3 MiB/s
    twofish-cbc   128b   212,7 MiB/s   385,7 MiB/s
        aes-cbc   256b   840,4 MiB/s  2788,0 MiB/s
        aes-xts   256b  2558,9 MiB/s  2560,5 MiB/s
        aes-xts   512b  2207,0 MiB/s  2223,6 MiB/s
```

cesnet

Je potřeba mít zavedený modul `aesni_intel`

cryptsetup benchmark bez podpory AES-NI

```
#     Algorithm | Key |  Encryption |  Decryption
        aes-cbc   128b   277,9 MiB/s   321,9 MiB/s
        aes-cbc   256b   215,8 MiB/s   241,1 MiB/s
        aes-xts   256b   327,4 MiB/s   325,0 MiB/s
        aes-xts   512b   245,2 MiB/s   243,0 MiB/s
```

Rychlost NVMe SSD

```
# hdparm -Tt --direct /dev/nvme0n1
/dev/nvme0n1:
Timing O_DIRECT cached reads: 2758 MB in  2.00 seconds = 1381.80 MB/sec
Timing O_DIRECT   disk reads: 4460 MB in  3.00 seconds = 1486.59 MB/sec
```

cesnet

Btrfs

- pokročilý souborový systém – podpora RAID, subvolumů a snapshotů
- dokáže nahradit RAID i LVM
- nepodporuje swap soubory
- nepodporuje šifrování
- stále je prohlašován za vývojový

cesnet

Btrfs subvolumes

- nezávislé části filesystému Btrfs
- samostatně se snapshotují
- dobrá praxe je mít kořenový systém souborů v subvolume
- plochý nebo zanořený model

  plochý všechny subvolumes jsou v kořeni, následně jsou připojeny pomocí vícenásobného záznamu v `/etc/fstab`

  zanořený subvolumes jsou umístěny přímo na svém místě, jsou připojeny automaticky s připojením rodiče
- vhodné pro logické oddělení uživatelských, systémových a dočasných dat

cesnet

Nástroje pro práci s diskovými oddíly

- `lsblk`
- `blkid`
- `fdisk`

cesnet

Zvětšení disku virtuálního počítače

1. zvětšit disk v hypervizoru
2. pomocí fdisk smazat a znovu vytvořit oddíl **od stejného místa**
3. restartovat, aby se změna načetla
4. provést online změnu souborového systému na velikost oddílu

cesnet

Základy systemd

cesnet

systemd

- moderní náhrada spouštěcích shell skriptů
- inspirováno spouštěcím systémem launchd z macOS
- spousta (i oprávněných) negativních emocí, adoptováno mlčící většinou
- jednodušší správa závislostí, rychlejší start systému
- mnoho malých utilit, např. `udev`

cesnet

udev

- dynamický správce speciálních souborů v `/dev`
- poslouchá události jádra
- vytváří a maže speciální soubory
- zajišťuje perzistenci názvů zařízení (včetně síťových karet)
- nastavuje oprávnění, včetně speciálních práv pro uživatele na konzoli (s pomocí `ConsoleKit`)
- standardní pravidla v `/lib/udev/rules.d/`
- přepis pravidel administrátorem v `/etc/udev/rules.d/`

cesnet

Spouštěcí systém systemd

- ovládání utilitou `systemctl`
- standardní pravidla v `/lib/systemd/system`
- přepis pravidel administrátorem v `/etc/systemd/system/`
- INI-soubory popisující *jednotky* (služby, sokety, cíle, časovače,...)
- nové režimy spouštění:
  - aktivace soketem (něco jako inetd)
  - umístění procesu ve vlastní control group (`systemd-cgls`)
  - podpora pro jednoduché služby

cesnet

Základní ovládání

`systemctl`	vypíše jednotky a jejich stav
`systemctl status`	vypíše stav systému
`systemctl start <unit>`	spustí jednotku
`systemctl stop <unit>`	zastaví jednotku
`systemctl enable <unit>`	povolí jednotku, aby se automaticky spouštěla
`systemctl disable <unit>`	zakáže automatické spouštění
`systemctl mask <unit>`	zneviditelní službu, nepůjde pustit ani ručně

cesnet

Editace jednotek

`systemctl cat <unit>`	vypíše konfiguraci aktuální jednotky
`systemctl show <unit>`	vypíše nízkoúrovňové detaily jednotky
`systemctl edit <unit>`	otevře editor, kam je možné připsat změny proti běžné konfiguraci
`systemctl edit --full <unit>`	překopíruje soubor jednotky do `/etc`, otevře editor
`systemctl edit --runtime …`	překopíruje soubor jednotky do `/run`, změny se po restartu ztratí
`systemctl daemon-reload`	vynutí nové načtení jednotek po jejich změně

cesnet

Nemazání obrazovky po startu

