Kurs administrowania systemem Linux

Zajęcia nr 6: Podstawowe czynności administracyjne w Linuksie (2)

Instytut Informatyki Uniwersytetu Wrocławskiego

5 kwietnia 2022

Ważne opcje montowania

- `ro` — read only (uwaga na księgowanie!)
- `discard` — wysyłaj polecenia TRIM do dysku (SSD). Dawniej polecano `fstrim(8)` jako alternatywę.
- `noatime`, `nodiratime` — nie zapisywać czasu ostatniego odczytu (uwaga na programy `mutt` i podobne).
- `user` — wolno montować zwykłemu użytkownikowi (zwykle CDROM, pendrive itp.). Por. UDisks2 i PolicyKit.
- `mode`, `umask`, `fmask`, `dmask` — nadaj odpowiednie prawa dostępu (plikom, katalogom): bardzo przydatne w przypadku FAT (np. `fmask=0177,dmask=0077`).
- `noexec` — zabraniaj uruchamiania programów z tego systemu.
- `nosuid` — zabraniaj uruchamiania programów z SUID.
- `noauto` — nie montować automatycznie w czasie rozruchu (`fstab`)
- `defaults` — w razie braku opcji (`fstab`)

Moje rozwiązanie

- Każdy dysk zewnętrzny, pendrive, karta SD itp. *dysk*, ma własny podkatalog `/media/`*dysk*`/`.
- Domyślne prawa dostępu do `/media/`*dysk*`/` to `000` (zabezpiecza przed omyłkowym pisaniem do katalogu bez zamontowanego urządzenia).
- Dysk *dysk* ma własny wpis w pliku `fstab` bazujący na `UUID` lub `LABEL`, jeśli trzeba, to z opcją `user`.
- Zawsze wiadomo, gdzie dany dysk się znajduje i czy jest zamontowany.
- Jak sprawdzić automatycznie, czy dysk *dysk* jest zamontowany?

Przykład: montowanie /tmp

ręcznie

```
# mount -t tmpfs none /tmp -o size=2000m,mode=1777,strictatime
```

/etc/fstab

```
none /tmp tmpfs size=2000m,mode=1777,strictatime 0 0
```

systemd: jednostka tmp.mount

```
[Unit]
Description=Temporary Directory
ConditionPathIsSymbolicLink=!/tmp
DefaultDependencies=no
Conflicts=umount.target
Before=local-fs.target umount.target
```

```
[Mount]
What=tmpfs
Where=/tmp
Type=tmpfs
Options=mode=1777,size=2000m,strictatime

[Install]
WantedBy=local-fs.target
```

Praca z jądrem obcym

- Zamontowany system plików może być podwiązany w innych punktach montowania.
- Ważne w przypadku `chroot` — proces *chrootowany* potrzebuje dostępu do `/sys`, `/proc` i `/dev`.
- Często przydatne podczas ratowania systemu: jądro systemu ratunkowego użyczone dla *userlandu* systemu ratowanego.

W-chroot-owanie w system ratowany

```
# mkdir /target
# mount /dev/rootfs-ratowanego /target
# for FS in proc sys dev; do \
      mount -o bind /$FS /target/$FS; done
# chroot /target
# jesteśmy w systemie ratowanym
```

Loopback

`losetup(8)`

- Tworzy urządzenie blokowe z obrazu w pliku.
- Opcja `-f`: znajdź wolny numer *n* urządzenia `/dev/loop`*n*.
- Opcja `--show`: wypisz nazwę utworzonego urządzenia.

Montowanie obrazu płyty ISO

```
# mount $(losetup -f --show plyta.iso) /mnt
```

Skrót: opcja `loop` programu `mount`

```
# mount -o loop plyta.iso /mnt
```

Partycje zaszyfrowane

- W Linuksie przeważnie `dm-crypt` z nagłówkiem LUKS.
- Narzędzie: `cryptsetup`.
- Idea: partycja zaszyfrowana jest mapowana na partycję wirtualną w `/dev/mapper/`.
- Konfiguracja: `/etc/crypttab`.

Montowanie systemu zaszyfrowanego

```
# cryptsetup open /dev/dysk nazwa
Pytanie o hasło
# mount /dev/mapper/nazwa /punkt_montażowy
```

Odmontowywanie systemu zaszyfrowanego

```
# umount /punkt_montażowy
# cryptsetup close nazwa
```

Access control

Rodzaje kontroli dostępu

- *Mandatory*: scentralizowana, prawa przydziela aministrator.
- *Discretionary*: rozproszona. Obiekty mają właścicieli. Prawa przydziela właściciel obiektu.
- W Uniksie stosuje się oba modele.

