Wrocław
University
of Science
and Technology

Administrowanie sieciowymi systemami operacyjnymi

Wykład 5

Zarządzenie użytkownikami i autoryzacja

dr inż. Jarosław Rudy

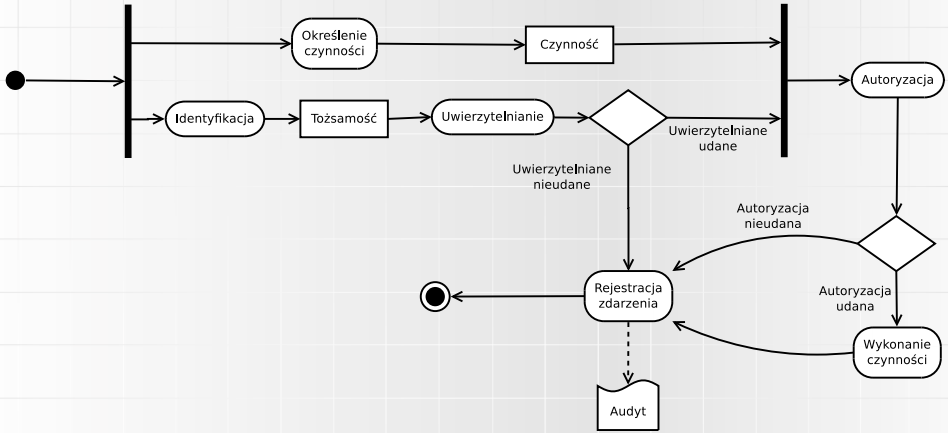
HR EXCELLENCE IN RESEARCH

Wprowadzenie

AAA – Authentication, Authorization, Audit.

- Authentication (uwierzytelnianie) – weryfikacja wiarygodności (potwierdzenie) tożsamości.
- Authorization (autoryzacja) – nadawanie uprawnień (pozwolenie lub zakaz) do danej czynności, zwykle dostępu do zasobu.
- Audit (audyt) – kontrola działań tożsamości, najczęściej poprzez rejestrowanie aktywności (logi), rzadziej kontrole finansowe (accounting).
- Identification (identyfikacja) – określenie (zadeklarowanie) tożsamości.
- Określenie czynności – zwykle oczywiste.

Wprowadzenie



Identyfikacja i uwierzytelnianie wykonywane są zwykle jednorazowo lub pozostają ważne przez pewien czas trwania sesji.

UID oraz GID

- W Linuksie czynność może dotyczyć pliku, zadania (procesu), gniazdka, kolejki komunikatów, współdzielonego obszaru pamięci itp.
- Czynności są wykonywane przez procesy.
- W celu identyfikacji tożsamości wykorzystuje się użytkowników. Ściślej wykorzystywane są numery UID (User ID) oraz GID (Group ID).
- UID i GID określają właściciela i prawa dostępu do większości obiektów, w szczególności procesów i plików.
- Użytkownik może należeć do wielu grup, z czego jedna jest grupą podstawową.

UID oraz GID

Typowe zakresy UID:

- 0: superuser (root). Ma prawa do dowolnej czynności.
- 1–99: zarezerwowane dla aplikacji systemu (przydział statyczny). Duże ryzyko konfliktu.
- 100–499: zarezerwowane dla aplikacji systemu (przydział dynamiczny). Ryzyko konfliktu.
- 500–999: generalnie powinny być dostępne. Niskie ryzyko konfliktu.
- od 1000: „zwykli” użytkownicy.

System generalnie przydziela użytkownikom pierwszy wolny UID wyższy od obecnie zajętych.

UID oraz GID

„Poziomy" identyfikatorów procesu:

- RUID, RGID (real) – „prawdziwe" identyfikatory właściciela procesu. Pochodzą od UID/GID użytkownika, który uruchomił proces.
- EUID, EGIG (effective) – identyfikatory używane do autoryzacji. Normalnie równe RUID/RGID, chyba że:
  - „zmieniono" użytkownika (`sudo`),
  - plik który wykonaliśmy miał ustawiony bit `setuid`.
- SUID, SGID (saved, stored) – używane gdy proces musi chwilowo zmniejszyć uprawniania:
  - ustaw SUID/SGID na EUID/EGID,
  - ustaw EUID/EGID na RUID/RGID,
  - wykonaj czynność,
  - ustaw EUID/GUID na SUID/SGID.

