# Konta użytkowników

**Przygotował Michał Tracewicz 2019**

## Spis treści



---

## Historia

System GNU-Linux powstał w roku 1991. Jest on wzorowany na systemie UNIX (lata 70-te dwudziestego wieku) wywodzącym się z Bell Labs. System ten był od początku projektowany z założeniem, że będzie to system przeznaczony do pracy wielu użytkowników. Było to spowodowane tym, że czasy systemu UNIX to czasy komputerów będących drogimi pojedynczymi urządzeniami do których logowało się wielu użytkowników za pomocą zewnętrznych terminali.

## Pliki

### Uprawnienia do plików

W systemach Linux możemy wyświetlić listę plików za pomocą polecenia ls.

```
ls -la
drwxr-xr-x 1 mtracewicz mtracewicz  4096 Oct  4 09:05 .oh-my-zsh
```

Kolejno od lewej wpis zawiera:

- Typ pliku:
    1. **-** dla plików zwykłych
    2. **d** dla katalogów
    3. **c** dla znakowych plików urządzenia
    4. **b** dla blokowych plików urządzenia
    5. **s** dla gniazd
    6. **p** dla nazwanych potoków
    7. **l** dla łączy symbolicznych
- **Uprawnienia kolejno dla:**
    1. Użytkownika

2. Grupy

3. Innych

Dla każdej z tych kategorii możemy wyróżnić trzy rodzaje uprawnień

( Myślnik '-' wyświetlany przez polecenie ls oznacza, że dany użytkownik nie posiada danego prawa)

W wypadku gdy jest to plik nie będący katalogiem

r - oznaczające możliwość czytania

w - oznaczające możliwość edycji

x - oznaczające możliwość uruchomienia

W wypadku przeciwnym

r - oznaczające możliwość czytania plików zawartych w katalogu (możemy wylistować pliki w nim zawarte)

w - oznaczające możliwość tworzenia i usuwania plików w katalogu

x - oznaczające możliwość dostępu do katalogu (możemy kopiować z katalogu oraz ustawić go jako katalog roboczy)

Możemy to interpretować jako:

r-x prawo dostępu do katalogu

--x prawo dostępu do plików o znanej nazwie

Uprawnienia te możemy również zapisać w postaci trzech liczb w systemie ósemkowym.

Gdzie:

| | | | |
|---|---|---|---|
| **0** | **---** | **4** | **r--** |
| **1** | **--x** | **5** | **r-x** |
| **2** | **-w-** | **6** | **rw-** |
| **3** | **-wx** | **7** | **rwx** |

- Liczba twardo dowiązanych łączy
- Właściciel
- Grupa
- Wielkość
- Data i godzina ostatniej modyfikacji
- Nazwa pliku

**Możemy modyfikować uprawnienia dostępu za pomocą polecenia chmod.**

Poniżej przykład użycia:

```
#Nadajemy użytkownikowi możliwość uruchomienia pliku
chmod u+x exampleFile
#Nadajemy grupie prawo edycji pliku
chmod g+w exampleFile
#Odbieramy pozostałym użytkownikom możliwość czytania pliku
chmod o-r exampleFile
#Odbieramy wszystkim użytkownikom możliwość uruchomienia pliku
chmod a-x exampleFile
#Wstawiamy uprawnienia w formacie rwxr-xr-x
chmod 755 exampleFile
```

**Mamy możliwość zmiany właściciela pliku oraz grupy za pomocą polecenia chown.**

```
#Zmieniamy właściciela pliku exampleFile na użytkownika mtracewicz a grupę na
student.
chown mtracewicz:student exampleFile
#Zmieniamy właściciela folderu exampleDir oraz wszystkich zawartych w nim plików
na mtracewicz.
chown -R mtracewicz exampleDir
```

Alternatywnie możemy zmienić grupę pliku za pomocą polecenia chgrp.

```
#Zmieniamy grupę pliku example file na student
chgrp student exampleFile
```

## Access Control Lists (ACL)

Jest to dodatkowy, bardziej elastyczny system kontroli dostępu do zasobów na dysku zaprojektowany aby uzupełniać ten znany z systemów Unix. Możemy go użyć np. w sytuacji gdy jakiś użytkownik nie jest członkiem grupy a chcielibyśmy mu dać dostęp do jakiegoś pliku nie czyniąc go członkiem grupy.

```
#Możemy wyświetlić jakie aktualnie mamy ograniczenia na pli poleceniem
getfacl testFile
#Aby nadać prawa użytkownikowi użyjemy:
setfacl -m "u:user:uprawnienia" /sciezkaPliku
#Aby nadać prawa grupie:
setfacl -m "g:grupa:uprawnienia" /sciezkaPliku
#Jeżeli użyjemy opcji -b usuniemy uprawnienia acl
setfacl -b /sciezkaPliku
```

W systemie Linux informacje o użytkownikach znajdują się w plikach:

- /etc/passwd
- /etc/group
- /etc/shadow

**Plik /etc/passwd**

W tym pliku przechowywane są informacje o użytkownikach.

```
#Wszyscy użytkownicy mają możliwość odczytu pliku, gdybyśmy ją odebrali nie
bylibyśmy w stanie zmienić użytkownika, a wiele aplikacji przestało by działać
poprawnie nie mając dostępu do danych w nim dostępnych (stąd późniejszy podział na
/etc/passwd i /etc/shadow)
-rw-r--r-- 1 root root 1594 10-02 21:50 /etc/passwd
#Przykładowy wpis w pliku /etc/passwd na Manjaro Linux
mtracewicz:x:1000:1001:Michał Tracewicz:/home/mtracewicz:/bin/bash
#|---1----|2|-3--|-4--|------5---------|------6--------|----7-----
#Składnia:
#1 - nazwa użytkownika
#2 - hasło (zwykle znajdziemy tu x ponieważ aktualnie przechowuje się je w pliku
/etc/shadow)
#3 - id użytkownika
#4 - id grupy
#5 - komentarz/opis/informacja o użytkowniku
#6 - folder domowy
#7 - powłoka domyślna
```