Kontrola dostępu do plików

- Każdy plik ma właściciela i grupę.
- Każdy plik ma *access mode*: 12 flag określających możliwe rodzaje dostępu.
- Niektóre systemy plików wspierają też *extended attributes* i ACL (*Access Control Lists*).

Tryb dostępu

12 bitów zapisywanych zwykle ósemkowo

04000	set user ID
02000	set group ID
01000	sticky bit
00400	read by owner
00200	write by owner
00100	execute program / search directory by owner
00040	read by group
00020	write by group
00010	execute/search by group
00004	read by others
00002	write by others
00001	execute/search by others

Szczególne tryby dostępu

- SETUID — dla plików wykonywalnych.
- SETGID — dla plików wykonywalnych i katalogów (zob. np. grupa `staff` i prawa `drwxrwsr-x` dla `/usr/local/`).
- *Sticky bit* (dla katalogów, dawniej też zwykłych plików).
- Prawa dostępu do dowiązań symbolicznych są ignorowane i nie można ich zmienić za pomocą polecenia `chmod`.
- Faktycznie tryb pliku jest przechowywany w jednym słowie razem z typem pliku.

Tryb dostępu — zapis literowy `rwx`

Podczas ujawniania informacji

- Ciąg 10 symboli ze zbioru `-dlbcprwxsStT`.
- Stosowany przez programy `ls`, `stat` itp.
- Dodatkowo informacja o typie pliku (`-dlbcps`): pierwszy symbol.
- SUID: `s` zamiast `x` w pierwszej trójce (`S` jeśli brak `x`).
- SGID: `s` lub `S` zamiast `x` w drugiej trójce.
- *sticky bit*: `t` lub `T` zamiast `x` w trzeciej trójce.
- Przykłady: `-rws--S---`, `drwxrwxrwt`.

W programie chmod

- Składnia: `[ugoa]*([-+=]([rwxXst]*|[ugo]))+|[-+=][0-7]+`
- Bardziej elastyczne, niż zapis ósemkowy, szczególnie w połączeniu z opcją `-R`.
- `X` oznacza `x` tylko dla katalogu lub jeśli już był.
- Przykład: `chmod -R go-wx,go+rX *`

Extended attributes

a	append only
A	no atime updates
c	compressed
C	no copy on write
d	no dump
D	synchronous directory updates
e	extent format
i	immutable
j	data journalling
s	secure deletion
S	synchronous updates

t	no tail-merging
T	top of directory hierarchy
u	undeletable

Read-only attributes

E	compression error
h	huge file
I	indexed directory
N	inline data
X	compression raw access
Z	compressed dirty file

Narzędzia: `lsattr`, `chattr`.

Dostęp użytkownika nieuprzywilejowanego do katalogów

Zapis możliwy tylko do:

- `/home/`*`user`*`/`
- `/tmp/`, `/var/tmp/`
- `/var/mail/`*`user`*
- I niewiele więcej, ale uwaga na drobiazgi, np.:
  `drwx-wx--T root crontab /var/spool/cron/crontab`

Odczyt

- Katalogi systemowe `/usr/`, `/var/` itp.
- Uwaga: domyślenie `drwxr-xr-x` dla katalogu `/home/`*`user`*`/`
- Domyślnie każdy użytkownik ma swoją grupę. Możliwość tworzenia dodatkowych *working groups*.
- Dobra izolacja danych różnych użytkowników, ale uwaga na dane spoza `/home/`*`user`*`/`.
- Warto zakładać dla siebie wiele kont w celu separacji danych.

Nazwy symboliczne i odpowiadające im numery

- Komputery posługują się wyłącznie liczbami (przeważnie 16- lub 32-bitowymi bez znaku).
- Ludzie wolą nazwy symboliczne (napisy).
- Popularne przestrzenie nazw:
  - Nazwy hostów (np. `www.ii.uni.wroc.pl`).
  - Nazwy protokołów sieciowych (różnych warstw, np. `ip`, `icmp`, `udp`).
  - Nazwy serwisów (portów, np. `ssh`, `domain`, `http`).
  - Nazwy użytkowników (np. `root`).
  - Grupy użytkowników (np. `staff`, `adm`).
- Różne rodzaje serwisów określają relacje między nazwami symbolicznymi i numerami.
- W Linuksie dostępem do nich zarządza *Name Service Switch* (GNU C Library).

