Instytut Teleinformatyki
Wydział Inżynierii Elektrycznej i Komputerowej
Politechnika Krakowska

Laboratorium Administrowania Systemami Komputerowymi

„Systemy autoryzacji”

ćwiczenie numer: 11

Spis treści

1. Wstępne informacje

1.1 TEMAT ĆWICZENIA

Autoryzacja jest procesem polegającym na przydzielaniu (bądź odmowie) praw dostępu do poszczególnych danych, obiektów, usług lub funkcji systemu klientowi (użytkownikowi lub programowi).
Autoryzacja następuje najczęściej na podstawie przeprowadzonego uprzednio uwierzytelnienia klienta.

Tematem tego ćwiczenia będą popularne systemy autoryzacji kerberos, pam i opie.

Kerberos

Bezpieczny system autoryzacji zapewniający dostęp do wybranych usług/serwerów tylko autoryzowanym użytkownikom, system zabezpiecza również użytkowników przed komunikacją z usługami/komputerami podszywającymi się pod autoryzowane serwery.
System jest oparty o politykę wydawania tzw. ticketów uprawniających do korzystania z danej usługi.
Tickety są wydawane przez serwer KDC (Key distribution service). Użytkownicy, hosty, usługi objęci kontrolą kerberosa są zapisani w bazie danych kerberosa wraz z przysługującymi im przywilejami.

PAM

Jest elastycznym systemem autentykacji użytkowników umożliwiającym tworzenie aplikacji całkowicie niezależnych od samej metody autentykacji. Nie narzuca on żadnych metod oraz nie ma ograniczeń na ich tworzenie. PAM używa zestawu "modułów autentykacyjnych", które umożliwiają implementację dowolnych metod autentykacji i używanie ich z dowolnymi aplikacjami.

OPIE

Narzędzie opie pozwala na wykorzystywanie jednorazowych haseł w celu autoryzacji dostępu do systemów komputerowych. Zapis informacji(hasła) odbywa się na 64 bitowym słowie, stworzonym na podstawie algorytmów szyfrowania takich jak: MD4, czy MD5, następnie na podstawie tego tworzone jest 6 angielskich słów, które stanowią jednorazowe hasło.

OpenLDAP

To implementacja protokułu LDAP (wersji 2 i 3). Zawiera serwer usług katalogowych, biblioteki oraz klientów do komunikacji z serwerem. LDAP - Lightweight Directory Access Protocol (Lekki Protokół Dostępu do Katalogu) jest protokołem przeznaczonym do korzystania z usług katalogowych. Główne zalety to scentralizowany dostęp do danych z różnych maszyn, replikacja tych danych, bezpieczeństwo, obsługa przestrzeni nazw itp.

1.2 ZAGADNIENIA DO PRZYGOTOWANIA

Przed przystąpieniem do wykonania ćwiczenia należy zapoznać się z następującymi zagadnieniami:

- Zasada działania i moduły systemu autoryzacji PAM.
- Zasada działania systemu Kerberos.
- Architektura i działanie systemu OPIE.
- Zasada działania usługi katalogowej LDAP (OpenLDAP).

1.3 CEL ĆWICZENIA

Dzięki temu ćwiczeniu wykonujący pozna:

- możliwości systemu PAM,
- przykłady konfiguracji niektórych modułów PAM,
- sposób konfiguracji niezbędnych składników systemu Kerberos,
- metody administracji systemem Kerberos,
- sposób generowania i zasadę działania sytemu haseł jednorazowych,
- sposób konfiguracji serwera OpenLDAP,
- sposób tworzenia kont użytkowników w OpenLDAP.

2. Przebieg ćwiczenia

2.1 Przygotowanie ćwiczenia

Po załączeniu komputera należy uruchomić system operacyjny o nazwie ASK. Jest to dedykowany system umożliwiający wykonanie niniejszego ćwiczenia.

Logowanie

W celu wykonania ćwiczenia konieczne jest zalogowanie się na konto administratora (login: root, hasło: lab).

