Użytkownicy i grupy w systemie Linux. Zasady udostępniania zasobów s systemie Linux.

1. Podstawowe informacje na temat zasobów w systemie Linux

System Linux w odmienny sposób traktuje zakładanie kont użytkowników niż system Windows. Każdy nowy użytkownik posiada ograniczone prawa do ingerencji w systemie – w pełni może zarządzać jedynie zawartością swojego katalogu domowego. Niektóre katalogi oraz pliki systemowe użytkownicy mogą odczytywać jednak nie mają do nich praw zapisu. Do większości zasobów zaś nie mają żadnego dostępu – te posiada jedynie super administrator (root).

W przeciwieństwie do systemu Windows, w systemie Linux nie istnieje **domyślnie** mechanizm ACL (Access Control List). Wszelkie prawa do plików nadaje się poprzez odpowiednie grupy. Domyślnie użytkownik posiada dostęp jedynie do plików, które sam utworzył. Można jednak jako „drugiego właściciela" ustanowić grupę (użytkowników), która będzie miała określone uprawnienia do zasobu (np. takie jak użytkownik – twórca pliku/folderu). Każdy użytkownik, którego dodamy do takiej grupy będzie automatycznie posiadał uprawnienia ustawione dla tejże grupy. Na końcu do określonego zasobu podajemy „pozostałych", czyli wszystkich użytkowników nie będących właścicielem ani nie należących do uprawnionej grupy.

Po zalogowaniu się na nasz serwer jako użytkownik test i wpisaniu w linii poleceń

ls -al

otrzymamy taką oto odpowiedź (może się ona różnić w zależności od zawartości katalogu domowego)

```
test@server1:~$ ls -al
razem 32
drwxr-xr-x 3 test test 4096 lut  2 12:49 .
drwxr-xr-x 4 root root 4096 sty 28 19:31 ..
-rw------- 1 test test  893 lut  2 16:03 .bash_history
-rw-r--r-- 1 test test  220 sty 28 19:31 .bash_logout
-rw-r--r-- 1 test test 3392 sty 28 19:31 .bashrc
-rw-r--r-- 1 test test    0 lut  2 12:50 key_ok.pub
-rw-r--r-- 1 test test  398 lut  2 12:50 key.pub
-rw-r--r-- 1 test test  675 sty 28 19:31 .profile
drwx------ 2 test test 4096 lut  2 12:51 .ssh
test@server1:~$
```

Nas najbardziej będzie interesować to co znajduje się w pomarańczowej oraz fioletowej ramce.

Pomarańczowa ramka zawiera informacje o prawach dostępu do danego, wymienionego zasobu. Ponieważ na początek wygląda to mało przyjaźnie, poniżej rozpisana zostanie pierwsza linia uprawnień: drwxr-x-r-x

Id zasobu	Bity właściciela	Bity grupy	Bity pozostałych uzytowników
d	rwx	r-x	r-x
	111 (4+2+1 = 7)	101 (4+1 = 5)	101 (401 = 5)
Dany zasób jest katalogiem	Właściciel zasobu posiada do niego pełne prawa – odczytu (r), zapisu (w) oraz wykonania (x)	Grupa użytkowników posiada prawa odczytu (r) oraz wykonania (x); nie posiada prawa zapisywania	Pozostali użytkownicy posiadają prawa odczytu (r) oraz wykonania (x); nie posiadają prawa zapisywania

Jak można wywnioskować system Linux ustawia uprawnienia do zasobów poprzez wykorzystanie 3 bitów na każdego z „uprawnionych" do zasobu. W zależności jak zostaną one ustawione taką liczbę

dziesiętną (a w zasadzie ósemkową) otrzymamy (wartości 0-7). Ponieważ zapis liczbowy jest krótszy to właśnie on wykorzystywany jest najczęściej przy ustawieniach uprawnień.
WAŻNE: Czasami możemy spotkać się z „nietypowymi" uprawnieniami. Prócz wyżej wymienionych (w tym momencie dominujące) w systemie Linux można spotkać:
s/t – bit lepkości
S/T – bit lepkości z wyłączeniem prawa wykonywania wskazanego pliku
Ponadto identyfikatorami zasobów mogą być:
- – oznacza brak jakiegokolwiek identyfikatora (po prostu zwykły plik)
d – zasób jest katalogiem
l – zasób jest dowiązaniem symbolicznym; przeważnie reprezentowany jest przez notację lrwxrwxrwx zwaną jako pusta/nieważna – prawdziwe uprawnienia do tego typu zasobu znajdują się przy źródle dowiązania (pliku/katalogu)
c – oznacza specjalny plik tekstowy
b – oznacza, iż zasób jest urządzeniem blokowym

INFORMACJA: Wskazany użytkownik może należeć do więcej niż jednej grupy użytkowników. Dlatego też możemy nawet przydzielać każdy zasób innej grupie użytkowników co i tak nie wywoła braku dostępu dla określonych użytkowników – wystarczy tych użytkowników dodać do odpowiedniej grupy.

