

Budapesti Műszaki és Gazdaságtudományi Egyetem  
Intelligens rendszerfelügyelet (VIMIA370)

## **Címtárak**

1B Házi feladat

Ferencz Endre  
2010. márc. 11

# 1 Bevezető

A dokumentum célja, hogy átfogó képet adjon a házi feladat megoldásának menetéről. Bemutatom a kidolgozás során felhasznált ötleteket és megoldásokat.

## 1.1 A házi feladat célja

A házi feladat célja egy egyszerű címtárkezelő Bash szkript megírása.

## 1.2 A feladat specifikációja

Az árulók és kettősügynökök lebuktatása a családok számára létfontosságú feladat, de sajnos napjainkra a hagyományos módszereket már nem tudják alkalmazni. Az informatika térhódításával az elektronikus bűnözésben van a pénz, és ezen a területen is meg kell őrizni a családi érdekeltségeket. Az egymással vetélkedő Corleone és Barzini család már bevezette a központi címtárak használatát az érdekeltségi köreikben, hogy ezáltal optimalizálják a működési költségeiket. Egy szerencsétlen véletlen folytán a címtárszolgáltatást ugyanazon szolgáltatónál rendelték meg, és a két család összes adatát ugyanazon címtár tárolja. A szolgáltató arra figyelt, hogy ezen adatok külön részében helyezkedjenek el és egymás adatainak eléréséhez ne legyen lehetőségük.

Don Corleone visszautasíthatatlan ajánlata után a szolgáltató rendszergazdája örömmel adott hozzáférést a teljes címtárhoz, így lehetőség nyílik a két felhasználói adattár összevetésére. Feladatunk, hogy megkeressük azon felhasználókat, melyek mindkét családnak dolgoznak és így nemkívánatosnak minősülnek. A címtár felépítését vizsgálva megállapíthatjuk, hogy míg a Corleone család a klasszikus felhasználó és csoport struktúrát alkalmazza, úgy a Barzini család egyszerűen az LDAP hierarchikus felépítése alapján strukturálta az adatbázisát.

A feladat elvégzéséhez készítsünk egy olyan Bash szkriptet, ami megkeresi a problémás felhasználókat, majd elmenti az adataikat egy fájlba. A szkriptnek egy megadott csoportba tartozó összes felhasználót kell összevetnie egy megadott szervezeti egységben (Organizational Unit) tetszőlegesen mély hierarchiaszinten lévő összes felhasználóval és azokat kigyűjtenie, amelyek mindkét struktúrában szerepelnek.

## 1.3 A szkripttel kapcsolatos formai követelmények

A szkript elnevezése: *check-squealer.sh*.

A szkript paraméterei:

- CorleoneGroup: a Corleone család egy adott csoportjának DN-je, melyen belüli felhasználókat vizsgáljuk, kötelező megadni,
- BarziniOU: a Barzini család egy szervezeti egységének DN-je, melyen belüli felhasználókat vizsgáljuk, kötelező megadni,
- OutFile: a kimeneti fájl elérési útja, kötelező.

Kimenete: a kimeneti fájl egy CSV fájl a következő oszlopnevekkel: login, fullName, DN.

## 1.4 További elvárások

A szkript megírásával kapcsolatban a specifikáció további elvárásokat fogalmaz meg. Előírás, hogy ellenőrizsem a beadott paramétereket.

A megoldás során az LDAP címtárat új elemekkel szükséges bővíteni.

## 2 A feladat kidolgozása

A továbbiakban részletezem a feladat megoldásának lépéseit.

### 2.1 LDAP címtár bővítése

A megoldás elején létrehoztam öt új felhasználót a címtárban a virtuális gépen megtalálható phpldapadmin kezelőfelülete segítségével.