```
# systemctl cat getty@tty1
…
[Service]
# the VT is cleared by TTYVTDisallocate
…
TTYVTDisallocate=yes
…

# systemctl edit getty@tty1
[Service]
TTYVTDisallocate=no
```

cesnet

Aktivace soketem

listening socket `socket() -> bind() -> listen()`

connected socket `accept(<listening socket>)`

- velmi vhodné pro síťové servery
- eliminuje složité řešení závislostí mezi službami – služba je dostupná, i když (ještě) neběží
- jednotka `<unit>.socket` vytvoří *listening socket*
- při příchozím spojení je spuštěna `<unit>.service`, *listening socket* je předán službě na `fd=3`
- alternativně je služba spuštěna při každém spojení a je jí předán přímo *connected socket* na `fd=0` – režim emulace `inetd`
- služba musí podporovat předávání soketů

cesnet

Instancované služby

- řešení pro služby, které je třeba spustit víckrát zároveň
- jednotka ve tvaru `getty@tty1.service` definována v souboru `getty@.service`
- zástupný znak `%I`/`%i` reprezentuje konkrétní instanci (čitelně, resp. escapovaně)
- jednotlivé instance lze případně upravit podle potřeby
- *generátory* jsou malé skripty, které na základě externí konfigurace vygenerují závislosti pro správné instance (například `openvpn.service` závisí na `openvpn@vpn1.service`)

cesnet

systemd-journald

- vylepšený logovací nástroj
- sbírá logy z `/dev/log`, jádra, výstupu spuštěných služeb systemd
- ukládá v binárním formátu, doplněné o metadata, s podporou bezproblémové rotace
- umožňuje připojit další *syslogd* do `/run/systemd/journal/syslog`
- standardně se ukládá neperzistentně do `/run/log/journal`
- nepodporuje logování přes síť

cesnet

journalctl v příkladech

`journalctl`	vypíše log, od počátku archivu
`journalctl -e`	vypíše log, ukáže hned konec
`journalctl -f -u <unit>`	sleduje výpisy konkrétní jednotky
`journalctl --since today`	ukáže log ode dneška
`journalctl /dev/sda1`	ukáže log týkající se zařízení

cesnet

Uživatelé a práva

cesnet

Stručně o uživatelích

- `adduser` vs. `useradd`
- skupina pro uživatele vs. skupina `users`
- editace souborů pomocí `vipw`, a `vigr`
- expirace účtů pomocí `chage`
- členství ve skupinách `gpasswd`

cesnet

Hashovaná hesla

- jsou uložena v `/etc/shadow` spolu s informacemi o platnosti
- používá se opakované SHA-512 (standardně 5000×)
- podpora `bcrypt` a `argon` se stále jen chystá

cesnet

Pluggable Authentication Modules

- koncept konfigurovatelných zásuvných modulů pro autentizaci
- konfigurace v `/etc/pam.d/<jméno služby>`
- možnost globálně ovlivnit způsob autentizace

Čtyři oblasti

account	správa účtu (existuje účet a je aktivní)
auth	autentizace uživatele (ověření hesla)
password	aktualizace autentizace (změna hesla)
session	činnosti před zahájením a po ukončení relace

cesnet

Zajímavé moduly PAM

`pam_unix`	ověřování unixových účtů a hesel, logování přihlášení
`pam_env`	nastavení proměnných prostředí po přihlášení
`pam_motd`	zobrazí Message of the Day
`pam_mail`	zobrazí informace o elektronické poště
`pam_limits`	nastaví limity procesů pro relaci
`pam_shells`	brání v příhlášení uživatelům, kteří nemají platný shell
`pam_wheel`	blokuje nečleny skupiny `wheel` od používání `su`/`sudo`
`pam_systemd`	informuje `systemd` o relaci; zabrání vytuhnutí SSH při restartu
`pam_tally2`	zablokuje účet po určitém počtu neúspěšných pokusů

cesnet

Stručně o právech

- čtyři číslice osmičkové soustavy – každá 3 bity s váhami (4, 2, 1)
- suid(*nnn), uživatel (n*nn), skupina (nn*n), ostatní (nnn*)
- čtení (r:4), zápis (w:2), vykonání (x:1)
- adresář bez práva `x` nelze otevřít
- první číslice: SUID(4), SGID(2), sticky(1)
- změna utilitou `chmod`
- hodnota `umask` maskuje oprávnění nově vytvořených souborů

cesnet

SUID, SGID

- binárka se spustí s právy vlastníka souboru
- častý způsob získání oprávnění `root`
- nefunguje s shell skripty
- SGID na adresáři vynutí použití stejné skupiny na vnořené soubory

Sticky/Restricted deletion

- na souborech nemá význam
- na adresářích brání mazání souborů jiným uživatelem než vlastníkem (typicky pro `/tmp`)

cesnet

Sdílený pracovní adresář