INFORMACJA: W systemach Linux istnieje możliwość nadawania praw zwykłym użytkownikom w taki sposób, by Ci mieli możliwość wykonywania niektórych (bądź wszystkich) poleceń administracyjnych (oraz możliwość przeglądania i modyfikowania określonych/wszystkich plików). W systemie musi zostać zainstalowany specjalny pakiet sudoers, który obecny jest niemal we wszystkich użytkowych systemach Linux (np. Fedora, S.u.S.E, Ubuntu, Mint). Taki pakiet można zainstalować również w systemie Debian. Dzięki temu taki użytkownik nie musi znać hasła administratora, a sam administrator może mieć kontrolę nad wszystkim w systemie i dokładnie wiedzieć kto co robił w systemie. Dodatkowo przydatność sudoers objawia się w chwili gdy chcemy uruchomić jakiś program użytkowy, który wymaga stałego konta użytkownika, nie mającego żadnych praw w systemie (ma odpowiadać tylko za działanie programu).

Fioletowa ramka wskazuje kto jest właścicielem pliku (pierwsza pozycja) oraz grupę uprawnioną do pliku (druga pozycja). Na powyższym zrzucie właściciel jest tożsamy z grupą. Oznacza to, że grupa posiadająca uprawnienia to grupa prywatna właściciela (w której jest przeważnie on sam). Innymi słowy – jeżeli do grupy nie należy nikt poza właścicielem (nikogo nie dodał do swojej prywatnej grupy) to dostęp do pliku może posiadać on sam.

2. Atrybuty plików.

System Linux również dysponuje rozszerzonymi atrybutami plików (odpowiednik zaawansowanych uprawnień w systemie Windows). W przeciwieństwie jednak do Windows, domyślnie atrybuty te są nieużywane (bazuje się na podstawowych ustawieniach). Czasami jednak, szczególnie w przypadku bardzo ważnych danych, może zajść konieczność dodania dodatkowych atrybutów.
W systemie istnieje następujące polecenie pozwalające na zarządzanie wspomnianymi atrybutami:

chattr

Jeżeli chcemy odczytać aktualne ustawienia atrybutów można wykorzystać polecenie:

lsattr

a) chattr

Pozwala na ustawianie bądź usuwanie dodatkowych atrybutów dla plików/katalogów. Właściciel/uprawnieni użytkownicy mogą modyfikować poniższe atrybuty poprzez dodawanie przed nimi znaku '+' (nadaj atrybut) bądź '-' (usuń atrybut). Podane atrybuty nie stanowią pełnego zbioru (wszystkie można zobaczyć wpisując w linie poleceń **man chattr**):
A – odpowiada za nadanie uprawnień do wpisu atime systemowej funkcji stat()
a – odpowiada za nadanie plikowi statusu możliwego do aktualizacji jednak braku uprawnień do zmiany jego aktualnej zawartości. Innymi słowy do pliku można coś dopisać, jednak po zapisaniu nie można tego zmodyfikować
c – pliki/katalogi z tym atrybutem są automatycznie kompresowane przez jądro systemu. Kompresja odbywa się w locie; zapis do takiego pliku powoduje najpierw kompresję zawartości, a dopiero następnie rzeczywisty zapis na dysk
D – atrybut głównie dla katalogów. Jego ustawienie powoduje, że zapis do katalogu następuje synchronicznie (standardowo asynchronicznie)
i – plik/katalog nie może być zmieniany (zawartość i nazwa), nie może zostać także przeniesiony, skopiowany czy usunięty. Zabezpieczenie to działa także na administratora systemu! (najpierw musi zostać ściągnięte by można coś zrobić z tak zabezpieczonym plikiem)
s – jeżeli plik/katalog z tym atrybutem zostanie usunięty, system plików dodatkowo zapisze przestrzeń (bloki) po nim samymi zerami
u – plik z tym parametrem można skasować, jednak zostanie on zachowany w systemie (zostanie jedynie skasowane dowiązanie do niego). Administrator nadal będzie mógł przywrócić plik.