Katalog laboratoryjny

Przed przystąpieniem do zajęć należy uruchomić skrypt inicjujący środowisko pracy.

```
stanowisko01:~/#/labpam/skrypty/prepare.sh
```


2.2 Zadanie nr 1 – PAM flagi kontrolne cz. 1 (moduł: pam_nologin.so)

W tym zadaniu wykorzystany zostanie moduł pam_nologin.so, służący do zablokowania możliwości logowania na serwer każdemu za wyjątkiem użytkownika root.

Proszę wykonać następujące czynności:

1. Przy pomocy dowolnego edytora tekstu otworzyć plik /etc/pam.d/login.

2. Proszę sprawdzić, czy istnieje reguła autentykacji (auth) pozwalającą na zablokowanie możliwości logowania zwykłym użytkownikom. Z jakiej flagi kontrolnej korzysta?

3. Proszę zablokować możliwość logowania zwykłym użytkownikom.

```
stanowisko01:~/# touch /etc/nologin
```

4. Proszę otworzyć konsolę tekstową (Alt+F1) i spróbować zalogować się na konto dowolnego zwykłego użytkownika (utworzone są dwa zwykłe konta: lab1 oraz lab2 posiadające hasła takie jak nazwa użytkownika, ewentualnie jeśli hasło się nie zgadza proszę spróbować haseł: lab1lab1, lab2lab2).

Próba logowania powinna być zakończona błędem `Login incorrect` jeszcze przed pytaniem o hasło. Takie zachowanie systemu nie jest bezpieczne ponieważ każdy użytkownik nawet nie mający konta na serwerze zauważy zmianę w jego funkcjonowaniu.

5. Proszę tak zmienić zastosowaną regułę by proces login zapytał o hasło i tym samym nie pokazał po sobie żadnej zmiany w konfiguracji serwera: konieczne jest przeniesienie w/w reguły na koniec pliku /etc/pam.d/login.

6. Proszę powtórzyć próbę logowania i porównać komunikaty.

7. Proszę wykonać próbę zalogowania się na konto root.

8. Proszę usunąć plik /etc/nologin

```
stanowisko01:~/#rm /etc/nologin
```


2.3 Zadanie nr 2 – PAM flagi kontrolne cz. 2 (moduł: pam_rootok.so)

Moduł pam_rootok.so pozwala korzystać użytkownikowi root z polecenia su bez podawania hasła. W zadaniu tym zmienione zostanie domyślne zachowanie systemu tak, aby użytkownik root zobligowany został do podania hasła w momencie wykonywania polecenia su.

Proszę wykonać następujące czynności:

1. Przy pomocy dowolnego edytora tekstu otworzyć plik /etc/pam.d/su.

2. Proszę usunąć z pliku regułę

```
auth sufficient pam_rootok.so
```

3. Proszę dokonać próby wykonania polecenia su

```
stanowisko01:~/# su – lab1
```


2.4 ZADANIE NR 3 –PAM PRZYKŁAD AUTORYZACJI – PLIK UŻYTKOWNIKÓW (MODUŁ: PAM_LISTFILE.SO)

W tym zadaniu wykorzystany zostanie moduł pam_listfile.so , w celu określenia którym użytkownikom pozwalamy/odmawiamy używać wybranej usługi.

Konfiguracja tego modułu wymaga podania odpowiednich opcji.

Nazwa	Opis
`onerr=succeed|fail`	Parametr określa czy w przypadku jakiegoś błędu (np. braku pliku z listą użytkowników) metoda autentykacji potwierdza autentykacje(succeed) czy też powiadamia o nieudanej autentykacji (fail).
`item=user|tty|rhost|ruser|group|shell`	Parametr określa rodzaj elementu który moduł będzie sprawdzał. Zgodnie z ustawioną wartością będziemy tworzyć później plik z nazwami elementów.
`sense=deny|allow`	Określenie logiki działania. Ustawienie deny powoduje zablokowanie chronionej usługi dla wszystkich wymienionych w pliku, ustawienie allow powoduje udostępnienie usługi użytkownikom wymienionym w pliku.
`file=nazwa_pliku`	Parametr to ścieżka do pliku z definicją użytkowników

Proszę wykonać następujące czynności:

1. Utworzyć plik /etc/pam.d/ssh_users

```
stanowisko01:~/# touch /etc/pam.d/ssh_users
```

2. Plik powinien składać się z nazw użytkowników, każda w nowej linii, proszę w pierwszej linii pliku umieścić nazwę użytkownika lab1.