```
# addgroup workgroup
# mkdir /home/shared
# chown :workgroup /home/shared
# chmod g+ws /home/shared
# gpassword -a user1 workgroup
# gpassword -a user2 workgroup
# echo "umask 0007" > /etc/profile.d/umask.sh
```

cesnet

ACL

- pro případy, kdy je dělení uživatel/skupina/ostatní příliš hrubé
- ovládání utilitami `getfacl` a `setfacl`
- signalizace pomocí + ve výpisu `ls -l`

Capabilities

možnost získat jistá privilegia bez nutnosti běžet pod uživatelem `root`

```
$ getcap /bin/ping
/bin/ping = cap_net_raw+ep
```

cesnet

Rozšířené atributy

```
$ wget https://www.nebezi.cz/ --xattr
…
$ getfattr -d index.html
# file: index.html
user.xdg.origin.url="https://www.nebezi.cz/"
```

cesnet

Konfigurace sítě

cesnet

Perzistetní názvy síťových karet

- jádro přiděluje názvy `eth*` v pořadí detekce jednotlivých karet
- většina uživatelů nemá problém, protože:
  - mají pouze jednu kartu daného druhu, nebo
  - karty jsou natolik rozdílné, že je pořadí detekce neměnné
- v případě souběhu detekce více karet může docházet ke změnám názvů karet po každém restartu
- udev to řeší algoritmickým pojmenováním síťových karet – vyžaduje rozumně funkční firmware
- lze napsat vlastní pravidla; ta by ale měla přejmenovávat na zaručeně neexistující názvy (tedy ne `eth*`)
- lze to vypnout buď volbou jádra `net.ifnames=0` nebo prázdným souborem `/etc/udev/rules.d/80-net-name-slot.rules`

cesnet

Konfigurace sítě

- zastaralé nástroje: `ifconfig, route, netstat, brctl`
- moderní nástroje: `ip, ip route, ss, ip link`
- všechny změny jsou neperzistentní; zajištění perzistence se liší podle distribuce
- neexistuje žádný *reset* síťového stacku

cesnet

iproute2

ip link	vytváření a editace síťových rozhraní
ip address	nastavování IP adres
ip route	práce se směrovacími tabulkami
ip rule	práce pravidly směrování (policy based routing)
ss	statistika otevřených soketů
bridge	nastavení mostů
tc	nastavení front

cesnet

Příklad ruční konfigurace sítě