Ponadto sam program posiada następujące parametry:
R – w przypadku katalogu narzędzie nada/usunie wskazane atrybuty także plikom i katalogom znajdującym się wewnątrz tego katalogu (tzw. rekursywność)
V – narzędzie podczas pracy ma informować co aktualnie wykonuje (i jaki jest tego skutek); bez tego parametru program wykonuje swoje zadanie w tle
f – ignoruje większość błędów (próbuje wymusić poprawne wykonanie polecenia)

Przykłady wywołania:

chattr -R +au Ważne_Dokumenty – nadaje katalogowi oraz jego zawartości jedynie możliwość dopisywania nowej zawartości oraz chroni je przed przypadkowym usunięciem

chattr -RV +D katalog – nada katalogowi oraz katalogom w nim się znajdującym atrybut synchronizacji przy zapisie zawartości

chattr +i-s wazny_plik – nadaje atrybut niezmienności zawartości pliku oraz zdejmuje opcję bezpiecznego usuwania

b) lsattr

To polecenie ma za zadanie wyświetlenie aktualnie ustawionych atrybutów dla wskazanego pliku/katalogu (bądź plików)

Przykłady użycia:

lsattr -a – wyświetli wszystkie aktualne ustawienia atrybutów dla plików i katalogów znajdujących się w aktualnie użytkowanym katalogu

lsattr – jak poprzednio jednak pominięte zostaną katalogi ukryte bądź puste
lsattr -aR – jak pierwsze polecenie, lecz wyświetla atrybuty także katalogów/podkatalogów oraz plików w nich się znajdujących

3. Tworzenie kont użytkowników oraz grup. Zarządzanie użytkownikami/grupami

1) Każda dystrybucja systemu Linux pozwala na tworzenie nowego konta użytkownika poprzez komendę:

useradd

Polecenie to posiada kila opcji. Ich pełna lista znajduje się w podręczniku systemowym (**man useradd**) oraz w sieci Internet. Poniżej najważniejsze z nich:
-b <katalog> – ustawia podany katalog jako podstawowy do utworzenia katalogu domowego użytkownika (takim katalogiem w systemie jest domyślnie /home); jeżeli nie podamy wartości <katalog> to zostanie wybrana wartość spod zmiennej systemowej BASE_DIR; możemy nie podawać tego parametru (system domyślnie pobiera systemową ścieżkę katalogu podstawowego)
-d <katalog> – ustawia wskazany katalog jako domowy użytkownika; przykładowo gdy za <katalog> podstawimy wartość 'wspolny' to tworzony użytkownika otrzyma taki katalog jako domowy we wcześniej wskazanym katalogu bazowym (czyli np. /home/wspolny). Jeżeli nie podamy wartości <katalog> to nazwa katalogu domowego będzie tożsama nazwie użytkownika systemu; BEZ PODANIA TEJ OPCJI UŻYTKOWNIK NIE BĘDZIE POSIADAŁ KATALOGU DOMOWEGO!
-m – opcja podobna do poprzedniej (tworzy katalog domowy) jednak tylko w wypadku, gdy takowy katalog nie istnieje; ponadto do katalogu zostają skopiowane wszystkie pliki i foldery określone w systemie jako szkielet konta (domyślnie znajdują się w /etc/skel)
-k – pozwala na określenie katalogu szkieletowego dla tworzonego użytkownika (z którego zostanie przekopiowana zawartość do nowego katalogu domowego). Nie podanie tej opcji spowoduje wykorzystanie katalogu /etc/skel
-g <nazwa_grupy> – nazwa podstawowej grupy użytkowników, do której będzie należał tworzony użytkownik (GRUPA MUSI ISTNIEĆ!). Jeżeli parametr ten zostanie pominięty to system (w domyślnej konfiguracji) utworzy nową grupę o nazwie takiej jak nazwa użytkownika; UWAGA: w niektórych systemach brak tej opcji spowoduje, że użytkownik zostanie dodany do domyślnej grupy z ID 100
-G <nazwa_grupy>,<nazwa_grupy2>,...,<nazwa_grupyN> – dodaje użytkownika do wszystkich (wymienionych po przecinku) grup jako grup dodatkowych. Nie ma to wpływu na grupę podstawową
-U – tworzy nową grupę użytkowników o nazwie identycznej do nazwy użytkownika, po czym dodaje tegoż użytkownika do niej (tzw. grupa prywatna)
-s – pozwala na wybranie domyślnej powłoki systemowej dla użytkownika (domyślnie wybierana jest /bin/sh)