Az újonnan létrehozott felhasználók (1. ábra Az új felhasználók):

1. `cn=Csaba Pocsai,ou=users,ou=Corleone,ou=families,dc=thefamily,dc=local`
  - a. Csoportja: dons
  - b. Család: Corleone
2. `cn=David Magyar,ou=users,ou=Corleone,ou=families,dc=thefamily,dc=local`
  - a. Csoportja: sons
  - b. Család: Corleone
3. `cn=Laszlo Hasznos,ou=users,ou=Corleone,ou=families,dc=thefamily,dc=local`
  - a. Csoportja: allies
  - b. Család: Corleone
4. `cn=David Magyar,ou=hitmans,cn=Emilio Barzini, ou=Barzini, ou=families, dc=thefamily, dc=local`
  - a. Csoportja: hitmans
  - b. Család: Barzini
5. `cn=Laszlo Hasznos,ou=partners,cn=Emilio Barzini, ou=Barzini, ou=families, dc=thefamily, dc=local`
  - a. Csoportja: partners
  - b. Család: Barzini

A felhasználókat olyan helyekre és névvel próbáltam a címtárban elhelyezni, hogy a szkript eredményét minél teljesebben tudjam tesztelni.



1. ábra Az új felhasználók

## 2.2 A Bash szkript

A szkript elején ellenőrzöm a paraméterek formális megfelelőségét:

```
if [ $# -ne 3 ]; then
    echo 'Harom bemeno parametert varok!';
elif [ -f $3 ]; then
    echo 'Mar letezik a parameterben megadott fajl';
...

```

A további, összetettebb ellenőrzéseket, már hibakezeléssel tesztelem: a csoport és az OU létezésének ellenőrzése.

A következő lépésben a szkript által többször felhasznált változókat hozom létre, a később egyszerűség és átláthatóság érdekében:

```
corleone_dn="${1}";
barzini_dn="${2}";
output="${3}";

ldap_host="localhost";
ldap_user="cn=don,ou=administrative,dc=thefamily,dc=local";
ldap_pass="money";
ldap_root="dc=thefamily,dc=local";

```

Az első három utasítás egyszerűen csak átnevezi a kapott paramétereket, azok jelentése függvényében. A maradék négy paraméter az LDAP rendszerrel való kapcsolat paramétereit rögzíti. Ha később ezek módosítására volna szükség, akkor itt egyszerűen megtehető.

Ezután következik a szkript leglényegesebb eleme, az `ldapsearch` parancs.

```
ldap_cmd="ldapsearch -LLL -x -s base -h ${ldap_host} -D ${ldap_user} -b
${corleone_dn} member";

```

A paraméterek értelmezése:

- `-LLL` – a kimeneten csak a lényeges elemek jelennek meg, így egyszerűbb a feldolgozás
- `-x` – egyszerű bejelentkezési mód alkalmazása
- `-s base` – a lekérdezés mélysége, amely ebben az esetben csak az adott objektumot szeretném lekérdezni
- `-h ${ldap_host}` – az LDAP szerver címe
- `-D ${ldap_user}` – az adminisztrátor felhasználóneve
- `-w ${ldap_pass}` – az adminisztrátor jelszava
- `-b ${corleone_dn}` – a keresés kiindulópontja
- `member` – milyen attribútumokra vagyok kíváncsi

A lekérdezés eredményét `grep` parancs segítségével szűröm, mivel a megkülönböztető név ebben az esetben nem érdekes. A második `grep` segítségével kiszűröm a csoport tagjainak a nevét (common name).

```
result=$((${ldap_cmd} | grep "^member:" | grep -o "cn=[A-Za-z0-9 ]*");
```

Amennyiben az előző parancs hibával tér vissza (hibakód különbözik 0-tól), akkor valószínűleg hibás az első paraméter, ezért ezt jelzem a felhasználónak. Itt feltételeztem, hogy a kapcsolatban

felépítésével nem lehet probléma, de kapcsolathiba esetén is megjelenik az erre utaló angol nyelvű hibaüzenet.

```
if [ $? -ne 0 ]; then
    echo "Az elso parameterben megadott DN nem letezik."
    exit 1;
fi
```

Egy hasonló ellenőrzést végzek a második paraméteren (csoportazonosító) is Mivel ez szemantikailag azonos az előző ellenőrzéssel, ezért ezt nem részletezem ismételtten.