**Plik /etc/group**

W tym pliku przechowywane są informacje o poszczególnych grupach w systemie. Dla przykładu:

```
-rw-r--r-- 1 root root 988 10-03 14:42 /etc/group
#Przykładowy wpis w pliku /etc/group na Manjaro Linux
sys:x:3:bin,mtracewicz
#|1|2|3|-------4------
#Składnia
#1 - nazwa grupy
#2 - hasło (zwykle puste, ale może zawierać zaszyfrowane hasło)
#3 - id grupy
#4 - lista użytkowników należących do grupy
```

Możemy sprawdzić do jakich grup należy dany użytkownik poprzez użycie polecenia groups.

```
#Przykład użycia polecenia groups dla użytkownika mtracewicz
groups mtracewicz
wheel lp sys network power autologin vboxusers mtracewicz
```

**Plik /etc/shadow**

W tym pliku przechowujemy hasła użytkowników.

```
#Możemy zauważyć, że w przeciwieństwie do popszednich plików plik /etc/shadow może
być zarówno czytany jak i edytowany tylko przez użytkownika root
-rw------- 1 root root 922 10-02 21:50 /etc/shadow
#Przykładowy wpis w pliku (wzięty z https://www.slashroot.in/how-are-passwords-
stored-linux-understanding-hashing-shadow-utils i delikatnie zmodyfikowany)
testUser:$1$Etg2ExUZ$F9NTP7omafhKIlqaBMqng1:15651:0:99999:7:::
#---1---|---------------2----------------|--3--|4|--5--|6|7|8|9
#1 - nazwa użytkownika
#2 - zaszyfrowane hasło (pod przykładem znajduje się informacja o tym jak wygląda
ten proces)
#3 - ile dni minęło od ostatniej zmiany hasło
#4 - ile minimalnie dni jest wymaganych miedzy zmianami hasła (jak często można
zmieniać hasła)
#5 - ile maksymalnie dni jest dopuszczalne między zmianami hasła
#6 - na ile dni przed następną wymaganą zmianą hasła użytkownik dostanie
ostrzeżenie
#7 - ile dni po wygaśnięciu hasła konto będzie wyłączone
#8 - po ilu dniach od 01.01.1970r. konto zostanie wyłączone
#9 - pole jeszcze nie jest obecnie używane
```

**W jaki sposób hasła są zabezpieczane?**

Hasło przechowywane w pliku /etc/shadow możemy podzielić na trzy części rozdzielone znakiem '$'. Przyjmuje ono postać $ID$SALT$HASHED.

**Algorytm hashujący** - algorytm, który z podanych danych tworzy unikatowy ciąg znaków zadanej długości. Jest to funkcja, której nie da się w prosty sposób odwrócić tzn. znając hash nie możemy odzyskać danych wejściowych (To odróżnia algorytm hashujący od szyfrującego, ten drugi jest łatwo odwracalny, gdy znamy jakąś kluczową informację np. klucz prywatny w RSA).

**ID** jest to wartość wskazująca jakiego algorytmu hashującego użyto. Może on przyjąć wartości:

- 1 - oznacza algorytm MD5 (Nie jest zalecane jego użycie, na obecnym sprzęcie jest łatwy do złamania)
- 2 - oznacza algorytm Blowfish (W przeciwieństwie do MD5 czy algorytmów z rodziny SHA jest to algorytm szyfrujący nie hashujący)
- 2a - oznacza algorytm eksblowfish (Jest to inna wersja poprzedniego algorytmu, jego nazwę można rozwinąć do "expensive key schedule blowfish " różnica w nich polega na funkcji, która jest użyta do przetransformowania kluczy w podklucze )
- 5 - oznacza algorytm SHA-256 (Długość słowa to 32 bity. Został już złamany za pomocą ataku opierającego się na odkryciu wiadomości na podstawie hashu i jego długości)
- 6 - oznacza algorytm SHA-512 (Długość słowa to 64 bity. Na dzień pisania tego referaty jest on niezłamany i w mojej opinii jest najlepszym z algorytmów dostępnych do wyboru)

```
#Możemy użyć poniższego polecenia aby wyświetlić aktualnie używany algorytm
authconfig --test | grep hashing
#W ten sposób możemy zmienić algorytm hashujący na sha512
authconfig --passalgo=sha512 --update
#Po zmianie algorytmu musimy pamiętać, że użytkownik musi zmienić hasło aby
zostało ono zahashowane nowym algorytmem. Możemy kazać użytkownikowi zmienić hasło
przy następnym logowaniu za pomocą polecenia
chage -d 0 testUser
#Więcej o wymuszaniu zmianny hasła użytkownika w późniejszym podpunkcie.
```

**Salt** jest to losowo wygenerowany ciąg znaków, który jest łączony z hasłem użytkownika w celu zwiększenia bezpieczeństwa.

**HASHED** jest to wartość wynikowa algorytmu hashującego na haśle użytkownika,które jest połączoe z saltem.

#### Co daje nam salt?

Salt pomaga nam zabezpieczyć nasze hasła przed atakami typu dictionary atack czy rainbow table (więcej o tym w następnym podpunkcie). Dzięki zastosowaniu wartości salt nawet dwa dokładnie te same hasła będą posiadały inny hash. Co za tym idzie nawet jeżeli osobie atakującej udało się złamać jedno hasło nie będzie ona w stanie znaleźć osoby o identycznym haśle, ponieważ ich zahaszowana wartość będzie inna.

#### Jak można łamać hasła?

Najprostszym sposobem łamania haseł są tak zwany dictionary atack i rainbow table.

Pierwszy z nich to atak oparty na prostej metodzie siłowej, gdzie znając algorytm hashujący próbujemy użyć go na wszystkich prawdopodobnych hasłach (najczęściej robi się to sprawdzając listę najczęstszych haseł oraz dodając do niej te same hasła tylko ze zmienioną wielkością liter czy podmieniając litery na cyfr np. 'A' -> 4, 'O'->0 itp.) i znaleźć takie, które zgadza się z jednym z tych które pozyskaliśmy.

Drugi sposób to pozyskanie bazy w której najpopularniejsze hasła są już zahaszowane wraz z informacją o tym jaki algorytm został użyty. Następnie sprawdzamy czy któryś z posiadanych przez nas hashy znajduję się w tej bazie i odczytujemy z niej hasło.

