Lab 2 - kontrola uprawnień w systemie Linux. Zarządzanie użytkownikami i grupami.

1. Uprawnienia w systemie Linux

Podstawowy opis uprawnień został już przedstawiony w lab 1, ale warto je przypomnieć. Każdy zasób jest opisany za pomocą uprawnień `rwx`, które odpowiednio oznaczają:

- `r - read` - odczyt,
- `w - write` - zapis,
- `x - execute` - wykonanie, w przypadku folderów przeszukiwanie.

Te uprawnienia są określone dla jednego z trzech bytów w systemie operacyjnym:

- właściciel (ang. `user`),
- grupa (ang. `group`),
- pozostali (ang. `others`).

Polecenie `chmod`

> Manual: http://manpages.ubuntu.com/manpages/focal/pl/man1/chmod.1.html

Uprawnienia zmieniamy poleceniem `chmod` (ang. change modifiers), które może pracować w różnych trybach. Na początek omówimy tryb, który jest nawiązuję do zapisu uprawnień w postaci `rwx`.

Przykład:

```
chmod g+r plik.txt
```

W powyższym przykładzie do zasobu `plik.txt` zostanie dodane uprawnienie do odczytu dla grupy, która aktualnie jest przypisana do tego zasobu.

Każdy z trzech bytów opisany jest stosowną literą (+ opcja obejmująca wszystkie trzy jednocześnie):

- `u - user - właściciel`
- `g - group - grupa`
- `o - others - pozostali`
- `a - all - wszyscy`

Dodatkowo mamy do dyspozycji trzy operatory: `+`, `-` oraz `=`, którymi możemy ustawiać odpowiednie uprawnienia. Uprawnienia możemy nadawać dla wielu bytów jednocześnie:

```
chmod go+rw plik.txt
chmod g=rw,o=x plik.txt
```

Ćwiczenia

1. W swoim folderze domowym utwórz nowy folder o nazwie `secret`.
2. Za pomocą polecenia `chmod` zmień uprawienia dla folderu `secret` tak, aby tylko właściciel miał dostęp do jego zawartości.
3. Zmień teraz uprawnienia tak, aby właściciel miał pełne uprawnienia, grupa tylko odczyt a pozostali brak uprawnień.

Ćwiczenie numer 3 wymagało dość sporo zachodu, gdyż dla każdego bytu uprawnienia musimy opisywać oddzielnie. Warto zatem poznać inną postać polecenia `chmod`, postać numeryczną.

Przykład:

```
chmod 755 plik.txt
```

Ten tryb pozwala jednym poleceniem zdefiniować uprawnienia dla wszystkich trzech bytów. Odpowiednio w powyższym przypadku poczynając od lewej strony, 7 dla właściciela, 5 dla grupy i 5 dla pozostałych. Znaczenie tych wartości przestawiono w tabeli poniżej.

Cyfra	Prawa	Litera	Binarnie
0	Brak praw	---	000
1	Wykonywanie	--x	001
2	Zapis	-w-	010
3	Zapis i wykonanie	-wx	011
4	Odczyt	r--	100
5	Odczyt i wykonanie	r-x	101
6	Odczyt i zapis	rw-	110
7	Odczyt, zapis i wykonanie	rwx	111

Tryb numeryczny jest używany dużo powszechniej niż tryb znakowy. Nierzadko zdarzają się sytuacje gdzie należy zdefiniować uprawnienia do zasobu z identycznymi uprawnieniami jak w już istniejącym zasobie i możemy wtedy skorzystać z opcji polecenia `chmod`, która skopiuje uprawnienia ze wskazanego zasobu:

```
chmod --reference=folder/plik inny_folder_plik
```

Podobnie jak przy wielu innych poznanych już poleceniach opcja `-R` służy do rekurencyjnego nadawania tych uprawnień w głąb struktury systemu plików:

```
chmod -R 755 test3
chmod -R --reference=test2 test2/
```

Ćwiczenia

1. Ponownie zmień uprawnienia dla folderu `secret` kopiując uprawnienia ze swojego folderu domowego.
2. Utwórz nowy folder w folderze secret. Wykonaj polecenie `chmod`, które zmieni uprawnienia na rwx r-x r-x dla folderu secret i wszystkich zasobów podrzędnych.

Uprawnienia specjalne

Oprócz omówionych podstawowych uprawnień istnieją również uprawnienia specjalne, które możemy z pomocą polecenia chmod zmodyfikować. Mowa tutaj o uprawnieniach `setuid`, `setgid` oraz tzw. `sticky bit` (bit lepkości).