UID oraz GID

- Procesy startowe (`init`, `/etc/init.d`) działają z uprawnieniami roota. Mogą jednak korzystać z obniżenia poziomu uprawnień (generalna zasada bezpieczeństwa).
- Procesy potomne dziedziczą identyfikatory po procesach rodzicielskich.
- Zmiana UID/GID możliwa m.in. z pomocą:
  - funkcje `setuid`, `seteuid`, `setgid`,
  - komenda `su` – zmiana użytkownika (prostsza, korzysta z wartości w pliku `/etc/login.defs`,
  - komenda `sudo` – wykonanie komendy jako inny użytkownik. Rozbudowana, może służyć do ustalenia złożonych obostrzeń. Podstawowa konfiguracja w `/etc/sudoers`, ale są inne możliwości (np. LDAP).

Podstawowe uwierzytelnianie

- Program `login`.
- Podstawowa baza danych kont użytkowników przechowywana jest w pliku `/etc/passwd`. Składnia:

  ```
  login:haslo:UID:GID:komentarz:katalog_domowy:powloka
  ```

- Dane o grupach przechowywane w pliku `/etc/group`. Składnia:

  ```
  nazwa:haslo:GID:lista_uzytkownikow
  ```

- Podstawowe uwierzytelnianie: porównanie hasha podanego hasła z hashem w pliku `/etc/passwd`.
- Hashe uzyskiwane są funkcją `crypt`[1], która udostępnia wiele trybów haszowania, oryginalny bazowany był na szyfrze DES.

[1] Nie należy tego mylić z przestarzałą komendą szyfrującą `crypt`!

Podstawowe uwierzytelnianie

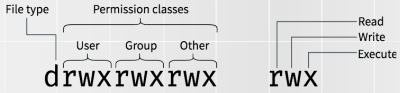
- `/etc/passwd` i `/etc/group` muszą być czytelne dla wszystkich, z ich danych (mapowanie nazwy użytkownika na UID, katalog domowy, powłoka) korzysta wiele komend np. `ls`.
- Hash hasła był więc widoczny dla wszystkich. Nie był to problem dla wczesniego Uniksa (względnie długi czas hashowania, bardziej „zaufana" społeczność użytkowników), jednak sytuacja zmieniła się.
- W związku z tym obecnie hashe haseł zostały przeniesione do osobnych plików `/etc/shadow` oraz `/etc/gshadow`, z odpowiednią („x") wzmianką w `/etc/passwd`. Odczyt tych plików wymaga praw roota. Składnia:

  ```
  username:hash:GID:last_change_date:min_change:max_change:
  warning_period:inactivity_period:account_expire_date
  ```

- Edycja tych plików może być ręczna (choć lepiej zostawić to komendom). Poprawność (spójność) można sprawdzić poprzez komendy `pwck` oraz `grpck`.

Prawa plikowe

- Plik posiada właściciela (user[2]) oraz właściciela grupowego (group).
- Podstawowo plik posiada 9 bitów praw po 3 dla właściciela, właściciela grupowego i innych (other). W każdej „sekcji" są kolejno prawa r (odczyt), w (zapis) oraz x (wykonanie).
- Podgląd praw (często są scalone z typem pliku jako „tryb") można uzyskać poprzez wywołanie `ls -l` lub `stat`, zaś zmiana praw odbywa się z użyciem komendy `chmod` (tylko właściciel/root).
- Prawa podawane są jako 3 cyfry ósemkowe (np. 755) lub symbolicznie (np. u=rwx,go=rx).



[2]Odradza się mylące tłumaczenie „owner".

Prawa plikowe

- Możliwe cztery przypadki, gdzie użytkownik jest:
  - właścicielem pliku (zgodność UID-ów),
  - właścicielem grupowym pliku (zgodność GID-ów dla którejś grupy),
  - zarówno właścicielem jak i właścicielem grupowym (zgodność zarówno UID-ów jak i GID-ów),
  - innym tj. nie jest ani właścicielem, ani właścicielem grupowym (niezgodne UID-y, ani GID-y dla żadnej grupy).
- Co jeśli prawo właściciela grupowego zezwala na dostęp, a prawo właściciela odmawia dostępu? Kolejność sprawdzania:
  - jeśli jesteśmy właścicielem, to liczą się prawa właściciela (niezależnie do grup),
  - w przeciwnym razie jeśli jesteśmy właścicielem grupowym, to liczą się prawa właściciela grupowego,
  - w przeciwnym razie liczą się prawa „innych".