W pierwszym przypadku zużywamy niewiele pamięci jednak bardzo dużo mocy obliczeniowej, w drugim ataku jest dokładnie odwrotnie. Przed oboma tymi atakami pomaga nam bronić się wartość salt. Dzięki generowaniu losowej wartości do naszych haseł mamy niemal pewność, że hash, który uzyskamy (nawet jeżeli użytkownik ustawi sobie hasło = haslo123!) nie znajdzie się w żadnej z rainbow tables. W przypadku dictionary atack dodanie wartości salt masywnie zwiększa liczbę możliwości, które atakujący musi sprawdzić a co za tymi idzie zwiększamy czas, który musi poświęcić na próbę złamania każdego z haseł.

#### Czym jest silne hasło?

Silne hasło to takie które zawiera minimum dwanaście znaków, zarówno wielkie jak i małe litery, znaki specjalne i cyfry. Dodatkowo nie powinno być zlepkom słów, które można znaleźć w słowniku oraz nie powinno polegać na prostych substytucjach jak 'o'->0'. (Na podstawie: https://www.howtogeek.com/195430/how-to-create-a-strong-password-and-remember-it/ )

Jeżeli nasze hasło zawiera tylko 12 małych liter to jest ich możliwie 26 ^ 12,natomiast w wypadku bezpiecznego hasła jest ich minimum 56 ^ 12 (liczba ta jest większa zależnie od tego jakie znaki dopuszczamy jako znaki specjalne).

Dodatkowo należy pamiętać,  że długość hasła ma istotny wpływ na jego bezpieczeństwo. Jak już pokazaliśmy ośmioznakowych haseł jest ~56 ^12 natomiast dodanie np. czterech znaków znacząco zwiększa ilość możliwości 56^16. Pokazuje to, że każdy kolejny znak zwiększa ilość obliczeń, którą musi wykonać ktoś, kto próbuje zgadnąć nasze hasło.

Warto także pamiętać o tym, że hasło nie powinno zawierać żadnych danych z nami związanych takich jak imię, nazwisko czy rok urodzenia. Użytkownicy często decydują się na hasło zawierające takie informacje, z powodu tego, że ułatwienia im to jego zapamiętanie. Jednak jest to problematyczne, ponieważ informacje te są łatwe do pozyskania i jako, że tego typu hasła występują często to próba ich zgadnięcia (na podstawie pozyskanych informacji) często jest jednym z pierwszych sposobów w jaki hakerzy próbują się włamać na konto użytkownika.

# Administracja kontami użytkowników

## Wyświetlanie listy aktualnie zalogowanych użytkowników

W systemie Linux możemy wyświetlić listę aktywnych użytkowników za pomocą polecenia users.

```
users
# Przykładowy wynik polecenia. Zawiera on listę zalogowanyhc użytkoników
rozdzielonych spacją.
test testUser exampleUSer
```

## Wyświetlanie ostatnich logowań użytkowników

W systemie Linux możemy wyświetlić listę ostatnich logowań użytkowników za pomocą polecenia last.

```
#Polecenie wyświetli logowania użytkownika mtracewicz w kolejności od najstarszych
do najnowszych, możemy także wyświetlić je dla konkretnego tty/host
last mtracewicz
#przykładowy wpis
mtracewicz pts/9    188.147.44.127.nat.umt pią  4 paź 09:15 - 09:18  (00:02)
#----1----|--2------|--------3------------|---------------4---------------|
#1 - nazwa użytkownika
#2 - tty (nazwa terminalu)
#3 - host z którego użytkownik się loguje/miejsce dostępu
#4 - data początku - końca logowania i w nawiasie czas trwania
```

## Dodawanie użytkowników

W systemie Linux możemy dodać użytkownika za pomocą polecenia useradd.

```
#Polecenie, które doda do systemu użytkownika test, pobierze domyślne wartości z pliku /etc/default/useradd może zostać wykonane tylko przez użytkownika root lub użytkownika posiadającego uprawnienia do polecenia sudo
useradd test
#Jeżeli chcemy utworzyć katalog domowy użytkownikowi musimy użyć opcji -m
useradd -m test
#Jeżeli użyjemy opcji -d możemy utworzyć katalog domowy w miejscu innym niż domyślne
#Jeżeli chcemy dodać użytkownika do grup użyjemy opcji -G
useradd test -G student,inf
#W tym wypadku utworzymy użytkownika test i dodamy go do grup student i inf
#Jeżeli checmy ustawić np. po ilu dniach wygasa hasło użyjemy opcji -K
useradd test -K PASS_MAX_DAYS = 3
#Jeżeli chcemy dodać komentarz jak np. imię i nazwisko to użyjemy opcji -c
useradd test -c "Jan Kowalski"
```

## Usuwanie użytkowników

W systemie Linux możemy usunąć użytkownika za pomocą polecenia userdel

```
#Tym poleceniem usuniemy użytkownika test, może ono zostać wywołane tylko przez użytkownika root lub użytkownik posiadający uprawnienia do polecenia sudo
userdel test
#jeżeli chcemy usunąć także katalog domowy użytkownika użyjemy opcji -r
userdel -r test
```

## Modyfikacja użytkowników

W systemie Linux możemy modyfikować użytkownika za pomocą polecenia usermod.

```
#Tym poleceniem zmieniamy katalog domowy użytkownika test na katalog /test
usermod -d /test test
#Jeżeli chcemy wraz ze zmianą katalogu domowego przenieść do niego pliki ze starego używamy opcji -m
usermod -d /test -m test
#Tym poleceniem zmienimy login uzytkownika test na jankowalski
usermod -l test jankowalski
#Tym poleceniem zmienimy id użytkownika test na 1000
usermod -u 1000 test
#Tym polceniem zmienimy główną grupę użytkownika test na pracownik (grupa musi już istnieć)
usermod -g pracownik test
#Tym poleceniem dodamy wiele grup (student,informatyka) dla użytkownika test. Opcja -a sprawia, że użytkownik nie utraci obecnie przypisanych grup
usermod -a -G student,informatyka       test
#Tym poleceniem zmienimy datę wygaśnięcia konta użytkownika test na pierwszy stycznia 2020. Data musi być w formacie YYYY-MM-DD
usermod -e 2020-01-01 test
```

```
#Tym polceniem zmeinimy powłokę użytkownika test na zsh
usermod -s /bin/zsh test
```