```
# ip link set dev eth0 up
# ip addr add dev eth0 192.168.1.2/24
# ip rou add default via 192.168.1.1
# echo "nameserver 1.1.1.1" >/etc/resolv.conf
```

cesnet

/etc/resolv.conf

nameserver <IP>	adresa serveru (max. 3×)
domain <d>	místní doména
search <d> <d>...	prohledávací seznam
options rotate	náhodně měnit použitý DNS server
options edns0	používat EDNS0 (např. DNSSEC)

Soubor může být spravován Network Managerem, utilitou `resolvconf` nebo jinak.

cesnet

systemd-resolved

- userspace implementace stub resolveru
- dbus, glibc a DNS API
- resolvování lokálních jmen z `/etc/hosts`, jména `_gateway`
- jména bez tečky jsou resolvována pomocí LLMNR
- je možné směrovat na různé DNS servery v závislosti na dotazovaném jménu
- podpora validace DNSSEC

cesnet

Konfigurace IPv6

- minimální implementace vestavěná v jádře
- dnes často vypnutá ve prospěch komplexnější implementace v userspace
- automatické nastavení směrování podle ohlášení směrovačů
- automatické nastavení IP adres pomocí SLAAC
- vyčištění pomocí `ip -6 addr flush dev eth0 scope global`
- konfigurace pomocí voleb `sysctl`:

`net.ipv6.conf.eth0.accept_ra` povolí zpracování RA

`net.ipv6.conf.eth0.autoconf` povolí automatickou konfiguraci adres (=2 i v režimu směrovače)

cesnet

Základní utility

ping	posílá ICMP echo-request
traceroute	hledá cestu pomocí UDP, TCP, nebo ICMP
mtr	lepší traceroute
arping	objevuje stanice na segmentu pomocí ARP
host	provádí DNS dotazy
iftop	vizualizuje toky na rozhraní
tcpdump	zaznamenává a analyzuje obsah přenášených zpráv

cesnet

Připojení k více sítím zároveň

- pro komunikaci se sousedy není problém
- nelze mít víc výchozích bran
- nelze mít víc DNS serverů

Jednoduché řešení pomocí `ip route from`

```
# ip link set dev eth1 up
# ip addr add dev eth1 172.17.1.2/24
# ip rout add default from 172.17.1.2 via 172.17.1.1
```

cesnet

Policy routing

- lze vytvořit víc směrovacích tabulek
- pravidla `ip rule` vybírají, která tabulka bude použita
- tabulky lze pojmenovat čísly nebo v `/etc/iproute2/rt_tables`

Příklad policy routingu

```
# ip link set dev eth1 up
# ip addr add dev eth1 172.17.1.2/24
# ip rou add default via 172.17.1.1 table 123
# ip rule add from 172.17.1.2 lookup 123
```

cesnet

Network namespaces

- stavební prvek linuxových kontejnerů
- lze použít samostatně
- každá síťová karta je právě v jednom NS

Přestěhování `eth0` do vlastního NS

```
# ip netns add testovani
# ip link set eth0 netns testovani
# ip netns exec testovani bash
…
# ip netns delete testovani
```

cesnet

Na co se často zapomíná

- směrování je ve výchozím stavu vypnuto
- pro úspěšné spojení musí existovat cesta tam i zpět
- privátní adresy na internet nepatří
- nasměrované, ale nepoužívané adresy zahazujeme

```
# sysctl net.ipv4.ip_forward=1
# ip route add unreachable 192.168.0.0/16
# ip -6 route add unreachable 2001:db8::/32
```

cesnet

Pokročilé použití OpenSSH

cesnet

OpenSSH

- svobodná implementace protokolu SSH od tvůrců OpenBSD
- nejde zdaleka jen o *šifrovaný telnet*
- standardizováno v IETF

cesnet

Běžné přihlášení

1. server drží privátní klíče od každého algoritmu
2. klient při prvním přihlášení potvrdí pravost otisku veřejného klíče serveru. Vazbu adresy a klíče si uloží klient v `~/.ssh/known_host`
3. uživatel se představí heslem
4. spustí se shell

cesnet

Jednoduchá vylepšení

- uživatelská konfigurace v souboru `~/.ssh/config`
- RandomArt obrázek `VisualHostKey yes`
- udržení spojení `ServerAliveInterval 10`
- ukončení mrtvého spojení pomocí `Enter ~ .`
- napovídání jmen z `known_hosts` pomocí `bash-completion` vyžaduje nehashovaná jména serverů: `HashKnownHosts no`

cesnet

Sdílené spojení

- multiplexování více nezávislých relací jedním SSH spojením
- realizováno pomocí řídicího soketu
  - *master* naváže spojení a vytvoří UNIX soket
  - *slave* se komunikuje soketem.
- autentizaci provádí pouze *master*

Konfigurace sdílení spojení