Pozostaje jeszcze ustawienie domyślnego hasła dla użytkownika. W systemach z rodziny Windows po prostu wpisuje się hasło jako parametr. W przypadku systemu Linux hasło musi zostać zaszyfrowane prostą metodą systemową **crypt()**. Dodatkowo jej wynik musi zostać podstawiony do parametry **-p**. Przykłady użycia useradd:

useradd -m -p $(perl -e'print crypt("tajnehaslo", "aa")') -g users testowy – dodaje do systemu użytkownika, do grupy users, z hasłem tajnehaslo, z katalogiem domowym tożsamym z jego nazwą (domyślnie będzie to /home/testowy)

useradd -b /home/users -d moders -U mod2 – tworzy użytkownika mod2, którego katalog

domowy został określony na moders, który z kolei ma zostać utworzony w /home/users (czyli /home/users/moders)

useradd blad – doda konto użytkownika blad bez jakichkolwiek ustawień. Spowoduje to niezdolność użytkowania takiego konta do momentu zmiany jego parametrów

2) Jeżeli chcemy modyfikować konta użytkowników to używamy polecenia:

usermod

W zasadzie polecenie ma podobne (za wyjątkiem kilku opcji) parametry do useradd. Oto najważniejsze z nich:
-g <grupa> - zmienia grupę podstawową użytkownika na <grupa>. Użytkownik zostaje usunięty z poprzedniej grupy podstawowej. Wszystkie pliki i katalogi w katalogu domowym użytkownika będą posiadały zmienione prawa własności grupy na nową grupę podstawową użytkownika. W katalogach poza domowym użytkownik sam musi zmienić przynależność plików do nowej grupy.
-G <grupa1>,<grupa2>,...,<grupaN> - polecenie dodaje użytkownika do nowej (nowych) grup dodatkowych. Jeżeli użytkownik należał do innej grupy/grup dodatkowych, niewymienionych aktualnie przy parametrze, zostanie z nich usunięty chyba, że zostanie dodatkowo użyty parametr -a
-a – parametr stosuje się WYŁĄCZNIE w parze z -G. Powoduje on podtrzymanie przynależności użytkownika do wszystkich grup dodatkowych, do jakich został on dodany (następuje realne dodanie do nowej grupy dodatkowej, a nie zmiana grup dodatkowych)
-L – blokuje hasło użytkownika (nie ma możliwości zalogowania się na konto, ponieważ użytkownik nie może podać poprawnego hasła (będzie ono odrzucane przez system).
-U – odblokowuje hasło użytkownika.
WAŻNE! Obie powyższe opcje nie blokują konta użytkownika! W celu zablokowania konta należy przestawić jego ważność na 1.
-l – ustawia dla użytkownika nowy login. Oczywiście nie zmieni to odwołania do katalogu domowego (trzeba to dodatkowo zmienić ręcznie, poprzez parametr **-m -d <katalog>** dla katalogu domowego)

Przykłady:

usermod -a -G root testowy – dodaje użytkownika do grupy root

usermod -L testowy – blokuje możliwość logowania się użytkownikowi testowy za pomocą swojego hasła (może jednak nadal wykorzystywać klucz SSH)

3) Dodanie nowej grupy użytkowników

groupadd

Najważniejsze opcje:
-f – wymusza sukces operacji; jeżeli podana przez nas grupa już istnieje zostaje zgłoszony sukces operacji
-p – ustawia hasło dla grupy; może ono stanowić dodatkowe zabezpieczenie przed uprawnieniem określonych użytkowników do wykonania jakichś operacji w systemie Linux (hasło ustawia się identycznie jak w przypadku kont użytkowników)
-r – tworzona grupa jest grupą systemową (ma specjalne uprawnienia)

Przykłady użycia:

useradd grupa1 – dodaje grupę o nazwie grupa1
useradd -f grupa1 – dodaje grupę o nazwie grupa1; jeżeli grupa już istnieje to zgłasza pozytywne wykonanie operacji

4) Modyfikowanie grup odbywa się poprzez polecenie

groupmod

Opcje do polecenia są niemal identycznie jak w poprzednim przykładzie. Przykłady wywołania:

groupmod -a grupa2 grupa1 – nazwa grupy grupa1 zostanie zamieniona na grupa2

groupmod -g 2001 grupa1 – identyfikator grupy grupa1 zostanie zamieniony na 2001

5) Usuwanie użytkownikami

userdel

Przykłady użycia:

userdel testowy – usuwa użytkownika z systemu

userdel -f testowy – usuwa użytkownika nawet w przypadku gdy korzysta on aktualnie z systemu bądź wykonują się jego procesy

userdel -r testowy – usuwa użytkownika oraz jego wszystkie pliki i katalogi. Jeżeli pliki i katalogi znajdują się poza katalogiem domowym/w innych systemach plików to należy je usunąć ręcznie