## Zmiany hasła

W systemie Linux możemy modyfikować hasło użytkownika za pomocą polecenia passwd.

```
#Każdy użytkownik może zmienić własne hasło
passwd
#Wyświetli nam się taki komunikat
Changing password for mtracewicz.
#Zostaniemy poproszeni o aktualne hasło
Current password:
#Następnie o nowe hasło
New password:
#Oraz powtórzenie w celu potwierdzenia
Retype new password:
#Użytkownik root może zmodyfikować hasło dowolnego użytkownika. Tym poleceniem
zminimy hasło użytkownika test (jako root nie zostaniemy zapytani o poprzednie
hasło)
passwd test
#Polecenie passwd pozwala nam też usunąć hasło opcją -d
passwd -d test
```

**Jak wymusić zmianę hasła?**

Aby wymusić zmianę hasła możemy użyć wcześniej wspomnianego polecenia passwd lub dedykowanego polecenia chage.

```
#Aby wymusić zmianę hasła przy pierwszym logowaniu hasłem nadanym prze root-a
możemy użyć opcji -e
passwd -e test
#Polcenie change służy do zarządzania wygasaniem haseł. Możemy użyć polecenia
change do wyświetlenia aktualnych informacji o datach związanych z hasłem
użytkownika w ten sposób:
chage -l mtracewicz
#Możemy zmienić maksymalną ilość dni między zmianami hasła z opcja -M. W tym
przykładzie ustawimy, że użytkownik mtracewicz musi zmienić hasło co maksymalnie 5
dni
chage -M 5 mtracewicz
#Jeżeli nie chcemy aby użytkownik zmieniał hasło codziennie możemy użyć opcji -m.
W tym przykładzie zmienimy, że użytkownik mtracewicz będzie mógł zminić hasło
najczęściej co dwa dni.
chage -m 2 mtracewicz
```

## Blokowanie / odblokowanie konta

Wcześniej wymienionym poleceniem usermod możemy zablokować lub odblokować użytkownika.

```
#tym polceniem blokujemy użytkownika
usermod -L test
#tym polceniem odblokujemy użytkownika
usermod -U test
```

## Zmiana tożsamości użytkownika

W systemie Linux mamy dwa polecenia służące do zmiany tożsamości: sudo, su.

Różnica między nimi polega na tym, że polecenie sudo służy do wykonania polecenia jako inny użytkownik zaś polecenie su służy do zmiany użytkownika

**Polecenie su:**

```
#Wykonanie polecenia su bez argumentów zmieni użytkownika na root
su
#Możemy dopisać nazwę użytkownika, aby wybrać w jakiego użytkownika chcemy się
zmienić
su testUser
#Użyjemy opcji -s kiedy chcemy wybrać powłokę
su -s /bin/zsh
#Opcja -s wybierze powłokę w kolejności:
#1. wprowadzona przez nas w poleceniu
#2. ze zmiennej $SHELL (jeżeli użyto opcji --preserve-enviroment, opcja ta
zachowuje nasze zmienne środowiskowe z wyjątkiem $PATH i $IFS)
#3. odczytaną z pliku /etc/passwd
```

Polecenie su możemy konfigurować za pomocą pliku /etc/login.defs. Możemy tam np. ustawić logowanie do pliku wszystkich poleceń wykonanych przez użytkownika po użyciu polecenia su.

**Polecenie sudo:**

```
#W tym przykładzie użyjemy polecenia sudo, aby zainstalować dodatkowe
oprogramowanie
sudo dnf install vim
#Możemy użyć opcji -u, aby wybrać jako jaki użytkownik checmy wykonać dane
polecenie
sudo -u test vim test.c
#Możemy użyć opcji -g, aby wykonać polecenie jakbyśmy byli członkami innej grupy
sudo -g 999 vim test.c
```

Szczegóły działania polecenia sudo sąkonfigurowane w pliku /etc/sudoers. Plik jest podzielony na trzy sekcje: defaults, aliases oraz user specifications. Sekcja defaults zawiera konfiguracje, które będą automatycznie

dopisywane do każdego rekordu, mogą one jednak być nadpisywane dla konkretnego wpisu. Sekcja aliases zawiera zmienne, które służą do grupowania wielu nazw do jednego słowa. Istnieją cztery typy aliasów:

- User_Alias - łączymy kilku użytkowników w grupę np.: User_Alias testowi = test1, test2. Nie musimy tu redefiniować grup, które zdefiniowaliśmy w systemie. Aby użyć grupy systemowej wstawimy przed jej nazwą '%' np.: User_Alias testowi = %testowi.
- Runas_Alias - jak wyżej z różnicą, że jest to grupa użytkowników jako którzy polecenie ma zostać wykonane.
- Host_Alias - służy do grupowania hostów z których użytkownik wykonujący polecenie sudo się loguje.
- Cmnd_Alias - służy do grupowania poleceń np.: Cmnd_Alias fileList = /bin/ls

Dla każdego z tych typów aliasów istnieje wbudowany alias ALL. Dodatkowo dodanie '!' przed nazwą polecenia oznacza, że użytkownik nie będzie mógł go wykonać

W sekcji user specifications zawieramy konkretne wpisy opisujące możliwości danego użytkownika.

```
#Wpis ma postać
user host = (runas) command[, command, ...]
#Przykładowy wpis
testUser ALL = (%students) /bin/ls
#----1--|-2--|-----3-----|---4---|
#1. użytkownik/grupa systemowa (poprzedona %)/User_Alias, któremu przyznajemy
prawa wykonania sudo(w tym wypadku testUser)
#2. host/Host_Alias z którego może on wykonać to polecenie (w tym wypadku dowolny)
#3. może wykonać jako użytkownik/grupa systemowa (poprzedona %)/User_Alias (w tym
wypadku grupa students)
#4. polcenia do których otrzymuje dostęp (w tym wypadku polecenie ls)
testUser2 ALL = (ALL) ALL,!/bin/vim
#W powyższym przykładzie daliśmy prawo wykonania wszystkich poleceń z wyjątkiem
polecenia vim użytkownikowi testUser2 na wszystkich hostach jako dowolny
użytkownik
```

Warto zaznaczyć, że domyślnie polecenie sudo pyta użytkownika o jego hasło, po czym zapamiętuje to hasło na pięć minut.