Flaga `setuid` pozwala na uruchomienie pliku wykonywalnego przez zwykłych użytkowników z uprawnieniami jego właściciela. Przykładem mogą być tutaj polecenia `su` oraz `sudo`.

```
ls -l /bin
...
-rwsr-xr-x  1 root root       67816 lip 21 09:49  su*
-rwsr-xr-x  1 root root      166056 lip 15 02:17  sudo*
...
```

Jak widać zamiast uprawnienia `x` dla właściciela widnieje tu litera `s`, która oznacza, że flaga setuid jest dla tych plików ustawiona. Zwykły użytkownik wykonujący ten plik uruchomi go jako użytkownik `root`. To uprawnienie należy przydzielać bardzo ostrożnie i tylko wtedy kiedy wiemy co robimy. Może to dać użytkownikom możliwość dostępu do ustawień systemu, których nie planowaliśmy udostępniać. Pozwala również na delegowanie części zadań na innych użytkowników, którzy nie posiadają konta superużytkownika, ale dzięki temu mechanizmowi mogą wykonywać część poleceń jak inny użytkownik (z reguły z wyższymi uprawnieniami).

Przeglądając manual polecenia `chmod` natkiemy się na linię `Każdy TRYB ma postać [ugoa]*([-+=]([rwxXst]*|[ugo]))+|[-+=][0-7]+` gdzie widać, że poszczególne specjalne uprawnienia są nadawane z wykorzystaniem ogólnej postaci polecenia `chmod`.

Polecenie

```
chmod u+s moj_plik.sh
```

nada pozostałym użytkownikom prawo do wykonania pliku jak jego właściciel. Bit ten ustawiamy na poziomie uprawnień właściciela.

Jak to się ma jednak do uprawnień określanych metodą numeryczną ? Te specjalne uprawnienia będą opisane za pomocą dodatkowej wartości poprzedzającej dotychczasowe trzy cyfry.

setuid	4	set user identifier bit	bit identyfikatora użytkownika

setgid	2	set group identifier bit	bit identyfikatora grupy
sticky	1	sticky bit	klejący bit

Przykład:

```
chmod 4755 moj_plik.sh
```

Bit `setgid` ustawiany jest na poziomie grupy i możemy go ustawić dla plików wykonywalnych oraz na katalogach. Z ustawionym bitem `setgid` plik jest wykonywany przez użytkowników, którzy nie są jego właścicielami, z przywilejami użytkowników należących do grupy. Przykładem może być polecenie `wall`:

```
ls -l /bin
...
-rwxr-sr-x  1 root tty        35048 lip 21 09:49  wall*
...
```

Ustawienie `setgid` dla katalogu powoduje automatyczne dziedziczenie przypisanej grupy do nowych plików i katalogów tworzonych w ramach tej struktury.

```
cd ~
mkdir dswp
sudo chmod g+s dswp
ls -la
drwxrwsr-x  2 kropiak   dswp    4096 lis 20 09:38 .
```

Lepki bit (ang. `sticky bit`) ustawiany jest w katalogach dostępnych publicznie aby zabezpieczyć pliki i podkatalogi należące do zwykłych użytkowników przed skasowaniem lub przeniesieniem przez innych użytkowników. Przykładem takiego katalogu może być katalog `/tmp`.

```
ls -l /
drwxrwxrwt  20 root root       4096 lis 20 10:06 tmp
```

Sticky bit ustawmiamy dla uprawnień dotyczących pozostałych użytkowników.

```
chmod o+t /folder
```

Zgodnie z manualem istnieje również możliwość określenia atrybutu uprawnień przez wielką literę `X`. Dzięki tej wartości możemy ustawić rekurencyjnie atrybut `x` dla wszystkich folderów podrzędnych, ale z pominięciem plików.

```
chmod -R u=rwX,g=rX,o=rX testdir/
```

Ćwiczenia

1. Za pomocą polecenia `find` wyszukaj wszystkie pliki w systemie z ustawionym bitem `setuid`.
2. Sprawdź w dokumentacji sposób użycia polecenia wall i wykonaj je. Usuń bit `setgid` dla polecenia wall i ponownie sprawdź jego działanie.
3. Ponownie ustaw bit `setgid` dla polecenia wall.
4. Dodaj możliwość przeszukiwania wszystkich folderów w Twoim folderze domowym dla członków grupy.