Name Service Switch (NSS)

Rodzaje serwisów

- files — Pliki tekstowe, zwykle w katalogu `/etc`.
- db — Bazy danych Berkeley DB, zwykle w `/var/db`. Szyszy dostęp, niż do plików testowych.
- nis — Network Information Service.
- nisplus — NIS+.
- dns — Domain Name Service (tylko dla nazw hostów).

Jest też kilka innych, zależnie od konfiguracji, np. compat lub ldap.

Name Service Switch (NSS)

serwis	zawartość	funkcja	plik w /etc
hosts	nazwy hostów i adresy IP	`gethostbyname(3)`	`hosts`
services	nazwy i numery portów sieciowych	`getservent(3)`	`services`
protocols	nazwy i numery protokołów sieciowych	`getprotoent(3)`	`protocols`
networks	nazwy sieci	`getnetent(3)`	`networks`
ethers	adresy MAC		`ethers`
aliases	aliasy pocztowe	`getaliasent(3)`	`aliases`
publickey	Secure RPC dla NFS i NIS+		`publickey`
rpc	nazwy i numery RPC	`getrpcbyname(3)`	`rpc`
passwd	informacje o użytkownikach	`getpwent(3)`	`passwd`
shadow	hasła użytkowników	`getspnam(3)`	`shadow`
group	grupy podstawowe użytkowników	`getgrent(3)`	`group`
initgroups	grupy dodatkowe użytkowników	`getgrouplist(3)`	`group`
netgroup	grupy użytkowników w sieci		`netgroup`

Plik nsswitch.conf(5)

nsswitch.conf

```
passwd:         compat nisplus
group:          compat nisplus
shadow:         compat nisplus
gshadow:        files nisplus
hosts:          files dns
networks:       files
protocols:      db files
services:       db files
rpc:            db files
netgroup:       nis
```

- Zob. też nss(5).
- Wiele programów ma opcję -n, która wyłącza usługę NSS.
- Odpytywanie: getent(1). Por. `getent hosts localhost` oraz np. `dig localhost`.

Użytkownicy i grupy

- Baza informacji o użytkownikach (lokalnych): `/etc/passwd`
- Baza haseł: `/etc/shadow` (tylko dla roota)
- Baza informacji o grupach użytkowników: `/etc/group`
- Programy `whoami(1)`, `groups(1)`
- Wiele grup zezwalających na dostęp do urządzeń: `cdrom`, `floppy`, `dialout`, `bluetooth`, `audio`, `video`, `wireshark`, `kvm`, `plugdev`, `netdev` i in.
- ... i wykonywanie czynności: `staff`, `operator`, `adm` itd.
- Zwykle instalator traktuje pierwszego konfigurowanego użytkownika jako szczególnie uprawnionego.
- System weryfikacji uprawnień jest dosyć szczelny. Warto tworzyć i używać konta w celu separacji dostępu do danych (por. `lp`, `mail`, `irc`, `nobody` itd.). Oczywiście piaskownice są lepsze.

Plik /etc/passwd (zob. passwd(5))

Każdy wpis zajmuje jeden wiersz, 7 pól oddzielonych znakiem „:”

1. nazwa użytkownika (*login name*)
2. zaszyfrowane hasło, znak x (por. /etc/shadow) lub puste
3. numer użytkownika (w Linuksie ≥ 1000 dla zwykłych użytkowników)
4. numer grupy głównej użytkownika (por. /etc/group)
5. pole GECOS (komentarz)
6. katalog domowy użytkownika
7. powłoka startowa użytkownika (opcjonalnie, por. chsh(1))

Pole GECOS (General Electric Comprehensive Operating Supervisor 1962), 5 pól oddzielonych przecinkami (por. chfn(1) i login.defs(5)).

1. imię i nazwisko lub nazwa programu (f)
2. numer pokoju (r)
3. numer służbowego telefonu (w)
4. numer prywatnego telefonu (h)
5. dodatkowe informacje kontaktowe (o)

/etc/shadow, zob. shadow(5)

9 pól w formacie `/etc/passwd`. Czasy w sekundach epoki Uniksa.