3. Przy pomocy dowolnego edytora tekstu proszę otworzyć plik /etc/pam.d/sshd.

4. Proszę dołączyć regułę autentykacji (auth) wykorzystując moduł pam_listfile.so, która będzie zabraniać logowania przez ssh tylko użytkownikom wymienionym w pliku /etc/ssh_users

```
auth required pam_listfile.so item=user sense=deny
onerr=fail file=/etc/pam.d/ssh_users
```

5. Proszę przetestować logowanie przez ssh na konta użytkowników root, lab1 i lab2.

6. Proszę zmodyfikować dodaną poprzednio regułę autentykacji w ten sposób, aby dla użytkowników wymienionych w pliku /etc/pam.d/ssh_users dostęp był zabroniony

```
 auth required pam_listfile.so item=user sense=allow
onerr=fail file=/etc/pam.d/ssh_users
```

7. Proszę przetestować logowanie przez ssh na konta użytkowników root, lab1 i lab2.

8. Proszę z pliku /etc/pam.d/sshd usunąć dodane wpisy.

2.5 ZADANIE NR 4 – PAM PRZYKŁAD AUTORYZACJI – BAZA UŻYTKOWNIKÓW (MODUŁ: PAM_PGSQL.SO)

W tym zadaniu wykorzystany zostanie moduł pam_pgsql.so. Służy on do autentykacji użytkowników, którzy są definiowani w bazie Postgresql.

Konfiguracja tego modułu wymaga podania odpowiednich opcji.
user, password, database, table, user_column, pwd_column, pw_type

Nazwa	Opis
`user=username`	Nazwa użytkownika uprawnionego do korzystania z bazy podtgresql
`password=dbname`	Hasło użytkownika uprawnionego do korzystania z bazy podtgresql
`database=dbname`	Nazwa założonej bazy zawierającej tabele kont użytkowników
`table=tablename`	Nazwa tabeli z kontami użytkowników
`user_column=colname`	Nazwa kolumny z nazwami użytkowników
`pwd_column=colname`	Nazwa kolumny z hasłami użytkowników
`pw_type`	Sposób szyfrowania haseł w bazie: clear, crypt, md5

Proszę wykonać następujące czynności:

1. Utworzyć bazę danych i tabelę dla kont użytkowników logując się uprzednio jako użytkownik postgres (polecenie `su -postgres`):

```
postgres@stanowisko01:~/# psql template1
```

```
template1=# CREATE TABLE konta (id integer, nazwa
varchar(100), haslo varchar(100));
```


2. Dodać dane użytkownika lab1 do tabeli konta:

```
template1=# INSERT INTO konta values(1, `lab1`, `lab1`);
```

3. Sprawdzić, czy utworzona została tabela konta i znajdują się w niej odpowiednie wpisy:

```
template1=# \d konta
```

```
template1=# select * from konta;
```

```
template1=# \q
```

4. Używając dowolnego edytora otworzyć plik konfiguracyjny pam dla usługi ssh. Dołączyć regułę autentykacji (auth) wykorzystując moduł pam_pgsql.so.

```
auth required pam_pgsql.so user=root database=template1
table=konta user_column=nazwa pwd_column=haslo pw_type=clear
```

5. Dodać konto użytkownika lab3 do bazy danych.

```
postgres@stanowisko01:~/# psql template1
```

```
template1=# INSERT INTO konta values(2, `lab3`, `lab3`);
```

```
template1=# \q
```