6) Usuwanie grup

groupdel

Przykład:

groupdel przykład – usuwa grupę z systemu o nazwie przykład

4. Przypisywanie praw do zasobów

1) nadanie praw do określonego zasobu odbywa się poprzez polecenie

chmod

Wybrane opcje:
-v – wyświetla informacje o dokonywanych operacjach (w przeciwnym razie wyświetlane są tylko błędy przerywające prace polecenia)
-f – wyłącza niemal wszystkie komunikaty o błędach polecenia
-R – ustanawia podane prawa również dla podkatalogów oraz plików zawartych w katalogu, dla którego zmieniamy uprawnienia

Przykłady użycia:

chmod 600 tajny_kat – ustawia właścicielowi katalogu prawo zapisu oraz odczytu (bit wykonania

nie jest potrzebny) natomiast grupie oraz innym użytkownikom zabrania jakichkolwiek operacji na katalogu. W ten sposób bronimy KOMUKOLWIEK dostępu do zawartości folderu; nawet gdyby wszystkie elementy w nim dawały nam pełne prawa dostępu!
Aby wszyscy użytkownicy mieli możliwość zajrzenia do katalogu i ewentualnie prawa modyfikacji do niektórych zasobów w nim zgromadzonych (np. skopiowania ich) MUSI zostać ustawiony bit wykonania. Oznacza to, że powyższą komendę należy zmienić na następującą:

chmod 655 tajny_kat

O tego momentu użytkownicy będą mogli zajrzeć do katalogu, zajrzeć do plików w nim się znajdujących (i katalogów) jednak nie będą mogli nic zapisać ani zmienić (chyba, że pozwolą im na to osobne prawa danych zasobów)

Jeżeli chcemy aby wszystko co znajdzie się w katalogu także posiadało identyczne uprawnienia musimy wydać polecenie:

chmod -R 655 tajny_kat

2) Aby zmienić właściciela oraz grupę właścicieli używamy polecenia:

chown

Przykład użycia:

chown test:test tajny_kat – zmienia aktualnego właściciela oraz grupę na test
chown -R test tajny_kat – zmienia właściciela katalogu (oraz jego zawartości) na test
chown :test tajny_kat – zmienia tylko grupę na test (np. z root)

ZADANIA:
1. W materiale zostały omówione reprezentacje uprawnień tekstowe oraz bitowe. Proszę wskazać jak jeszcze można reprezentować uprawnienia do zasobów w systemie Linux
2. Co to jest SELinux? Do czego można to wykorzystać?
3. Proszę wskazać przydatność polecenia umask wraz z jego zastosowaniem (przykład).
4. W jaki sposób można zmienić ustawienia domyślne polecenia useradd?
5. Czy w systemie Debian istnieją inne polecenia do zarządzania użytkownikami i grupami? Jeżeli tak, to jakie?
6. Proszę dodać do systemu 3 nowych użytkowników (nazwy dowolne; najlepiej różnymi poleceniami, celem sprawdzenia ich działania). Następnie należy dodać dwie grupy użytkowników (nazwy dowolne). Do grupy pierwszej należy dodać dwóch pierwszych użytkowników, natomiast do drugiej – dwóch ostatnich. W następnej kolejności należy, jako root utworzyć 5 katalogów, by później ponadawać odpowiednie uprawnienia:
- katalog1 ma posiadać pełne prawa dla 1 grupy
- katalog2 ma posiada pełne prawa dla 2 grupy
- katalog3 ma posiadać pełny dostęp dla wszystkich użytkowników; wewnątrz mają znaleźć się katalogi 4 oraz 5 posiadające odpowiednio pełne prawa dla użytkownika 1 oraz 3
Teraz proszę spowodować, że użytkownik 2 będzie miał dostęp do obu tych katalogów (teraz zapewne nie będzie ich miał).
7. Czy w systemie Linux istnieje mechanizm znany z systemu Windows (ACL)? Jeżeli tak to w jaki sposób można go wykorzystać. Jeżeli nie istnieje to czy można go w jakiś sposób zastąpić (o podobnym działaniu)?

Źródła:
http://askubuntu.com/questions/518259/understanding-chmod-symbolic-notation-and-use-of-octal
http://en.wikipedia.org/wiki/File_system_permissions
http://en.wikipedia.org/wiki/Chattr
http://manpages.ubuntu.com/manpages/saucy/pl/man1/chattr.1.html