```
ControlMaster auto
ControlPath ~/.ssh/controlsock-%h-%p-%r
ControlPersist  30
```

cesnet

Použití uživatelského klíče

- vygenerujeme klíč pomocí `ssh-keygen`
- dobrá praxe vyplnit smysluplný komentář `-C oskar@latte`
- klíč chráníme silným heslem
- různé algoritmy na výběr
  - DSA – 1024bit, deprecated, nepodporováno od verze 7.0 (2015)
  - RSA – volitelná délka, bezpečná výchozí 2048 bitů
  - ECDSA – 256, 384, nebo 521 bitů, k dispozici od 5.7 (2011)
  - Ed25519 – 256 bitů, k dispozici od 6.4 (2014)
- pro maximální kompatibilitu s ne-OpenSSH implementacemi jedině RSA
- kopírování na server ručně nebo pomocí `ssh-copy-id`

cesnet

Volby v souboru authorized_keys

`from="2001:db8::1,192.0.2.1"`	omezení IP adresy
`no-port-forwarding`	zakáže tunelování
`restrict`	vypne všechny tunely, pty, atd. včetně případných budoucích funkcí
`environment="NAME=value"`	nastaví proměnné prostředí (např. `GIT_AUTHOR_NAME`)
`command="command"`	vynutí spuštění konkrétního příkazu

cesnet

SSH agent

- klíčenka s privátními klíči uživatele, nepustí privátní klíč
- ad-hoc spuštění pomocí `ssh-agent bash`
- přidání klíče pomocí `ssh-add -c` vynutí vyžádání potvrzení před každým vystavením podpisu
- automatické přidání při prvním použití volbou `AddKeysToAgent confirm`
- lze omezit dobu života klíče v klíčence pro každý klíč i pro agenta: `-t8h`
- agenta je možné tunelovat pomocí `ssh -A` nebo volby `ForwardAgent yes`
  - (super-)uživatel vzdáleného systému **má ke klíčence přístup**!
  - lépe omezit jen na důvěryhodné servery a/nebo vyžadovat potvrzení před použitím klíče

cesnet

OpenSSH certifikáty

- možnost delegovat ověření klienta nebo serveru na certifikační autoritu
- nejde o X.509, CA je obyčejný SSH klíč
- certifikát v samostatném souboru `<jméno klíče>-cert.pub`
- použití například jako:

  osobní CA pro všechny klíče uživatele – není třeba pravidelně aktualizovat `authorized_keys`

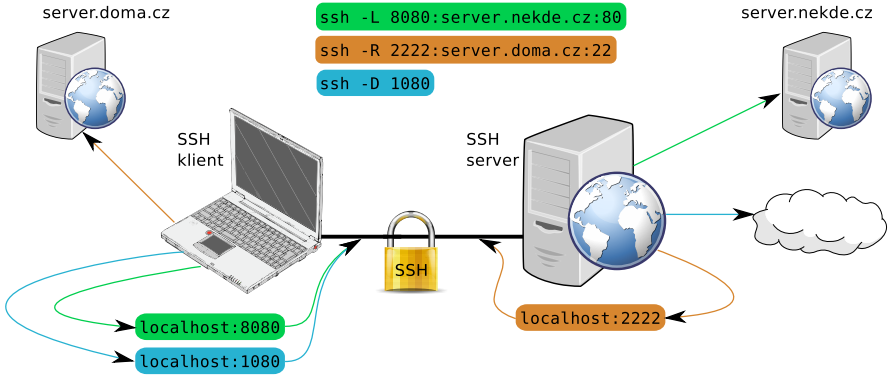
  serverová CA automatická důvěra veřejným klíčům serverů

  centrální uživatelská CA v certifikátu je uvedené jméno uživatele, v konfiguraci sshd je globální volba `TrustedUserCAKeys`
- revokace pouze distribucí revokačních seznamů

cesnet

Tunelování

- statické i dynamické (SOCKS v4 / SOCKS v5)
- nejde o TCP-in-TCP ale tři nezávislá TCP spojení
- na poslouchací straně omezeno na localhost



cesnet

Režim netcat

- přepínač `-W <host>:<port>` spojí *stdio* klienta k TCP spojení na straně serveru
- hodí se na tunelování SSH spojení SSH spojením.
- výborně se kombinuje s volbou `ProxyCommand`

~/.ssh/config