**Dobre praktyki**

Przy konfiguracji pliku /etc/sudoers warto pamiętać o kilku prostych zasadach aby polepszyć bezpieczeństwo naszego systemu. Przde wszystkim warto wyłączyć każdemu z użytkowników możliwość użycia polecenia su przez polecenie sudo. Jest to ważne, ponieważ w przeciwnym wypadku dowolny użytkownik może się zalogować jako root używając swojego hasła.

```
#W wypadku braku tego zabezpieczenia poniższym poleceniem możemy się zalogować na
użytkownika root z użyciem hasła do naszego konta!
sudo su
```

Dodatkowo warto wyłączyć możliwość uruchamiania plików z katalogów do których zwykły użytkownik ma prawo zapisu. Dzięki temu zwykły użytkownik nie będzie w stanie uruchomić programów pobranych z Internetu. Możemy to zrobić poprzez dodanie aliasu "Cmnd_Alias NAZWA_ALIASU = /home/, */tmp/*, /var/tmp/*" oraz dodając przeciwny alias do zaufanych lokacji programów "Cmnd_Alias BEZPIECZNE = /sbin:/bin:/usr/sbin:/usr/bin";

```
#Przykładowy plik /etc/sudoers (wzorowany na
https://stelfox.net/blog/2016/02/better-practices-with-sudo/)
# /etc/sudoers
#Alias do poleceń, których nie chcemy aby użytkownicy używali, w naszym wypadku
jest to polecenie su z wyżej wymienionego powodu
Cmnd_Alias BLACKLISTED_APPS = /bin/su
#Alias do folderów z którch nie chcemy aby użytkownik mógł uruchamiać programy
Cmnd_Alias USER_WRITEABLE = /home/*, /tmp/*, /var/tmp/*
#Dopisujemy do wszystkich rekordów foldery z, których chcemy pozwolić uruchamiać
programy
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin
#Pozwalamy użytkownikowi root robić wszystko
root ALL = (ALL) ALL
#Aplikujemy nasze zasady dla wszystkich pozostałych użytkowników
%zwykliUzytkownicy ALL = (root) ALL, !BLACKLISTED_APPS, !USER_WRITABLE
```

## Dodawanie grup

W systemie Linux możemy dodać grupę za pomocą polecenia useradd.

```
#Tym poleceniem dodamy grupę testGroup
groupadd testGroup
#Z opcją -g możemy sami wybrać id grupy (musi być unikatowe i nie ujemne)
groupadd -g 999 testGroup
```

## Usuwanie grup

W systemie Linux możemy usunąć grupę za pomocą polecenia groupdel.

```
#Tym polceniem usuniemy grpuę testGroup. Grupa musi istnieć i my jako
administratorazy musimy zadbać aby grupa, którą usuwamy nie była główną grupą dla
żadnego z użytkowników
groupdel testGroup
```

## Modyfikacja grup

W systemie Linux możemy zmodyfikować grupę za pomocą groupmod.

```
#Możemy zmodyfikować id grupy przy użyciu opcji -g
groupmod -g 999 testGroup
#Możemy też zmodyfikować nazwę grupy za pomocą opcji -n. W tym przykładzie zminimy
nazwę grupy testGroup na myGroup
groupmod -n myGroup testGroup
```

### Zmiana tożsamości grup

W systemie Linux możemy zmienić aktualną grupę na inną za pomocą polecenia newgrp.

```
#W wypadku nie podania argumentów program zaloguje nas do naszej domyślnej grupy
nadanej nam w /etc/passwd
newgrp
#Zmienimy grupę od id 999
newgrp 999
```

Jeżeli grupa ma hasło, a nie jesteśmy jej członkiem zostaniemy poproszeni o hasło. W wypadku gdy grupa ma puste hasło i nie jesteśmy członkami grupy to dostęp nie zostanie nam przyznany. W wypadku, gdy użytkownik nie ma hasła a grupa ma, to zostanie on poproszony o jego wpisanie (nie dotyczy użytkownika root).

### Sprawdzanie dostępnych tożsamości

Za pomocą polecenia id możemy sprawdzić dane o tożsamości użytkownika oraz wszystkie grupy w systemie.

```
#W wypadku nie podania argumentów polecenie id zwróci nam informacje o naszym id
użytkownika, id grupy i wszystkich grupach do których należymy np.:
id
uid=1000(mtracewicz) gid=1000(mtracewicz)
groups=1000(mtracewicz),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio)
,30(dip),44(video),46(plugdev),108(lxd),114(netdev)
#Możemy tym poleceniem wylistować informacje o innych użytkownikach
id root
uid=0(root) gid=0(root) groups=0(root)
#Jest też możliwość wyświetlenia wszystkich grup w naszym systemie za pomocą opcji
"g"
id -g
#Jeżeli chcemy otzrymać nazwy zamiast id grup użyjemy opcji "n"
id -gn
```

## Zasoby systemowe

**Listowanie procesów i ich zasobów**

W systemie Linux istnieje kilka możliwości wyświetlenia aktywnych procesów. Możemy użyć do tego poleceń: ps, top, fuser oraz lsof.