Prawa plikowe

Sytuacje szczególne.

- Użytkownik root ma nieograniczony dostęp.
- Podstawowe prawa katalogu:
  - read – możliwość odczytania (wylistowania np. `ls`) katalogu,
  - write – możliwość zmiany zawartości katalogu czyli jego wpisów (dodanie lub usunięcie pliku, zmiana nazwy pliku) i atrybutów,
  - execute – możliwość wejścia do katalogu (np. `cd`) oraz uzyskania dostępu do zawartych w nim plików.
- Plik generalnie nie posiada własnych praw do jego skasowania – liczy się prawo zapisu dla katalogu nadrzędnego.
- Dostęp do pliku wymaga prawa wykonania w katalogu nadrzędnym i w każdym wcześniejszym katalogu na ścieżce!

Prawa plikowe

Dodatkowa („zerowa") cyfra ósemkowa odpowiada za następujące bity praw:

- Set user ID (setuid, s) – dla pliku wykonywalnego oznacza przyjęcie EUID po UID właściciela pliku, a nie UID użytkownika wykonującego plik. Potrzebne do chwilowego uzyskania wyższych uprawnień (np. komendy `ping` lub `su`). Powinien być używany tylko ze sprawdzonymi programami. W Linuksie setuid dla katalogu jest ignorowany.
- Set group ID (setgid, s):
  - Dla pliku wykonywalnego (niebędącego katalogiem) oznacza przyjęcie EGID po GID właściciela grupowego pliku zamiast GID użytkownika wykonującego plik. Przykładem jest komenda `wall`.
  - Dla katalogu `dir` ustawienie bitu setgid sprawi, że pliki/katalogi tworzone w `dir` będą miały grupę dziedziczoną po `dir`, a nie po tworzącym je użytkowniku.

Prawa plikowe

- Restricted deletion/sticky bit (t) – dla katalogu `dir` oznacza, że plik `file` w `dir` może być usunięty tylko przez (1) roota, (2) właściciela `dir` lub (3) właściciela `file`. Używany np. w publicznie dostępnym `/tmp`.
  - Historycznie sticky bit był w niektórych systemach używany do oznaczenia, by plik pozostał w swap do szybszego użycia w przyszłości (skąd nazwa).
- Specjalne oznaczenie `X` dla komendy `chmod` w celu ułatwienia niektórych scenariuszy nadawania praw.
- umask – maska procesu mówiąca które bity praw należy usunąć (wyzerować) przy tworzeniu nowych plików.

Atrybuty plikowe

Dodatkowe atrybuty plików wspierane przez systemy plików `ext2`, `ext3` oraz `ext4`.

- `a` (append only),
- `A` (no atime updates),
- `d` (no dump/no backup),
- `E` (encrypted),
- `i` (immutable),
- `j` (data jouranling),
- `s` (secure deletion),
- `S` oraz `D` (synchronous (directory) updates),
- `T` (top of directory hierarchy),
- `u` (undeletable).

Podgląd i zmiana (nie zawsze, czasem tylko root) za pomocą komend `lsattr` oraz `chattr` z pakietu `e2fsprogs`.

Access Control Lists

- POSIX-owe Access Control Lists (ACL) to nadbudowa na zwykłe prawa plikowe, umożliwiająca dokładniejszą kontrolę uprawnień dla użytkowników.
- Wymaga wsparcia jądra (obecnie domyślnie w praktycznie wszystkich dystrybucjach) i zamontowania systemu pliku z opcją `acl`.
- Każdy plik może mieć zwykłe ACL (access ACL).
- Katalogi mogą mieć domyślne ACL (default ACL) – dziedziczone przez tworzone w nich pliki (dla których stają się access ACL).
- ACL składa się z listy wpisów, zaś każdy wpis posiada typ, kwalifikator (czasami pusty) i bity praw. Przykłady:

  `u::rwx` `g:www:r-x` `o::---` `m::rw-`

- Podgląd i ustawienie ACL (ogólnie tylko właściciel pliku i root) za pomocą komend `getfacl` oraz `setfacl`, pakiet `acl`.