```
Host server-behind-firewall
    ProxyCommand ssh -W 10.0.0.1:22 firewall.nekde.cz
```

cesnet

Přímá podpora `ProxyJump`

- k dispozici od OpenSSH 7.3
- automaticky naváže tunely skrz vyjmenované *jump hosts*
  `-J [<user>]@<host>[:<port>],…`
- spojení se po cestě nerozšifrovává
- vylučuje se s volbou `ProxyCommand`

~/.ssh/config

```
Host *.mgmt.cesnet.cz
    ProxyJump oskar@oskarpc.cesnet.cz
```

cesnet

Serverové klíče

- stejné jako uživatelské klíče
- vygenerované při instalaci, nebo prvním startu
- nejsou chráněny heslem, je možné použít SSH agenta
- jeden[1] pro každý algoritmus
- obtížné rolování
- volba klienta `UpdateHostKeys yes` pro automatickou aktualizaci klíčů (od verze 6.8)

[1] nebo i více, ale používá se vždy první

cesnet

Ověření serveru pomocí DNSSEC

- vygenerujeme otisk **z klíče serveru na disku** pomocí `ssh-keygen -r <owner>`
- klientovi nastavíme volbu `VerifyHostKeyDNS <yes|ask>`
- výhoda – klíč je možné bezešvě rolovat

Příklad

```
The authenticity of host 'server.example.com (192.0.2.1)'
can't be established. RSA key fingerprint is
aa:55:cc:9c:a5:c6:1b:f1:a5:d2:be:eb:7e:1c:53:05.
Matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)?
```

cesnet

OpenSSH HPN patche

- OpenSSH není optimalizováno na výkon
- pomalé při vyšší latenci kvůli malé a fixní velikosti bufferů
- velké buffery zase zabíjejí interaktivitu (*Bufferbloat*)
- problém řeší sada patchů označených jako HPN
  - dynamická velikost bufferu podle TCP okna
  - možnost nulového šifrování
  - možnost paralelizace některých procesů

cesnet

Mosh – Mobile Shell

- zcela nový protokol pro příjemnější terminálové sezení na nekvalitní lince
- využívá UDP zprávy a inteligentní lokální odezvu
- autentizace pomocí SSH, žádný nový démon
- usnutí a probuzení, či změna IP adresy klienta za běhu
- synchronizuje pouze stav obrazovky
- implementuje pouze UTF-8 terminál
- nedokáže nic tunelovat
- nedokáže roamovat mezi IP adresami serveru

cesnet

Shrnutí OpenSSH

- chraňte osobní klíče heslem – **i na šifrovaném disku**
- používejte agenta, ale vyžadujte potvrzení každého použití klíče
- nemažte soubor `known_hosts`!
  Lepší je `ssh-keygen -R <jméno serveru>`
- zapněte ověřování SSHFP záznamů v DNS
- vyzkoušejte: `ssh whoami.filippo.io`
- více informací: Root.cz: Pokročilé vlastnosti OpenSSH

cesnet

Linuxový firewall – iptables a nftables

cesnet

Linuxový firewall – netfilter

- možnost ovlivnit data během jejich průchodu
- záchytné body tvoří základní řetězy (`iptables`) nebo místa k zavěšení vlastních řetězů (`nftables`)

záchytné body

`PREROUTING`	provoz těsně po příchodu ze sítě
`INPUT`	příchozí provoz určený lokálnímu procesu
`FORWARD`	příchozí provoz určený k odchodu jinam
`OUTPUT`	odchozí provoz z lokálních procesů
`POSTROUTING`	provoz těsně před odchodem do sítě

cesnet

Tabulky, řetězy, pravidla

tabulky kontejnery pro řetězy – v `iptables` pevně dané: `filter`, `nat`, `mangle`

řetězy kontejnery pro pravidla – v `iptables` vestavěné a vlastní, v `nftables` jen vlastní, typu `filter`, `nat`, nebo `route`

pravidla procházejí se postupně, každé má počítadlo a nejvýše jeden cíl (v `iptables`)

cesnet

iptables(-legacy)

- tradiční rozhraní netfiltru
- rigidní struktura tabulek a řetězců
- samostatná utilita pro každý protokol – `ip`, `ip6`, `arp`, `eb`
- rozšíření pomocí jaderných modulů a userspace knihoven

nftables

- moderní rozhraní netfilteru
- flexibilní pravidla, prázdný výchozí stav
- implementace v jádře používající virtuální stroj
- rozšiřitelnost čistě pomocí userspace
- nástroje `iptables-nft` pro snadný přechod

Kontrola stavu firewallu