**ps**

```
#Podstawowe wywołanie
ps
#Wyjście polecenie ma format:
#PID TTY          TIME CMD
#i wyświetla tylko procesy aktualnego użytkownika
#Aby wyświetlić wszystkie procesy w systemie możemy użyć dwóch wersji polecenia ps
(posiada ono różne wersje w standardzie UNIX, BSD i GNU)
#Różnicą między tymi poleceniami jest format wyświetlonego wyjścia
ps -ely
#Wyjście polecenia ma format
#S   UID   PID  PPID  C PRI  NI   RSS    SZ WCHAN  TTY          TIME CMD
ps -axu
#Wyjście polecenia ma format
#USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
#Jeżeli chcemy wyświetlić procesy danego użytkownika używamy opcji -U
ps -U testUser
```

**top**

```
#Polecenie top w przeciwieństwie do polecenia ps jest dynamicznie aktualizowane i
wyświetla aktualny stan zasobów systemu
top
#Przykładowy wynik polecenia top:
top - 14:20:31 up  1:41,  0 users,  load average: 0.52, 0.58, 0.59
Tasks:   5 total,   1 running,   4 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.8 us,  2.3 sy,  0.0 ni, 95.1 id,  0.0 wa,  0.8 hi,  0.0 si,  0.0 st
KiB Mem :  8241956 total,  4504524 free,  3508080 used,   229352 buff/cache
KiB Swap: 25165824 total, 25064552 free,   101272 used.  4600144 avail Mem
PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
1 root      20   0    8892    296    260 S   0.0  0.0   0:00.04 init
63 root      20   0   19464    744    576 S   0.0  0.0   0:00.00 sshd
242 root      20   0    8904    208    160 S   0.0  0.0   0:00.01 init
243 mtracew+  20   0   17012   3696   3588 S   0.0  0.0   0:00.33 bash
442 mtracew+  20   0   17620   2032   1504 R   0.0  0.0   0:00.01 top
# widzimy tu status aktualnie uruchomionych zadań, obciążenie CPU, pamięć wolną,
zajętą a także przeniesioną do swap
```

**fuser**

```
#W przeciwieństwie do poprzednich poleceń fuser nie wyświetla listy aktualnie
działających procesów, lecz to jakie procesty aktualnie korzystają z danego pliku
(plik ten może być katalogiem,zwykłym plikiem, pilkiem wykonywalnym, etc.) lub
gniazda.
#Wywołanie bez opcji spowoduje pokazanie pomocy. Aby program działał musimy wskazć
plik
```

```
fuser .
#Powyższe polecenie wskaże nam jakie procesy korzystają z obecnego katalogu. W
wyniku otrzymamy listę id procesów zakończonych literą wskazującą typ dostępu,
może on przyjmować następujące wartości:
#1. c - obecny katalog
#2. e - plik wykonywalny jest uruchomiony
#3. f - otwarty plik (omijany w standardowym wyświetlaniu)
#4. F - plik otwarty do zapisu (omijany w standardowym wyświetlaniu)
#5. r - folder root
#6. m - zmapowany plik lub biblioteka współdzielona
#Przykładowe wyjście
/home/mtracewicz: 1490c  1491c  1493c  1496c  1528c  1535c  1601c  1631c  1641c
1646c  1647c  1652c
1655c  1659c  1660c  1662c  1665c  1681c  1685c  1689c  1695c  1704c  1708c  1718c
1723c  1728c
1733c  1738c  1745c  1746c  1748c  1752c  1754c  1756c  1759c  1760c  1764c  1766c
1772c  1783c
1790c  1796c  1799c  1805c  1807c  1836c  1874c  1886c  1934c  1975c  1982c  1983c
1987c  1991c
1999c  2015c  2110c  2132c  2151c
#Możemy użyć flagi -u pokaże nam użytkownika do którego należy dany proces
/home/mtracewicz:     1490c(mtracewicz)  1491c(mtracewicz)  1493c(mtracewicz)  ...
#Możemy także użyć opcji -v aby pokazać rozbudowane wyjście
                     USER        PID ACCESS COMMAND
/home/mtracewicz:    mtracewicz   1490 ..c.. dbus-broker-lau
                                  mtracewicz   2151 ..c.. zsh
```

**lsof**

```
#Polecenie służące do wyświetlania listy otwartych plików
lsof
#Możemy użyć opcji "u" aby określić użytkownika dla którego chcemy wyświetlić
otwarte pliki
lsof -u mtracewicz
#Przykładowe wyjście
COMMAND PID       USER  FD TYPE DEVICE    SIZE             NODE NAME
bash    943 mtracewicz cwd  DIR    0,2    4096  844424930426347 /home/mtracewicz
#--1--|--2--|---3-----|--4-|---5---|--6---|---7--|-------8-------|--------9----|
#1. polecenie, które owtorzyło plik
#2. id procesu, które otworzyło ten plik
#3. użytkownik, który go otworzył
#4. deskryptor pliku: cwd - obecny folder, rtd - folder root, txt - tekst
programu, mem -plik zmapowany do pamięci, (tutaj jest numer)r - deskryptor do
czytania, (tutaj jest numer)w - deskryptor do pisania,(tutaj jest numer)u -
deskryptor do czytania i pisania
#5.typ pliku np.: DIR - folder, REG - plik zykły etc.
#6. urządzenie
#7. rozmiar pliku
#8. węzeł
#9. nazwa pliku
#Możemy np. wyświetlić pliki używające tcp na porcie 80
```

```
lsof -i TCP:80
#Lsof jest bardzo potężnym narzędziem posiadającym wiele opcji. Wszystkie są
dokładnie opisane w
man lsof
```

**Limitowanie zasobów systemowych dla użytkownika**

W systemie Linux możemy użyć polecenia ulimit, aby nakładać limit na zasoby systemowe.

```
#Możemy wyświetlić obecne limity dla zwykłego użytkownika użyjemy opcji -a
ulimit -a
#Przykładowe wyjście
-t: cpu time (seconds)              unlimited
-f: file size (blocks)              unlimited
-d: data seg size (kbytes)          unlimited
-s: stack size (kbytes)             8192
-c: core file size (blocks)         unlimited
-m: resident set size (kbytes)      unlimited
-u: processes                       19678
-n: file descriptors                1024
-l: locked-in-memory size (kbytes)  64
-v: address space (kbytes)          unlimited
-x: file locks                      unlimited
-i: pending signals                 19678
-q: bytes in POSIX msg queues       819200
-e: max nice                        0
-r: max rt priority                 0
-N 15:                              unlimited
#W tym wyjściu widzimy też możliwe dla nas opcje. Widzimy np. opcję "-u", która
pozwala nam zmienić limit procesów. Jeżeli użyjemy którejś z tych opcji bez
wpisania wartości, wyświetli on aktualny miękki limit.
ulimit -u
19678
#W systemie rozróżniamy dwa typy limitów:
# miękki - jest on pilnowany przez jądro systemu
# twardy - służy on za górną wartość dla limitu miękkiego
#Teraz ustawiamy limit( nie podając opcji "S" lub "H" ustawimy naraz oba
limity,miękki i twardy)
ulimit -u 50
#Opcja "S" pozwala ustawić limit miękki
ulimit -Su 50
#A "H" twardy( ważne jest aby "H"/"S" znajdowały się przed inną opcją jak "u"
inaczej zamiast ustawić nowy limit, wyświetlimy odpowiedni limit a wartość którą
chcemy nadać zostanie zignorowana)
ulimit -uS 50
#Powyższe polecenie wyświetli miękki limit dla procesów dla aktualnego użytkownika
```