1. nazwa użytkownika (*login name*)
2. zaszyfrowane hasło (ew. poprzedzone `!` lub `*`) lub puste
3. data ostatniej zmiany hasła
4. minimalny wiek hasła do zmiany
5. maksymalny wiek hasła do zmiany ($<$ poprz., zmiana zablokowana)
6. okres ostrzegania o konieczności zmiany hasła
7. okres możliwości zalogowania z wymuszeniem zmiany hasła po wygaśnięciu jego ważności
8. data wygaśnięcia konta (jeśli 0, tj. 1/1/1970, konto zablokowane)
9. pole zarezerwowane

Dodatkowo pliki:

- `/etc/{passwd-,shadow-,group-,gshadow-,subuid-,subgid-}` — zawartość plików sprzed ostatniej zmiany
- `/var/backups/{passwd,shadow,group,gshadow}.bak` — periodyczne kopie zapasowe (zob. `/etc/cron.daily/passwd`)

Grupy, podużytkownicy i podgrupy

Grupy

- Plik `/etc/group` — 3 pola: nazwa grupy, hasło, lista użytkowników.
- Hasła do grup zwykle w `/etc/gshadow`. Wówczas także możliwość zdefiniowania administratorów grup.
- Można być członkiem grupy lub mieć hasło do grupy.
- Polecenie `newgrp(1)`.
- Polecenia `su(1)` i `sg(1)`.

Podużytkownicy i podgrupy

- Pliki `/etc/{subuid,subgid}`
- Potrzebne np. przy uruchamianiu kontenerów nieuprzywilejowanych.

Jak zmienić zapomniane hasło roota?

Zwykle działa

- Uruchom system ratunkowy, np. z pendrive'a.
- Zamontuj *rootfs* systemu ratowanego np. w `/target/`.
- Pierwszy wiersz `/target/etc/passwd` zmień na `root::0:0:root:/root:/bin/bash`
- Uruchom system ratowany.
- Zaloguj się na konto `root` podając puste hasło.
- Ustaw nowe hasło roota poleceniem `passwd(1)`.

Warianty

- Usunąć hasło z `/etc/shadow`.
- W-chroot-ować się w system ratowany i wykonać polecenie `passwd(1)`.

Morał

- W razie fizycznego dostępu do komputera hasło roota nie jest zabezpieczeniem.
- Rootfs powinien się znajdować na zaszyfrowanej partycji.

Narzędzia do zarządzania użytkownikami

- Zamiast ręcznie edytować `/etc/passwd` itd. — specjalne programy.
- W Debianie pakiety: `passwd`, `shadow-utils` i `adduser`.
- Niskopoziomowe narzędzia `useradd(8)`, `userdel(8)`, `usermod(8)`, `vipw(8)`, `vigr(8)`.
- Zakładanie wielu użytkowników na raz: `newuser(8)`.
- Narzędzia Debiana: `adduser(8)`, `deluser(8)`, `addgroup(8)`, `delgroup(8)`. Konfiguracja w `adduser(5)`, `deluser(5)`.

Zarządzanie użytkownikami

- Dodanie użytkownika: `adduser` *`user`*
- Dodanie użytkownika do grupy: `adduser` *`user group`*
  Uwaga: użytkownik będzie należał do tej grupy w sesji logowania rozpoczętej po wykonaniu tego polecenia — trzeba się wylogować i zalogować.
- Zablokowanie użytkownika: `usermod -e 1970-01-01` *`user`*.
- Odblokowanie użytkownika: `usermod -e` *`user`*
- Zablokowanie/odblokowanie *hasła* użytkownika: `passwd [-l | -u] user`
- Zmiana hasła użytkownika: `passwd` *`user`*
- Zmiana czasów ważności hasła: `chage(1)`
- Wykonanie powłoki jako użytkownik: `su` *`user`*
- Wykonanie programu w podanej grupie: `sg` *`grupa program`*

/etc/sudoers

- sudo — selektywne nadawanie uprawnień do wykonywania jako root pojedynczych programów.
- W Debianie pakiet sudo.
- Baza danych: plik /etc/sudoers, zob. sudoers(5).
- Nie modyfikować zwykłym edytorem! Program visudo(8): brak hazardów czasowych (zakłada locka) i pozostawiania kopii zapasowych. Sprawdza poprawność składniową pliku przy zapisie.
- Także sudoedit, sudo -e — edycja plików zamiast wykonywania.