6. Przetestować logowanie przez ssh na konta root, lab1, lab2 oraz lab3.

7. Zmienić konfigurację autentykacji w ten sposób, aby hasła w bazie danych były przechowywane w postaci zaszyfrowanej z użyciem funkcji crypt. W pliku konfiguracyjnym pam dla usługi ssh proszę zmienić wpis na poniższy:


```
auth required pam_pgsql.so user=root database=template1
table=konta user_column=nazwa pwd_column=haslo pw_type=crypt
```

8. Przygotowujemy hasło w zaszyfrowanej postaci dla użytkownika lab1.

```
postgres@stanowisko01:~/# perl –e `print
crypt($ARGV[0],”ab”)` lab1
```

9. Na ekranie wyświetlone zostanie hasło lab1 w postaci zaszyfrowanej. Hasło to należy umieścić w bazie danych:

```
postgres@stanowisko01:~/# psql template1
```

```
template1=# UPDATE konta SET haslo=`zaszyfrowane_haslo`
where nazwa=`lab3`;
```

```
template1=# \q
```

10. Przetestować logowanie przez ssh na konto lab1.

11. Po zakończeniu zadania proszę uruchomić skrypt /labpam/skrypty/prepare.sh

2.6 ZADANIE NR 5 – KERBEROS – KONFIGURACJA SERWERA KDC (KEY DISTRIBUTION CENTER)

Działanie serwera KDC polega na wydawaniu ticket'ów (ograniczonych czasowo biletów pozwalających na korzystanie z konkretnej usługi) autoryzowanym użytkownikom.

Zadanie polega na zapisaniu odpowiedniej zawartości plików konfiguracyjnych kerberosa: krb5.conf, kdc.conf, założeniu kont administratorskich oraz uruchomieniu demona serwera krb5kdc.

Konfiguracja pliku krb5.conf
Plik ma strukturę windows'owych plików INI. Sekcje są poprzedzone nagłówkiem zawierającym nazwę sekcji, zapisaną w nawiasach kwadratowych.

Sekcje konfiguracji wykorzystywane w ćwiczeniu:

Nazwa	Opis
`libdefaults`	Zawiera domyślne wartości używane przez biblioteki kerberosa. Najważniejszą wartością jest **default_realm**, która określa domyślny zestaw serwerów.
`realms`	Definicje zestawów serwerów, każdy zestaw powinien definiować przynajmniej jeden serwer **kdc** oraz jeden **admin_server**
`logging`	Nazwy plików z logami kerberosa

Konfiguracja:

```
[libdefaults]
    default_realm = ITILAB

[realms]
    ITILAB = {
        kdc = kdctest
        admin_server = kdctest
    }

[logging]
    kdc = FILE:/var/log/kerberos.log
    admin_server = FILE:/var/log/kerberos.log
    default = FILE:/var/log/kerberos.log
```

Prosze wykonać następujące czynności:

1. Otworzyć do edycji plik /etc/krb5.conf i w taki sposób zmodyfikować plik konfiguracyjny, aby znalazły się w nim wszystkie wpisy pokazane w powyższej konfiguracji. Proszę żadnych wpisów z pliku nie usuwać, a jedynie zmodyfikować lub uzupełnić istniejące.

2. Konfiguracja pliku /etc/krb5kdc/kdc.conf
Plik ma strukturę windowsowych plików INI. Sekcje są poprzedzone nagłówkiem zawierającym nazwę sekcji, zapisaną w nawiasach kwadratowych.

Sekcje konfiguracji wykorzystywane w ćwiczeniu:

Nazwa	Opis
kdcdefaults	Zawiera domyślne wartości używane przez serwer KDC. Najważniejszą wartością jest **kdc_ports**, która określa numery portów na jakich serwer zostanie uruchomiony.
realms	Definicje atrybutów KDC dla zestawów serwerów, każdy zestaw powinien definiować atrybuty takie jak w poniżej prezentowanym przykładzie.