Jeżeli chcemy ustawić dla konkretnych użytkowników musimy to zrobić w pliku /etc/security/limits.conf.

```
#Wpis w tym pliku ma format:
#<domain>      <type>  <item>         <value>
#Gdzie domian to użytkownik/grupa
#Type to "soft"/"hard" (wyjaśnione w poprzednim przykładzie)
#Item to zasobób (lista zasobów jest widoczna w poprzednim przykładzie jako lista
opcji)
#Value wskazuje na ile ustawiamy limit
#Przykładowy wpis
mtracewicz soft nproc  50
```

# Quoty

Qouty są to ograniczenia miejsca na dysku jakie może zajmować konkretny użytkownik/grupa. Limity qouty możemy podzielić na dwa rodzaje względem typu limitu, które nakładają:

- miękkie - miejsce nie jest rezerwowane w momencie jego ustawienia. Pozwala to na uzyskanie większego wykorzystania dysków (np. jeżeli będziemy w pierwszym dniu używać dysk 100 gb to użytkownicy mogą z niego korzystać mimo, że suma limitów quoty jest ustawiony na 500 gb a następnie zmienić dysk na większy gdy zajdzie taka potrzeba. W wypadku zaniedbania administratora może jednak dojść do sytuacji gdy miejsce na dysku się skończy a limit quoty nie zostanie osiągnięty). Ustawienie tego limitu może jednak prowadzić do sytuacji, w której użytkownicy użyją więcej miejsca niż mają przydzielone w tym typie quoty co musimy rozwiązać przez ustawienie "grace period" więcej o tym później .
- twarde - w momencie ustawienia quoty, system zarezerwuje miejsce na dysku natychmiast i nie pozwoli przekroczyć ustawionego limitu.

Oraz na dwa rodzaje względem tego na co nakładamy limit

- przestrzeń dyskowa (ilość danych na dysku)
- węzły (przechowują metadane o plikach, możemy to interpretować jako dopuszczalna ilość plików )

Aby móc ich używać musimy najpierw włączyć je dla konkretnego systemu plików. Możemy to zrobić edytując plik konfiguracyjny: /etc/fstab. Musimy dodać do opcji usrquota i/lub grpquota.

```
#Przykład z dokumentacji systemu Fedora
/dev/VolGroup00/LogVol02 /home     ext3    defaults,usrquota,grpquota  1 2
#--------------1--------|--2----|---3----|-----------------4----------|5|6|
#1. nazwa użądzenia
#2. system plików
#3. format plików
#4. opcje
#5. wsparcie dump (0 - wyłączone,1 - włączone)
#6. czy powinen być autoskanowany przy montowaniu (0 - nie, 1 - tak, 2 - tak dla
wszystkich partycji nie będących partycją root)
```

Następnie musimy odpiąć urządzenie i ponownie je podłączyć, aby zmiany zostały wprowadzone. Należy to zrobić resetując system lub używając polecenia:

```
#W miejscu "/home" powinniśmy wstawić nasz system plików. W tym przykładzie
zastosujemy "/home" dla spójności z poprzednim przykładem
mount -o remount /home
```

W kolejnym kroku musimy utworzyć plik bazy danych dla quoty. Użyjemy do tego polecenia:

```
#Sytuacja z "/home" jak wyżej. Opcje:
# "c" - stwórz plik
# "u" - aquota.user
# "g" - aquota.group
quotacheck -cug /home
#Wywołanie polecenia tylko z opcją "c" zaskutkuje utworzeniem tylko pliku quota
dla użytkownika.
```

Następnie wprowadzimy do naszej bazy dane o obecnym zapełnieniu wszystkich dysków zamontowanych lokalnie z włączoną qoutą.

```
#Opcja "a" oznacza wszystkie lokalne systemy plików, opcja "v" odpowiada za
bardziej informacyjny format wyjścia programu, pozostałe opcje jak w poprzednich
przykładach
quotacheck -avug
```

Zwykle zalecane jest aby quotacheck było wykonywane gdy system plików nie jest w użyciu a działanie quoty jest wyłączone. Jeżeli nie zastosujemy się do tego zalecenia możemy doprowadzić do zepsucia pliku quoty. W wypadku gdyby jednak plik quoty został zepsuty, program wykonując polecenie quotacheck wejdzie w tryb interaktywny i będzie próbował go naprawić. Inaczej będzie gdy użyjemy opcji "n". W takim wypadku jeżeli w zepsutym pliku pojawi się dwa razy wpis dla tego samego użytkownika/grupy, program sam automatycznie wybierze pierwszy z nich. Dodatkowo możemy użyć opcji "F" i wskazać format plików quoty aby zapobiec jego autodetekcji.