```
# iptables --list -v
# iptables -t nat --list -v
# iptables-save
# nft list ruleset -a
```

Vyčištění firewallu

```
# iptables --policy INPUT ACCEPT
# iptables --flush
# iptables --delete-chain
# nft flush ruleset
```

cesnet

Vytváření pravidel

```
# iptables --add INPUT [matchers] [--jump <target>]
```

Cíle (a verdikty) (část)

`ACCEPT`	přijmi
`DROP`	zahoď
`REJECT`	odmítni
`<jméno>`	skoč do příslušného řetězu
`RETURN`	návrat z předchozího řetězu
`LOG`	ulož do logu

cesnet

Základní matchery

`-i` vstupní rozhraní

`-o` výstupní rozhraní

`-s` zdrojová IP adresa

`-d` cílová IP adresa

`-p` protokol transportní vrstvy

`-m` rozšiřující modul

`--sport` zdrojový port (pro tcp, udp)

`--dport` cílový port (pro tcp, udp)

cesnet

Connection tracking

- Linux sleduje všechna procházející TCP, UDP a ICMP spojení
- data lze použít pro jednoduchý stavový firewall nebo NAT
- sledovaná spojení lze vyčíst z `/proc/net/nf_conntrack`
- na zatížených serverech to lze vypnout a ušetřit prostředky

Možné stavy spojení

`NEW`	paket zahajuje nové spojení
`ESTABLISHED`	paket patří ke známému spojení
`RELATED`	paket se vztahem k existujícímu spojení
`INVALID`	o stavu nejsou informace a nejde o nové spojení
`UNTRACKED`	sledování stavu bylo vypnuto

Jednoduchý firewall s iptables

```
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -4 -m limit --limit 10/s -j ACCEPT
-A INPUT -p ipv6-icmp -6 -m limit --limit 10/s -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A INPUT -p udp --dport 33434:33499 -j REJECT
-P INPUT DROP
```

Pozn.: Nutno načíst dvakrát, pomocí `iptables-restore` a `ip6tables-restore`

cesnet

Jednoduchý firewall s nftables

```
table inet firewall {
  chain input {
    type filter hook input priority 0; policy drop;
    iifname "lo" accept
    ct state established,related accept
    meta l4proto icmp meta nfproto ipv4 \
                      limit rate 10/second accept
    meta l4proto ipv6-icmp meta nfproto ipv6 \
                           limit rate 10/second accept
    tcp dport { ssh, https } accept
    udp dport { 33434-33499 } reject
  }
}
```

Ovládání utility nft

```
# nft add table inet asdf
# nft list table inet asdf -ann
# nft add chain inet asdf ghjk { type filter hook input \
                                          priority 0 \; }
# nft chain inet asdf ghjk { policy drop \; }
# nft add rule inet asdf ghjk tcp dport { ssh, https } \
                                                accept
# nft delete rule inet asdf ghjk handle 5
# nft add rule inet asdf ghjk position 5 ct state \
                                invalid counter drop
```

cesnet

Závěr

cesnet

Děkuji za pozornost

Ondřej Caletka
Ondrej.Caletka@cesnet.cz
https://**Ondřej.Caletka.cz**

Prezentace je již nyní k dispozici ke stažení.