Konfiguracja:

```
[kdcdefaults]
   kdc_ports = 750,88

[realms]
   ITILAB = {
         database_name = /var/lib/krb5kdc/principal
         admin_keytab = FILE:/etc/krb5kdc/kadm5.keytab
         acl_file = /etc/krb5kdc/kadm5.acl
         key_stash_file = /etc/krb5kdc/stash
         kdc_ports = 750,88
         max_life = 10h 0m 0s
         max_renewable_life = 7d 0h 0m 0s
         supported_enctypes = des3-hmac-sha1:normal des-cbc-
   crc:normal des:normal des:v4 des:norealm des:onlyrealm
   des:afs3
         default_principal_flags = +preauth
   }
```

Proszę w taki sposób zmodyfikować plik kdc.conf, aby zawierał wpisy pokazane powyżej.

3. Proszę utworzyć plik /etc/krb5kdc/kadm5.acl z listą autoryzowanych administratorów dodając do niego linię uprawniającą dowolnego użytkownika z instancją admin do wykonywania wszystkich możliwych czynności administracyjnych. Należy w tym pliku umieścić poniższą linię:

```
*/admin@ITILAB *
```

4. Stworzyć początkową bazę danych Kerberos:

```
stanowisko01:~/# kdb5_util create –r ITILAB -s
```


Po podaniu hasła (proszę podać hasło lab) powinna zostać stworzona baza danych systemu Kerberos. Świadczy o tym pojawienie się 5 plików w katalogu /var/lib/krb5kdc.

5. Dodać co najmniej jedno konto administratora do bazy systemu Kerberos. Należy to wykonać przy użyciu programu kadmin.local
Uruchomienie tego programu, wywoła interpreter poleceń administracyjnych, dzięki którym można zarządzać kontami użytkowników. Proszę dodać konto z instancją administratora używając komendy addprinc, według takiego wzorca:
kadmin.local: addprinc nazwa_uzytkownika/nazwa_instancji@NAZWA_REALM

```
stanowisko01:~/# kadmin.local
```

```
addprinc root/admin@ITILAB
```

Jako hasło użytkownika root proszę podać lab. Konsolę opuszcza się poleceniem quit.

6. Uruchomienie demonów systemu Kerberos.

Aby KDC zaczął działać należy uruchomić dwa demony obsługujące żądania jego klientów:

```
stanowisko01:~/# /etc/init.d/krb5-kdc restart
```

```
stanowisko01:~/# /etc/init.d/krb5-admin-server restart
```

Pierwszy z nich służy do autentyfikacji klientów oraz do wydawania biletów, natomiast drugi udostępnia konsolę administracyjną dla uprawnionych użytkowników.

Proszę sprawdzić czy uruchomiły się prawidłowo przeglądając logi:

```
stanowisko01:~/# tail –f /var/log/kerberos.log
```

Powinny się pokazać między innymi dwa wpisy:
Oct 02 12:35:47 stanowisko01 krb5kdc[3187](info): commencing operation
Oct 02 12:35:52 stanowisko01 kadmind[3189](info): starting

2.7 ZADANIE NR 6 – KERBEROS – POBIERANIE I NISZCZENIE TICKET'ÓW

Aby zwykły użytkownik mógł korzystać z systemu Kerberos, powinien mieć założone przez administratora konto w bazie Kerberos'a.
Jeśli ten warunek jest spełniony użytkownik wykorzystując programy kinit, klist, kdestroy może zarządzać swoimi ticketami.

Prosze wykonać następujące czynności:

1. Wykorzystując konto root, proszę dodać konto zwykłego użytkownika do bazy Kerberos'a. Należy do tego celu wykorzystać program kadmin w analogiczny sposób jak w poprzednim ćwiczeniu, gdy korzystaliśmy z kadmin.local.
Nie należy definiować konta zwykłego użytkownika z instancją admina. W definicji nazwę instancji można pominąć, wówczas komenda będzie wyglądała zgodnie z poniższym wzorcem :
kadmin: addprinc nazwa_uzytkownika@NAZWA_REALM

```
stanowisko01:~/# kadmin
```

```
addprinc lab1@ITILAB
```

Jako hasło użytkownika lab1 proszę podać lab1. Konsolę opuszcza się poleceniem quit.