Teraz możemy już nałożyć ograniczenia na konkretnego użytkownika. Aby to zrobić możemy użyć polecenia:

```
#Polecenie to uruchomi nasz domyślny edytor (ustawiony w zmiennej środowiskowej
"EDITOR")
edquota testUser
#Powinno nam wyskoczyć coś podobnego do:
Disk quotas for user testuser (uid 501):
Filesystem                blocks     soft     hard   inodes   soft   hard
/dev/VolGroup00/LogVol02  440436   500000   550000    37418      0      0
#-----------1------------|--2-----|----3---|---4---|----5----|---6---|--7--|
#1. nazwa użądzenia
#2. pamięć obecnie zajmowana przez użytkownika
#3. miękki limit na pamięć
#4. twardy limit na pamięć
#5. ilość węzłów
```

```
#6. miękki limit na węzły
#7. twardy limit na węzły
#0 w dowolnej z kategorii 3/4/6/7 oznacza, że urzytkownik nie ma narzuconego tego
limitu
#Powyższe dane pochodzą z przykładu z dokumentacji systemu Fedora
```

Możemy także ustawić limit dla grupy. Aby to zrobić możemy użyć polecenia:

```
#Ponownie powinno uruchmić nam domyślny edytor
edquota -g devel
#Przykładowy plik do edycji (Dokumentacja Fedory)
Disk quotas for group devel (gid 505):
Filesystem                blocks    soft     hard    inodes    soft    hard
/dev/VolGroup00/LogVol02  440400       0        0     37418       0       0
#Widzimy, że składnia jest dokładnie taka sama jak ta w pliku dla użytkownika
```

Kolejnym korkiem może być ustawienie tak zwanego "grace period". Nie jest to krok obowiązkowy ponieważ domyślnie wartość ta wynosi siedem dni. "Grace period" jest to okres, w którym po przekroczeniu miękkiego limitu quoty użytkownik nadal ma możliwość zapisu plików (jeżeli w tym czasie osiągniemy limit twardy quoty to i tak nie będziemy w stanie pisać). Aby ustawić ten okres użyjemy polecenia:

```
edquota -t
#Taki plik pokaże nam się do edycji w domyślnym edytorze
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/mapper/system-tmp                 5days                  5days
#W tym pliku mamy kolejno od lewej system plików,"grace period" na pamięć, "grace
period" na węzły
```

Aby włączyć/wyłączyć quoty na systemy plików na których były one już aktywowane, możemy używać poleceń:

```
#Listujemy akualne stany wszystkich quot
quotaon -ap
#Uruchamiamy quoty dla użytkownika
quotaon -u /home/
#Uruchamiamy quoty na grupy
quotaon -g /home/
#Wyłączamy quoty dla grup i użytkowników
quotaoff -ug /home/
```

Możemy wyświetlić raport o aktualnym stanie quot za pomocą polecenia:

```
#Dla wszystkich użytkowników
repquota -a
#Dla konkretnego użytkownika
repquota -u testUser
#Możemy dodać opcję "v" aby wyświetlić więcej informacji
repquota -av
```

# Bibliografia


**Polecenie last**

- https://www.golinuxhub.com/2014/05/how-to-check-last-login-time-for-users.html
- man last

**Polecenie users**

- man users

**Polecenie SU**

- man su

**Polecenie Sudo**

- https://www.lifewire.com/what-to-know-sudo-command-3576779
- https://www.ixsystems.com/blog/best-practices-in-unix-access-control-with-sudo/
- https://stelfox.net/blog/2016/02/better-practices-with-sudo/

**Polecenie id**

- man id

**Dostęp do plików**

- http://www.penguintutor.com/linux/file-permissions-reference
- http://mediologia.pl/katalogi-i-pliki-linux/2-4-atrybuty-plikow-uzywanych-w-systemie-linux-polecenie-ls
- https://www.hostingadvice.com/how-to/change-file-ownershipgroups-linux/
- https://linuxconfig.org/identifying-file-types-in-linux
- https://www.geeksforgeeks.org/access-control-listsacl-linux/

**Pliki z informacjami o użytkownikach/grupach**

- https://www.cyberciti.biz/faq/understanding-etcgroup-file/
- http://www.yourownlinux.com/2015/07/etc-passwd-file-format-in-linux-explained.html

**Hasła użytkowników**

- https://www.cyberciti.biz/faq/understanding-etcshadow-file/
- https://www.howtogeek.com/195430/how-to-create-a-strong-password-and-remember-it/

- https://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils\
- https://blog.jscrambler.com/hashing-algorithms/
- https://www.cyberciti.biz/faq/rhel-centos-fedora-linux-upgrading-password-hashing/
- https://www.usenix.org/legacy/events/usenix99/provos/provos_html/node4.html
- https://en.wikipedia.org/wiki/SHA-2

**Tworzenie, usuwanie i modyfikacja kont użytkowników**

- https://www.lifewire.com/create-users-useradd-command-3572157
- https://www.linuxnix.com/delete-user-account-linux/
- https://www.itzgeek.com/how-tos/linux/how-to-modify-user-accounts-in-linux-using-usermod-command.html

**Blokowanie użytkowników**

- https://www.linuxnix.com/lock-user-account-linux/
- https://www.2daygeek.com/lock-unlock-disable-enable-user-account-linux/

**Procesy**

- https://linux.101hacks.com/unix/fuser/
- https://linux.101hacks.com/unix/top/
- https://linux.101hacks.com/monitoring-performance/ps-command-examples/

**Zasoby**

- https://ss64.com/bash/ulimit.html
- https://ss64.com/bash/limits.conf.html
- https://www.networkworld.com/article/2693414/setting-limits-with-ulimit.html
- man lsof
- https://www.tecmint.com/10-lsof-command-examples-in-linux/

**Quoty**

- https://www.linux.com/tutorials/step-step-using-user-quotas-linux/
- https://www.looklinux.com/how-to-manage-disk-quota-in-linux/
- https://docs.fedoraproject.org/en-US/Fedora/14/html/Storage_Administration_Guide/ch-disk-quotas.html
- https://www.howtoforge.com/tutorial/linux-quota-ubuntu-debian/
- https://www.itworld.com/article/2811509/storage-quotas---hard-vs--soft---explained.html
- https://en.wikipedia.org/wiki/Disk_quota#Common_Unix_disk_quota_utilities
- https://www.itworld.com/article/2811509/storage-quotas---hard-vs--soft---explained.html
- https://www.golinuxhub.com/2018/08/step-by-step-guide-implement-quota-edquota-grace-period-linux.html#AddSetGracePeriod

**Linux**

- https://en.wikipedia.org/wiki/Linux
- The Complete History of Linux (Abridged) -Bryan Lunduke