Access Control Lists

Sześć możliwych typów wpisów:

- `ACL_USER_OBJ` – właściciel pliku (user), bez kwalifikatora,
- `ACL_USER` – nazwany użytkownik (UID lub nazwa w kwalifikatorze),
- `ACL_GROUP_OBJ` – właściciel grupowy pliku (group), bez kwalifikatora,
- `ACL_GROUP` – nazwana grupa (GID lub nazwa w kwalifikatorze),
- `ACL_MASK` – maska praw maksymalnych dla `ACL_USER` oraz `ACL_GROUP`, bez kwalifikatora,
- `ACL_OTHER` – pozostali (other), bez kwalifikatora.

Typy `ACL_USER_OBJ`, `ACL_GROUP_OBJ` oraz `ACL_OTHER` mapują się *obustronnie* do zwykłych praw pliku.

Access Control Lists

Poprawne (valid) ACL dla pliku muszą spełniać następujące warunki:

- dokładnie po jednym (zwykłym) wpisie typu `ACL_USER_OBJ`, `ACL_GROUP_OBJ` oraz `ACL_OTHER`,
- co najwyżej jeden (zwykły) wpis typu `ACL_MASK`,
- jeśli są (zwykłe) wpisy typu `ACL_USER` lub `ACL_GROUP` to wpis typu `ACL_MASK` jest obowiązkowy,
- powyższe reguły muszą być spełnione dla domyślnych ACL (jeśli są),
- liczba wpisów `ACL_USER` oraz `ACL_GROUP` nie jest limitowana.

Access Control Lists

Algorytm autoryzacji (priorytety wpisów):

1. Jeśli EUID procesu pasuje do wpisu `ACL_USER_OBJ` to decyduje ten wpis.
2. Jeśli EUID procesu pasuje do któregoś wpisu `ACL_USER` to decyduje ten wpis w połączeniu z `ACL_MASK` (prawo musi być w obu).
3. Jeśli EGID procesu lub GID którejś z grup pobocznych pasuje do wpisu `ACL_GROUP_OBJ` lub któregoś z wpisów `ACL_GROUP` to:
   - 3.1 Jeśli istnieje wpis `ACL_MASK` to:
     - 3.1.1 Jeśli prawo jest zarówno w `ACL_MASK` jak i w którymś z pasujących wpisów `ACL_GROUP_OBJ` lub `ACL_GROUP` to prawo przyznane, w przeciwnym razie odmowa.
   - 3.2 W przeciwnym razie (oznacza to brak `ACL_GROUP` bo brak `ACL_MASK` decyduje wpis `ACL_GROUP_OBJ`.
4. Decyduje wpis `ACL_OTHER`.

Access Control Lists

Przykład (z dokumentacji `getfacl`):

```
 1: # file: somedir/
 2: # owner: lisa
 3: # group: staff
 4: # flags: -s-
 5: user::rwx
 6: user:joe:rwx                #effective:r-x
 7: group::rwx                  #effective:r-x
 8: group:cool:r-x
 9: mask::r-x
10: other::r-x
11: default:user::rwx
12: default:user:joe:rwx        #effective:r-x
13: default:group::r-x
14: default:mask::r-x
15: default:other::---
```

Limitowanie zasobów

- Komenda `ulimit` – raportowanie i ustawianie limitów zasobów wykorzystywanych przez procesy
  - limit dla dostępnej pamięci (w tym segmentu danych procesu), priorytetu procesów (wartość nice), liczby otwartych deskryptorów plików, liczby wątków, liczby procesów, rozmiaru potoków/kolejek komunikatów, rozmiaru stosu, czasu procesora itp.,
  - różne wielkości mają różne jednostki,
  - specjalne wartości `soft`, `hard` oraz `unlimited`,
  - zwykły użytkownik nie może zwiększyć limitu miękkiego poza limit twardy,
  - zwykły użytkownik nie może zwiększyć limitu twardego (ale może go zmniejszyć),
  - root nie ma ograniczeń.

Limitowanie zasobów

- Pakiet `quota` – limitowanie przestrzeni dyskowej
  - opcje `usrquota` oraz `grpquota` w pliku `/etc/fstab`,
  - komenda `quotacheck` – skanowanie dysków, tworzenie (opcja `-c` i naprawa plików quota,
  - komendy `quotaon` oraz `quotaoff`,
  - komenda `edquota` – edycja limitów przestrzeni dyskowej (w blokach) i dozwolonej liczby plików użytkownika lub grupy. Uruchamia osobny edytor
    - limit miękki może być przekroczony do pewnego czasu (grace period),
    - limit twardy nieprzekraczalny.
  - komenda `repquota` – raportowanie ustawionych limitów.