2. Zalogować się na konto zdefiniowanego przed chwilą użytkownika i wykonać program klist. W wyniku powinniśmy otrzymać pustą listę ticket'ów.

```
lab1@stanowisko01:~$# klist
```

3. Pobrać ticket używając programu kinit (należy podać hasło lab1).

```
lab1@stanowisko01:~$# kinit
```

4. Wyświetlić listę pobranych ticket'ów.

```
lab1@stanowisko01:~$# klist
```

5. Zniszczyć wszystkie pobrane ticket'y wykorzystując polecenie kdestroy.

```
lab1@stanowisko01:~$# kdestroy
```


6. Po zakończeniu zadania proszę uruchomić skrypt /labpam/skrypty/prepare.sh

2.8 ZADANIE NR 7 – HASŁA JEDNORAZOWE

Działanie systemu OPIE pozwalającego na wykorzystywanie haseł jednorazowych zaprezentowane zostanie na przykładzie usługi ssh.

Proszę wykonać następujące czynności:

1. W pliku /etc/ssh/sshd_config proszę ustawić wartość opcji ChallengeResponseAuthentication na yes.

2. Po dokonaniu zmiany konfiguracji serwera ssh konieczne jest jego ponowne uruchomienie:

```
stanowisko01:~/# /etc/init.d/ssh restart
```

3. W pliku konfiguracyjnym usługi ssh systemu PAM (/etc/pam.d/ssh) proszę dodać jako pierwszy następujący wpis:

```
auth sufficient pam_opie.so
```

4. Proszę zalogować się jako użytkownik lab1.

5. Polecenie opiepasswd tworzy hasło dla użytkownika w systemie haseł jednorazowych:

```
lab1@stanowisko01:~$# opiepasswd –f -c
```

Jako hasło proszę podać lab1lab1lab1. Informacja jaka pojawi się na ekranie będzie podobna do poniższej:

ID lab1 OTP key is 499 st1699

Istotne są dwa ostatnie elementy, jako że konieczne jest ich wykorzystanie w kolejnym kroku.

6. Generowanie listy 10 haseł:

```
lab1@stanowisko01:~$# opiekey –n 10 499 st1699
```


Proszę podać hasło lab1lab1lab1. Na ekranie pojawi się lista haseł jednorazowych wraz z odpowiadającymi im numerami. Hasła te będą dalej używane do zalogowania się z wykorzystaniem usługi ssh.

7. Proszę zalogować się na konto użytkownika lab1 z wykorzystaniem usługi ssh. Proszę podać jednorazowe hasło o takim numerze o jaki użytkownik zostanie poproszony.

8. Proszę powtórzyć próbę logowania. Użytkownik zostanie poproszony o podanie kolejnego hasła z listy.

2.9 ZADANIE NR 8 - KONFIGURACJA SERWERA OPENLDAP

1. Konfiguracja:
   Sprawdzamy czy serwer LDAP jest uruchomiony, możemy to zrobić np. poleceniem ps:

```
stanowisko01:~/# ps aux | grep slapd | grep -v grep
```

Jeżeli serwer nie jest uruchomiony proszę go uruchomić:

```
stanowisko01:~/#  /etc/init.d/slapd start
```

Proszę sprawdzić czy serwer nasłuchuje na właściwym porcie (389 zwykły LDAP; 636 LDAP po SSL):

```
stanowisko01:~/# netstat -ltnp | grep :389
```

Powinien się pojawić komunikat podobny do poniższego:

```
tcp 0 0 0.0.0.0:389 0.0.0.0:* LISTEN 2845/slapd
tcp6 0 0 :::389 :::* LISTEN 2845/slapd
```