Polecenie `chown`

> Manual: http://manpages.ubuntu.com/manpages/focal/pl/man1/chown.1.html

Aby mieć możliwość zmiany uprawnień do zasobu musimy być jego właścicielem lub posiadać uprawnienia superużytkownika. Możemy również chcieć nadać innym użytkownikom te uprawnienia gdy pracujemy z wieloma innymi użytkownikami na serwerze plików.

Polecenie `chown` pozwala zmienić właściciela i grupę dla zasobu.

Przykład

```
chown jkowalski ~/share/jkowalski
```

Ćwiczenia

1. Wykorzystując polecenie `chown` zmień właściela dla jednego z folderów w swoim folderze domowym na użytkownika root.
2. Sprawdź w dokumentacji polecenia chown jak działa opcja `from` i zmień właściciela i grupę na swojego użytkownika i grupę dla wszystkich zasobów w Twoim folderze domowym, których aktualnym właścicielem jest root.

2. Użytkownicy i grupy w systemie Linux.

Większość istotnych informacji o użytkownikach jest przechowywana w plikach:

- **/etc/passwd** - podstawowe informacje o kontach użytkowników (ew. zaszyfrowane hasło),
- **/etc/group** - podstawowe informacje o grupach użytkowników,
- **/etc/shadow** - rozszerzone informacje o kontach użytkowników (np. daty ważności) i zaszyfrowane hasło (w systemie shadow),
- **/etc/gshadow** - rozszerzone informacje o grupach użytkowników (w systemie shadow).
-

Przykładowa linia z pliku /etc/passwd:

```

```

```
test:x:1001:1001:Jan Testowy,,,:/home/test:/bin/bash
```

Znaczenie wartości poczynając od lewej strony:

- nazwa użytkownika,
- pole hasła, w tym przypadku wartość 'x' oznacza, że informacje i haśle znajdują się w pliku `/etc/shadow`
- ID użytkownika, wartość numeryczna powiązana z użytkownikiem,
- ID grupy, której domyślnym członkiem jest dany użytkownik,
- pełna nazwa użytkownika oraz inne informacje tekstowe, np. adres, telefon o ile zostały zdefiniowane
- katalog domowy użytkownika,
- aplikacja uruchamiana po zalogowaniu się użytkownika, w tym przypadku powłoka (shell), Linia z informacjami o grupie z pliku `/etc/group` wygląda następująco: `test:x:1001:` Analogicznie do wpisów z informacjami o użytkowniku mamy tutaj kolejno dane o nazwie grupy, haśle (również w pliku `/etc/shadow`) oraz ID grupy. Na końcu może się również znajdować lista użytkowników (rozdzielona przecinkiem), którzy należą do danej grupy. Listę grup, do których należy zalogowany użytkownik można sprawdzić poleceniem `groups`.

Najpopularniejsze sposoby tworzenia kont użytkowników to:

- wykorzystanie narzędzia `useradd`,
- wykorzystanie narzędzia `adduser`,
- ręczna edycja plików z definicjami użytkowników (ostrożnie!)

Polecenie `useradd` w minimalnej swojej postaci czyli

```
sudo useradd bolek
```

stworzy nowego użytkownika o nazwie `bolek` (o ile już nie istnieje) z domyślnymi ustawieniami dla ścieżki domowej (`/home/bolek`) oraz powłoką - `/bin/sh`. Natomiast folder domowy nie zostanie utworzony. Jego utworzenie należy wskazać poprzez dodanie opcji `-m` do wywołania polecenia `useradd`.

Inną istona rzeczą, na którą należy zwrócić uwagę jest fakt przechowywania i zarządzania wartościami domyślnymi polecenia `useradd`. Wywołanie polecenia:

```
useradd -D
```

wyświetli jego domyślne ustawienia, np.:

```
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
```

```
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Próba zmiany domyślnych ustawień bez `sudo` się nie powiedzie:

```
useradd -D -s /bin/bash
# wyświetli
useradd: nie można utworzyć nowego pliku z ustawieniami domyślnymi
```

Spróbujmy więc z `sudo`:

```
sudo useradd -D -s /bin/bash
# i teraz
useradd -D
# wyświetli
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Dziwi brak wartości dla `SHELL`, ale gdy wyświetlimy te ustawienia również używając `sudo`:

```
udo useradd -D
# wyświetli
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

Pozostaje nam przetestować czy to ustawienie działa poprawnie:

```
sudo useradd testowy
# sprawdzamy ostatnią linią w pliku /etc/passwd
tail -1 /etc/passwd
# i otrzymamy
testowy:x:1008:1010::/home/testowy:/bin/bash
```

Zmiana wartości domyślnej dla powłoki przyniosła pożądany efekt.

Polecenie `adduser` w domyślnej postaci działa nieco inaczej:

```
sudo adduser lolek
# wyświetli
odawanie użytkownika "lolek"...
Dodawanie nowej grupy "lolek" (1008)...
Dodawanie nowego użytkownika "lolek" (1007) w grupie "lolek"...
Tworzenie katalogu domowego "/home/lolek"...
Kopiowanie plików z "/etc/skel" ...
Nowe hasło :
Proszę ponownie wpisać nowe hasło :
passwd: hasło zostało zmienione
Zmieniam informację o użytkowniku lolek
Wpisz nową wartość lub wciśnij ENTER by przyjąć wartość domyślną
    Imię i nazwisko []:
    Numer pokoju []:
    Telefon do pracy []:
    Telefon domowy []:
    Inne []:
Czy informacja jest poprawna? [T/n]
```

To polecenie w podstawowej formie działa jak kreator pozwalający przejść kolejne kroki tworzenia konta użytkownika informując nas również o kolejnych wykonanych czynnościach.

Inną postacią tego polecenia jest postać:

```
sudo adduser lolek marketing
```

które pozwala dodać istniejącego użytkownika do istniejącej grupy. W tym przypadku użytkownik `lolek` zostanie dodany do grupy `marketing`.

Istniejące konto można modyfikować na kilka sposobów:

- polecenie `chfn` zmienia informacje GECOS (imię, nazwisko, itp.) o użytkowniku,
- polecenie `chsh` zmienia powłokę,
- narzędzie `usermod` modyfikuje dowolne parametry konta,
- narzędzie `groupmod` j.w. dla grupy,
- polecenie `passwd` zakłada nowe hasło, a w systemie shadow passwords zmienia daty ważności konta.

Usuwanie kont i grup:

- konto użytkownika można usunąć przy pomocy polecenia `userdel`,
- powyższe polecenie z opcją `-r` usuwa katalog domowy użytkownika,
- polecenie `groupdel` usuwa grupy użytkowników,
- przy pomocy polecenia instrukcji `find` można odnaleźć i usunąć pliki których właścicielem jest podany użytkownik lub grupa.

> Polecenie `userdel` domyślnie nie usuwa folderu domowego usuwanego uzytkownika, nalezy sprawdzić w maunalu opcję, która doda tę operację to procesu usuwania konta użytkownika.

Blokowanie dostępu do konta:

- poleceniem `passwd` z opcją `-l`,
- przez ręczną modyfikację hasła w pliku `passwd/shadow`,
- poprzez zmianę powłoki użytkownika na program nie dopuszczający logowania.

W zależności od narzędzia system przyjmie pewne wartości domyślne w przypadku tworzenia użytkowników np. wybrana powłoka, domyślny folder domowy. Część informacji domyślnych dla całego systemu znajduje się w pliku `/etc/profile`, ale znacznie więcej znajdziemy w `/etc/login.defs`.

Jeżeli korzystamy z narzędzia adduser warto zobaczyć jakie są jego domyślne ustawienia, które znajdziemy w `/etc/adduser.conf`. Dodatkowo podczas tworzenia konta przy pomocy `adduser` tworzony jest folder użytkownika oraz pewna struktura folderów wewnątrz. Istnieje możliwość wpływania na to jak ta struktura ma wyglądać – zobacz `/etc/skel`.

Po utworzeniu konta użytkownika informacje o jego ustawieniach można znaleźć w jego folderze domowym. Jeżeli nie znamy nazw plików, które te informacje przechowują musimy wyświetlić zawartość folderu `~` wraz z ukrytymi plikami.

Zadania

1. Korzystając z polecenia `useradd`:
   1.1 Dodaj nowego użytkownika o nazwie test1 bez dodatkowych parametrów, sprawdź ustawienia dla tego użytkownika w pliku `/etc/passwd`, `/etc/group` oraz `/etc/shadow`, spróbuj zalogować się na konto tego użytkownika. Czy pojawiły się jakieś problemy i jak je rozwiązać ?
   1.2 Dodaj nowego użytkownika o nazwie test2 a w opcjach dodatkowych ustaw folder domowy oraz powłokę ze wskazaniem na `/bin/bash`,
   1.3 dodaj nowego użytkownika definiując w linii poleceń UID o wartości 2000, sprawdź zapis w pliku `/etc/passwd`,
   1.4 Dodaj użytkownika test3 z ważnością konta na miesiąc do przodu, ponownie sprawdź pliki `/etc/passwd` oraz `/etc/group`. Zwróć uwagę na sposób przyznawania numerów UID przez narzędzie useradd,
   1.5 Za pomocą `useradd -D` sprawdź ustawienia domyślne a następnie zmień:
   - domyślną grupę (może być jedna z utworzonych wraz z użytkownikami z poprzedniego ćwiczenia)
   - domyślny katalog domowy
   - domyślną powłokę na /bin/bash

2. Przy pomocy polecenia `adduser`:
   2.1 Dodaj użytkownika gdzie imię i nazwisko to Jan Kowalski a nazwa użytkownika to jkowalski,
   2.2 Dodaj użytkownika gdzie jego UID to 1111, imię i nazwisko to Marian Maliniak a nazwa użytkownika to mmaliniak. Sprawdź jakie zmiany zaszły w pliku `/etc/passwd` i porównaj mechanizm nadawania UID z narzędziem useradd,
   2.3 Dodaj użytkownika Anna Siębała (asiebala).
   2.4 Poprzez edycję pliku `/etc/group` dodaj nowe grupy – marketing, sprzedaż, bok a następnie użytkowników jkowalski i mmaliniak do grupy marketing, sprzedaż a asiebala do grupy marketing, bok,

2.5 Za pomocą polecenia `groups` sprawdź czy w/w użytkownicy są przypisani do odpowiednich grup,
2.6 Za pomocą polecenia `delgroup` usuń grupy marketing, sprzedaż i bok,
2.7 Za pomocą polecenia `groupadd` dodaj ponownie usunięte grupy z punktu 2.6,
2.8 Za pomocą `adduser` dodaj użytkowników do grup jak w podpunkcie 2.4,
2.9 Przy pomocy polecenia `chfn` zmień imię użytkownika Marian Maliniak na Mateusz Maliniak,
2.10 Jeżeli wcześniej nie zostało to zrobione – dodaj hasła dla użytkowników mmaliniak, jkowalski i asiebala,
2.11 Przy pomocy polecenia `login` zaloguj się na konto jkowalski. Sprawdź czy użytkownik posiada folder domowy i jeżeli tak to przeglądnij zawartość plików `.profile, .bashrc` w jego folderze domowym. Za pomocą polecenia `exit` wyloguj się z tego konta,
2.12 Przy pomocy polecenia `passwd` wymuś zmianę hasła dla użytkownika jkowalski przy kolejnej próbie logowania. Po tej operacji zaloguj się na konto jkowalski następnie wyloguj się,
2.13 Przy pomocy `passwd` zablokuj konto użytkownika jkowalski a następnie spróbuj się zalogować na to konto,
2.14 Odblokuj konto jkowalski poprzez edycję odpowiedniego pliku konfiguracyjnego.

3. Uprawnienia do zasobów.
   3.1 W udziale `/usr/share` utwórz folder `MARKETING` i zezwól użytkownikom z grupy marketing na przeglądanie i odczyt tego folderu,
   3.2 W folderze `MARKETING` dodaj dwa nowe foldery sprzedaż i bok a następnie ustaw uprawnienia tak, aby odpowiednio członkowie grupy sprzedaż i bok mieli możliwość odczytywania, zapisywania, przeglądania (lub wykonywania) plików w tych folderach,
   3.3 Zaloguj się na konto asiebala i sprawdź czy można utworzyć zasób w folderze `MARKETING`, bok i sprzedaż,
   3.4 W folderach odpowiednich grup, np. jkowalski należy do grupy sprzedaż więc w folderze `/usr/share/MARKETING/sprzedaz` utwórz folder jkowalski za pomocą komendy `mkdir` i zmiennej systemowej `USER` (wszystkie zmienne można zobaczyć za pomocą polecenia `printenv`) – należy najpierw zalogować się na konto jkowalski. Utwórz pozostałe dwa foldery (dla mmaliniak i asiebala) w dowolny sposób, ale tak, aby to ci użytkownicy byli ich właścicielami,
   3.5 W każdym folderze utworzonym w punkcie 3.4 utwórz plik `użytkownik_grupy.txt`, np. jkowalski_grupy.txt, którego zawartość to:

- pierwsza linia to nazwa użytkownika
- kolejna linia to lista grup, do której użytkownik należy

Do wykonania ćwiczenia można wykorzystać przekierowanie domyślnego wyjścia do pliku (patrz materiały) oraz polecenie `groups`.