Zarządzanie użytkownikami

Komenda `useradd`

- „Niskpoziomowe” tworzenie użytkownika.
- Stworzenie użytkownika wymaga praw roota.
- Modyfikuje `/etc/passwd`, `/etc/shadow`, `/etc/group` oraz `/etc/gshadow`.
- `/etc/default/useradd` – ustawienia domyślne (powłoka, grupa, ścieżka na katalogi domowe, lokalizacja pliku `skel`).
- `/etc/login.defs` – kontrola logowania (komunikaty, wymagania dotyczące haseł i ich wygaśnięcia, zakres ID dla użytkowników).
- `/etc/skel` – katalog zawierający domyślną strukturę (szkielet) nowo tworzonych katalogów domowych.

Zarządzanie użytkownikami

- `useradd opcje nazwa_uzytkownika`,
- `-b sciekza` – ścieżka bazowa katalogu domowego (np. `/home`). Jeśli katalog nie jest fizycznie tworzony to musi istnieć.
- `-d nazwa` – nazwa katalogu domowego (domyślnie jest to nazwa użytkownika), doklejana do ścieżki bazowej. Całość domyślnie nie musi istnieć.
- `-m` – wymuś utworzenie katalogu domowego jeśli nie istniał (wraz ze skopiowaniem pliku z `/etc/skel`.
- `-u UID` – wybór UID.
- `-g GID` lub `-g grupa` – wybór podstawowej grupy (musi istniec).
- `-G lista,grup` – wybór dodatkowych grup.

Zarządzanie użytkownikami

- `-s sciezka` – wybór powłoki.
- `-e YYYY-MM-DD` – wybór daty wygaśnięcia ważności konta.
- `-f liczba` – wybór liczby dni od wygaśnięcia hasła po których wygasa konto.
- `useradd -D` – wyświetla domyślne wartości tworzonych użytkowników. Wraz z niektórymi opcjami pozwala zmienić domyślne wartości dla:
  - ścieżki bazowej katalogu domowego,
  - daty wygaśnięcia konta,
  - czasie wygaśnięcia konta po wygaśnięciu hasła,
  - grupy podstawowej,
  - powłoki.

Zarządzanie użytkownikami

Komenda `passwd` – ustawienie lub zmiana hasła.

- Zmiana nie zawsze jest możliwa.
- Hasło musi być odpowiednio złożone (konkretne reguły są zwykle zawarte w pliku `/etc/pam.d/common-password`).
- `-e` – natychmiastowe wygaśnięcie hasła (zmusza do zmiany).
- `-i` – czas do wygaśnięcia konta po wygaśnięciu hasła.
- `-n liczba` – minimalna liczba dni pomiędzy zmianami hasła.
- `-x liczba` – maksymalna liczba dni pomiędzy zmianami hasła (termin wygaśnięcia).
- `-w liczba` – liczba dni przed wygaśnięciem hasła kiedy ma być wyświetlane ostrzeżenie o potrzebie jego zmiany.

Zarządzanie użytkownikami

- `userdel` – usuwanie użytkowników. Można wymusić usunięcie zalogowanego użytkownika oraz skasować pliki w jego katalogu domowym.
- `usermod` – modyfikacja istniejącego użytkownika (opcje podobne do useradd). Możliwość przenoszenia katalogu domowego.
- `groupadd` – dodanie grupy.
- `adduser` oraz `deluser` – w niektórych dystrybucjach „przyjaźniejsze" nakładki na `useradd` oraz `userdel` z konfiguracją w `/etc/adduser.conf`.
- Wyświetlenie informacji o użytkownikach:
  - `id`, `groups` – informacje o danym użytkowniku (UID, GUID itp.).
  - `finger` – katalog domowy, powłoka, terminal, czas zalogowania (dany użytkownik lub wszyscy zalogowani).
  - `who` oraz `w` – informacje o zalogowanych użytkownikach.
  - `lslogins` – wyświetlenie znanych użytkowników.

Sieciowe zarządzanie użytkownikami

- Wspólna (lokalna i zdalna) baza kont użytkowników
  - System NIS (Network Information Service) oraz NIS+ (dawniej Yellow Pages, yp) dla Linuksa,
  - usługa Active Directory dla Windowsa.
- Koncepcja systemu NIS
  - lokalne pliki `/etc/passwd`, `/etc/shadow` itp.,
  - uzupełniane przez dane pobierane z serwerów NIS,
  - uwierzytelnianie realizowane lokalnie,
  - możliwość pobrania zdalnego katalogu domowego po zalogowaniu (NFS).