Teraz sprawdzimy czy możemy się połączyć z serwerem LDAP. Po wpisaniu polecenia:

```
stanowisko01:~/# ldapsearch -x -b "dc=przykladowy,dc=com"
```

powinniśmy zobaczyć rezultat jak poniżej:

```
# extended LDIF
#
# LDAPv3
# base <dc=przykladowy,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# przykladowy.com
dn: dc=przykladowy,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: test.com
dc: test
```



```
# admin, przykladowy.com
dn: cn=admin,dc=przykladowy,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Następnie proszę wpisać polecenie `slapcat,` które pozwala zobaczyć dodatkowe atrybuty nie pokazywane przy poleceniu `ldapsearch,` np. `userPassword`:

```
stanowisko01:~/# slapcat
```

```
dn: dc=przykladowy,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: przykladowy.com
dc: przykladowy
structuralObjectClass: organization
entryUUID: 350a2db6-87d3-102c-8c1c-1ffeac40db98
creatorsName:
modifiersName:
createTimestamp: 20080316183324Z
modifyTimestamp: 20080316183324Z
entryCSN: 20080316183324.797498Z#000000#000#000000

dn: cn=admin,dc=przykladowy,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e2NyeXB0fVdSZDJjRFdRODluNHM=
structuralObjectClass: organizationalRole
entryUUID: 350b330a-87d3-102c-8c1d-1ffeac40db98
creatorsName:
modifiersName:
createTimestamp: 20080316183324Z
modifyTimestamp: 20080316183324Z
entryCSN: 20080316183324.804398Z#000000#000#000000
```

2. Tworzenie podstawowej struktury:
   Dane w systemie LDAP grupowane są w strukturze przypominającej drzewo katalogów. Elementem najwyższego poziomu drzewa jest zazwyczaj nazwa domenowa. W przypadku domeny `przykladowy.com` będzie to `dc=przykladowy, dc=com`.

Na następnym poziomie stworzymy jednostki `People` i `Group`. Aby to zrobić proszę otworzyć w dowolnym edytorze tekstowym plik `/var/tmp/ou.ldif` i zmodyfikować go tak, aby zawierał wpisy podane poniżej:

```
dn: ou=People,dc=przykladowy,dc=com
ou: People
objectClass: organizationalUnit

dn: ou=Group,dc=przykladowy,dc=com
ou: Group
objectClass: organizationalUnit
```

Aby umieścić na serwerze plik proszę użyć polecenia **slapadd**:

```
stanowisko01:~/# invoke-rc.d slapd stop
stanowisko01:~/# slapadd -c -v -l /var/tmp/ou.ldif
stanowisko01:~/# invoke-rc.d slapd start
```

Następnie proszę użyć polecenia **ldapsearch,** które pozwoli sprawdzić czy wpisy zostały utworzone**:**

```
stanowisko01:~/# ldapsearch -x ou=people
```

Powinien ukazać się następujący komunikat:

```
# extended LDIF
#
# LDAPv3
# base <dc=przykladowy, dc=com> (default) with scope subt
# filter: ou=people
# requesting: ALL
#

# People, przykladowy.com
dn: ou=People,dc=przykladowy,dc=com
ou: People
objectClass: organizationalUnit

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```


2.10 ZADANIE NR 9 - TWORZENIE KONT UŻYTKOWNIKÓW W OPEN LDAP

Teraz przystąpimy do utworzenia grupy i użytkownika *uzytkownik1* należącego do niej.

Proszę utworzyć i edytować plik `/var/tmp/uzytkownik1.ldif`, a nastepnie wypełnić go następująco:

```
dn: cn=uzytkownik1,ou=group,dc=przykladowy,dc=com
cn: uzytkownik1
gidNumber: 20000
objectClass: top
objectClass: posixGroup

dn: uid=uzytkownik1,ou=people,dc=przykladowy,dc=com
uid: uzytkownik1
uidNumber: 20000
gidNumber: 20000
cn: Uzytkownik1
sn: Uzytkownik1
objectClass: top
objectClass: person
objectClass: posixAccount
objectClass: shadowAccount
loginShell: /bin/bash
homeDirectory: /home/uzytkownik1
```