A következőkben a Corleone család megfelelő csoporttagjait végigellenőrzöm, hogy szerepelnek-e a Barzini család megjelölt csoportjában. Ezt egy for ciklus segítségével végzem.

```
OLDIFS=$IFS;
IFS=$'\n';
for line in ${result}; do
    ...
done
```

A ciklusmagban lekérdezem a Barzini család tagjait. Amennyiben találok egyezést a név (cn) alapján, akkor annak az adatait megjelenítem az eredményfájlban.

```
ldap_cmd2="ldapsearch -LLL -x -h ${ldap_host} -D ${ldap_user} -w
${ldap_pass} -b ";
filter="(${line})";

result2=$((${ldap_cmd2} "${barzini_dn}" "${filter}" uid | grep "uid: " | sed
's/uid: //g');
if [ "$result2" != "" ]; then
    echo -n "\"${result2}\", " >> ${output};
fi
```

A többi kért paramétert is ezzel analóg módon iratom ki a cél fájlba. Lehetséges lett volna egy lekérdezésben lehívni az összes adatot, de mivel van olyan attribútum, amelyik csak több sorban jelenik meg a kimeneten, nehezen lehetett volna a kimenetet megfelelően kezelni.

## 3 Tesztelés

Ebben a fejezetben az elkészült szkript működését fogom demonstrálni néhány teszteseten keresztül. A tesztesetek során az eredeti LDAP struktúrát használom, amelyen az előzőekben bemutatott változtatásokat végeztem el.

### 3.1 Első teszteset

A megadott paraméterek:

1. CorleoneGroup
  - "cn=sons,ou=groups,ou=Corleone,ou=families,dc=thefamily,dc=local"
2. BarziniOU
  - "ou=Barzini,ou=families,dc=thefamily,dc=local"
3. OutFile:

- ./tobeeliminated.csv

A kapott kimenet:

```
"magyari","David Magyar","cn=David Magyar,ou=hitmans,cn=Emilio Barzini,ou=Barzini,ou=families,dc=thefamily,dc=local"
```

A kapott kimenet megfelel az elvárásoknak.

### **3.2 Második teszt**

A megadott paraméterek:

1. CorleoneGroup
  - "cn=sons,ou=groups,ou=Corleone,ou=families,dc=thefamily,dc=local"
2. BarziniOU
  - "cn=Emilio Barzini,ou=Barzini,ou=families,dc=thefamily,dc=local"
3. OutFile:
  - ./tobeeliminated.csv

A kapott kimenet:

```
"magyari","David Magyar","cn=David Magyar,ou=hitmans,cn=Emilio Barzini,ou=Barzini,ou=families,dc=thefamily,dc=local"
```

A kapott kimenet megfelel az elvárásoknak.

### **3.3 Harmadik teszt**

A megadott paraméterek:

1. CorleoneGroup
  - "cn=sons,ou=groups,ou=Corleone,ou=families,dc=thefamily,dc=local"
2. BarziniOU
  - "ou=partners,cn=Emilio Barzini, ou=Barzini, ou=families, dc=thefamily, dc=local"
3. OutFile:
  - ./tobeeliminated.csv

A kapott kimenet üres.

A kapott kimenet megfelel az elvárásoknak.

### **3.4 Negyedik teszt**

A megadott paraméterek:

4. CorleoneGroup
  - "cn=sons,ou=groups,ou=Corleone,ou=families,dc=thefamily,dc=local"
5. BarziniOU
  - "ou=Barzini,ou=families,dc=thefamily,dc=local"
6. OutFile:
  - ./tobeeliminated.csv

A kapott kimenet:

```
"luca","Luca Brasi","cn=Luca Brasi, ou=hitmans, cn=Emilio Barzini, ou=Barzini, ou=families, dc=thefamily, dc=local"
```

```
"carlo","Carlo Rizzi","cn=Carlo Rizzi, ou=hitmans, cn=Emilio Barzini, ou=Barzini, ou=families, dc=thefamily, dc=local"
```

```
"laszlo","Laszlo Hasznos","cn=Laszlo Hasznos, ou=partners,cn=Emilio  
Barzini, ou=Barzini, ou=families, dc=thefamily, dc=local"
```

A kapott kimenet megfelel az elvárásoknak.

## **4 Összefoglaló**

A feladat megoldása során felhasználtam az Intelligens Rendszerfelügyelet című tárgyban elsajátított ismereteket és megoldottam egy olyan feladatot, melynek elemei megjelenhetnek a valós életben is. A megoldásom több tesztet is kipróbáltam, ezért feltételezem, hogy az esetek túlnyomó többségében megfelelően működik.