Sieciowe zarządzanie użytkownikami

Strona klienta

- Użycie Name Switch Service (NSS) wraz z konfiguracją w pliku `/etc/nsswitch.conf`

```
passwd: files ldap
group: files ldap
shadow: files ldap
hosts: dns nis files
```

  Różne możliwe bazy danych (użytkownicy, hosty, aliasy, usługi itp.), różne możliwe źródła (files, nis, nisplus, dns, ldap itp.).

- `ypbind` – daemon NIS
  - okresowy odczyt map z serwera NIS,
  - wpis `ypserver adres` w pliku `/etc/ypconf`,
  - nazwa domeny NIS w pliku `/etc/defaultdomain`.

Sieciowe zarządzanie użytkownikami

Zmiany w `/etc/passwd`, `/etc/group` itp.

- `+::::::` – dodanie całej mapy NIS (`+:::` dla pliku `/etc/group`),
- `+` i `-` – kontrolują możliwość logowania i dane domyślne,
- `@` kontroluje grupy sieciowe (netgroup),
- kolejność ma znaczenie.

```
root:x:0:10:super user:/:/bin/sh
fran:x:121:100:Fran Sisco:/u/fran/:/bin/ksh
-renee:
-@marketing:
+diego::::::
+:::::/u/guest:/bin/rksh
+@developers:
```

Sieciowe zarządzanie użytkownikami

Strona serwera

- pakiet `nis`,
- konfiguracja daemonów
  - wpisy `domain <domena> server <serwer>` w pliku `/etc/yp.conf` – mapowanie domen na obsługujące serwery,
  - plik `/etc/default/nis`
    - włączenie NIS, wpis `NISSERVER = master`,
    - podanie pliku mapy użytkowników, wpis `YPPWDDIR=/etc`,
    - inne.
  - plik `/etc/ypserv.securenets` – ograniczenie dostępu z określonych domen i hostów

```
255.255.255.10 192.168.0.1
host 13.13.14.1
host 13.13.14.
```

Sieciowe zarządzanie użytkownikami

- Budowa mapy użytkowników
  - plik `/var/yp/Makefile` – określenie interesujących nas baz danych np. `ALL = passwd group`,
  - kompilacja poprzez `cd /var/yp; make` lub `make -C /var/yp`.
- Użytkownicy dodawani poprzez `adduser`/`addgroup`, po czym należy wpisy przenieść do map serwera
- Ze względów bezpieczeństwa zwykle należy też usunąć wpisy utworzonego użytkownika ze zwykłych plików `/etc/passwd`, `/etc/shadow` itd. serwera.

Sieciowe zarządzanie użytkownikami

- Zmiana hasła NIS odbywa się normalnie poprzez `passwd` (o ile NIS jest odpowiednio ustawiony). Dawniejszy sposób (komenda `yppasswd`) jest uznana za przestarzałą.

- `ypcat` oraz `ypmatch` – listowanie wartości dla części lub całości kluczy NIS.

Uwierzytelnianie poprzez LDAP

Lightweight Directory Access Protocol (LDAP) – otwarty standard dostępu do rozproszonych usług katalogowych.

- Uwzględnienie LDAP w pliku `nsswitch.conf`.
- Klienta LDAP – pakiety `libnss-ldap` lub nowszy `libnss-ldapd`.
- Przydatny daemon `nscd` – cache'owanie zapytań o usługi katalogowe.
- Zarządzanie zawartością bazy LDAP
  - Pakiety `ldap-utils` oraz `ldapscripts`.
  - Komendy `ldapadduser`, `ldapsetpasswd`, `ldapmodifyuser`.

Uwierzytelnianie poprzez LDAP

- Pluggable Authentication Module (PAM) – wysokopoziomowe API dostępu do różnych niskopoziomowych metod zarządzania kontami, uwierzytelnianiem, hasłami i sesją.
- Konfiguracja ogólna oraz specyficzna dla poszczególnych aplikacji korzystających z PAM.
- Dla LDAP:
  - instalacja pakietu `libpam_ldap`,
  - konfiguracja pliku `/etc/ldap.conf`,
  - konfiguracja plików `/etc/pam.d/common-*`
    - dodanie odwołań do biblioteki `pam_ldap.so`.