Aby umieścic plik na serwerze proszę użyć polecenia **ldapadd**:

```
ldapadd -c -x -D cn=admin,dc=spinlock,dc=hr -W -f
            /var/tmp/uzytkownik1.ldif
```

Aby ustalić hasło nowego użytkownika proszę wpisać polecenie **ldappasswd**:

```
stanowisko01:~/# ldappasswd -x -D cn=admin,dc=przykladowy,dc=com
      -W -S uid=uzytkownik1,ou=people,dc=przykladowy,dc=com
```

Proszę użyć polecenia **ldapsearch,** aby sprawdzić czy wpis użytkownika został utworzony.

```
stanowisko01:~/# ldapsearch -x uid=uzytkownik1
```


Poniższy komunikat oznacza, iż udało się utworzyć i uruchomić konto dla użytkownika *uzytkownik1*:

```
# extended LDIF
#
# LDAPv3
# base <dc=przykladowy, dc=com> (default) with scope subtree
# filter: uid=uzytkownik1
# requesting: ALL
#

# uzytkownik1, people, przykladowy.com
dn: uid=uzytkownik1,ou=people,dc=przykladowy,dc=com
uid: uzytkownik1
uidNumber: 20000
gidNumber: 20000
cn: Uzytkownik1
sn: Uzytkownik1
objectClass: top

objectClass: person
objectClass: posixAccount
loginShell: /bin/bash
homeDirectory: /home/uzytkownik1

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Po wykonaniu powyższych poleceń proszę usunąć dodane wpisy z pliku `/var/tmp/ou.ldif` oraz `/var/tmp/uzytkownik1.ldif`.

2.11 ZAKOŃCZENIE ĆWICZENIA

Po zakończeniu ćwiczenia proszę usunąć wszystkie pliki i katalogi, które zostały utworzone podczas wykonywania ćwiczeń.

Jeżeli ćwiczenie wykonywane było zgodnie z instrukcją, na zakończenie ćwiczenia wystarczy uruchomic skrypt prepare.sh

```
stanowisko01:~# /labpam/skrypty/prepare.sh
```

2.12 OPRACOWANIE ĆWICZENIA I SPRAWOZDANIE

Wykonanie ćwiczenia polega na praktycznej realizacji wszystkich zadań **Rozdziału 2** niniejszej instrukcji zatytułowanego „**Przebieg Ćwiczenia**". Należy sporządzić sprawozdanie z wykonania ćwiczenia (w formie dokumentu elektronicznego) i w ciągu najdalej dwóch tygodni od dnia wykonania ćwiczenia oddać je prowadzącemu zajęcia.

Kompletne opracowanie ćwiczenia powinno zawierać:

- ✓ Część opisową odnoszącą się do teorii przerabianego ćwiczenia. Ta część sprawozdania powinna wykazać dobrą ogólną znajomość zagadnień leżących u podstaw przerabianego tematu, znajomość odnośnej literatury, samodzielność myślenia i umiejętność pisania opracowań o charakterze technicznym.

- ✓ Wnioski praktyczne wynikające z wykonania ćwiczenia, a w tym:

  - uwagi odnoszące się do przebiegu ćwiczenia (np. czy dane ćwiczenie może być wykonane z pełnym rozumieniem zawartych w nim czynności i problemów, czy ćwiczenie jest możliwe do wykonania w czasie przeznaczonym na zajęcia, czy ćwiczenie jest zbyt trudne/ zbyt łatwe, itp.,
  - uwagi odnoszące się do sposobu przygotowania i jakości (waloru dydaktycznego) instrukcji do ćwiczenia,
  - uwagi odnoszące się do ewentualnych utrudnień technicznych lub organizacyjnych pojawiających się w trakcie wykonywania ćwiczenia,
  - postulaty merytoryczne i techniczne dotyczące usprawnienia/ulepszenia jakości wykonywanego ćwiczenia,
  - inne

  Wnioski z drugiej części sprawozdania posłużą do usprawnienia i poprawy zajęć laboratoryjnych w latach następnych.