Składnia w skrócie

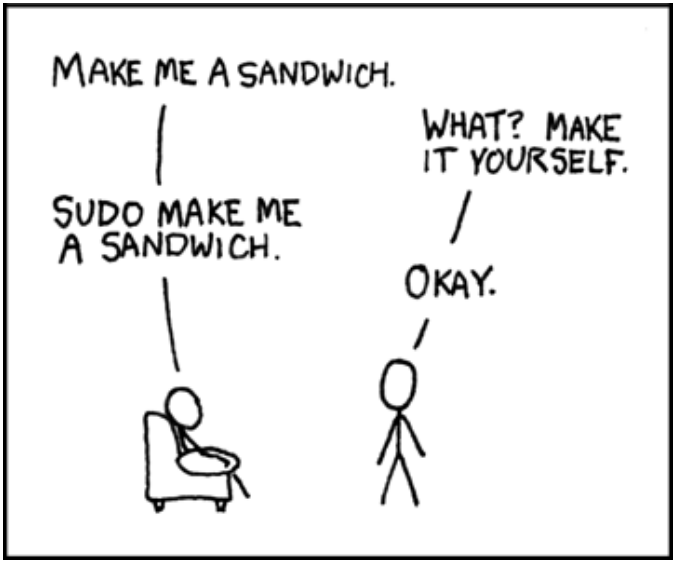
kto skąd=(*jako-kto*:*z-jaką-grupą*) *co-wykonać*

- ALL oznacza wzorzec pasujący do wszystkiego.
- Przykład: jan localhost=(root:staff) /bin/ip
  — jan może uruchomić ip(8) jako root w grupie staff.
- Używać bezwzględnych ścieżek do programów!

Program sudo

- Wykonanie pojedynczego polecenia jako root:
  `su - -c` *`polecenie`*
  wymaga podania hasła *roota*.
- Wykonanie pojedynczego polecenia jako root:
  `sudo` *`polecenie`*
  wymaga podania hasła *użytkownika*.
- Użytkownik nie musi znać hasła roota.
- Hasło roota może w ogóle nie być dostępne.
- `sudo [-u` *`user`*`] -i` — uruchomienie powłoki jako użytkownik *`user`*. Lepsze niż `sudo su` lub `sudo /bin/bash`.
- Uwaga: `sudo` w skyptach.
- Pamiętać o opcji `-k`.
- W Ubuntu był *exploit* na `sudo -k`.
- Nie używać bez potrzeby opcji `:NOPASSWD`!

XKCD sudo


MAKE ME A SANDWICH.
WHAT? MAKE IT YOURSELF.
SUDO MAKE ME A SANDWICH.
OKAY.

Hasło roota?

Czy blokować?

- Wszystko, co nie jest używane, powinno być zablokowane.
- W niektórych dystrybucjach domyślnie hasło roota jest wyłączone.
- Instalator Debiana pyta, choć sugeruje, żeby pozostawić włączone.
- Można zablokować: `sudo passwd -l root`, a jak się nie spodoba — odblokować `sudo passwd -u root`.
- Zawsze można zresetować, jeśli nawet się zapomni.
- Uwaga: jedyne hasła, których *absolutnie nie wolno* zapomnieć, to hasła do kryptografii (zaszyfrowane partycje itp.).

Krytyka sudo

- Program bardzo duży i skomplikowany.
- Skomplikowany plik konfiguracyjny — łatwo błednie skonfigurować.
- Wykryto poważne podatności, zob. np. Animesh Jain: CVE-2021-3156: Heap-Based Buffer Overflow in Sudo (Baron Samedit).
- W OpenBSD `doas(1)`, zob. Ted Unangst: `https://flak.tedunangst.com/post/doas`.

Logi systemowe

- Kopalnia wiedzy o systemie.
- Warto je stale przeglądać i analizować.
- Katalog `/var/log/`.
- Większość plików do odczytu dla grupy `adm` — warto dodać siebie do tej grupy, by móc przeglądać logi jako zwykły użytkownik.
- Klasycznie: demon `(r)syslog`, zob. `rsyslog.conf(5)`, `rsyslogd(8)`.
- W systemd: `journalctl(1)`.
- Polecenie `logger(1)`.
- Automatyczne usuwanie starych logów: `logrotate(8)`.
- Warto wydłużyć „czas życia" logów w `/etc/logrotate.conf`, `/etc/logrotate.d/`.
- Programy `ccze(1)`, `clog(1)`, `colortail(1)`, `lwatch(1)` itp.