Sessão 9: LDAP

1) Instalação do servidor OpenLDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

Instale e configure um servidor LDAP na máquina *Server_Linux* (192.168.0.10). Você deverá instalar os seguintes pacotes: `slapd, ldap-utils, migrationtools, attr, libpam-ldap, libnss-ldap, nscd`.

Tabela 1. Configuração libpam-ldap e libnss-ldap

Parâmetro	**Valor**
LDAP URI	`ldap://127.0.0.1`
Search base	`dc=empresa,dc=com,dc=br`
LDAP Admin	`cn=admin,dc=empresa,dc=com,dc=br`
LDAP Admin como usuário `root` local	Sim
LDAP requer autenticação	Não
Versão do LDAP	3

Após a instalação e configuração inicial, execute o comando `# dpkg-reconfigure slapd`.

Tabela 2. Configuração do slapd

Parâmetro	**Valor**
Omitir configuração LDAP	Não
Nome DNS	empresa.com.br
Nome da Organização	Empresa
Backend	MDB
Remover base atual em caso de `purge`	Não
Mover base de dados antiga	Sim
Permitir LDAPv2	Não

Finalmente, edite o arquivo `/etc/ldap/ldap.conf` e edite os parâmetros `BASE` e `URI` de acordo com o configurado nesta atividade. Reinicie o servidor LDAP e verifique se está operacional — faça uma consulta-teste usando o comando `ldapsearch`.

1. Instale o OpenLDAP e programas auxiliares que serão utilizados:

```
# apt-get install slapd ldap-utils migrationtools attr libpam-ldap libnss-ldap nscd
```

2. Reconfigure o pacote `slapd`, respondendo as perguntas de acordo com o exposto na tabela acima.

```
# dpkg-reconfigure slapd
```

3. Inicie o *daemon* `slapd` e verifique seu estado.

```
# systemctl start slapd

# systemctl status slapd
● slapd.service - LSB: OpenLDAP standalone server (Lightweight Directory Access
Protocol)
   Loaded: loaded (/etc/init.d/slapd)
   Active: active (running) since Sex 2018-08-10 15:06:46 -03; 3s ago
  Process: 5095 ExecStop=/etc/init.d/slapd stop (code=exited, status=0/SUCCESS)
  Process: 6128 ExecStart=/etc/init.d/slapd start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/slapd.service
           └─6133 /usr/sbin/slapd -h ldap:/// ldapi:/// -g openldap -u openldap
-F /etc/ld...

Ago 10 15:06:46 servidor systemd[1]: Starting LSB: OpenLDAP standalone server
(Lightwei......
Ago 10 15:06:46 servidor slapd[6132]: @(#) $OpenLDAP: slapd  (Jun 14 2018 21:56:48)
$
                                                buildd@x86-csail-01:/build/openldap-
Yko3W...apd
Ago 10 15:06:46 servidor slapd[6133]: slapd starting
Ago 10 15:06:46 servidor systemd[1]: Started LSB: OpenLDAP standalone server
(Lightweig...l).
Ago 10 15:06:46 servidor slapd[6128]: Starting OpenLDAP: slapd.
Hint: Some lines were ellipsized, use -l to show in full.
```

4. Edite o arquivo `/etc/ldap/ldap.conf` com os valores apropriados:

```
# grep '^BASE\|^URI' /etc/ldap/ldap.conf
BASE    dc=empresa,dc=com,dc=br
URI     ldap://127.0.0.1
```

5. Finalmente, consulte a raiz da *search base* do diretório para verificar seu funcionamento.

```
# ldapsearch -x -b 'dc=empresa,dc=com,dc=br' -s base '(ObjectClass=*)'
# extended LDIF
#
# LDAPv3
# base <dc=empresa,dc=com,dc=br> with scope baseObject
# filter: (ObjectClass=*)
# requesting: ALL
#

# empresa.com.br
dn: dc=empresa,dc=com,dc=br
objectClass: top
objectClass: dcObject
objectClass: organization
o: Empresa
dc: empresa

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

2) Usando o *migrationtools*

Esta configuração será realizada na máquina virtual *Server_Linux*.

O *migrationtools* é um conjunto de *scripts* que permite importar as contas locais de um sistema Linux para um diretório LDAP, que já foi instalado na máquina *Server_Linux* (192.168.0.10) durante a atividade 1.

1. Edite o arquivo `/etc/migrationtools/migrate_common.ph`, substituindo as variáveis `$DEFAULT_MAIL_DOMAIN` e `$DEFAULT_BASE` pelos valores configurados na atividade anterior.

```
# grep '^\$DEFAULT_' /etc/migrationtools/migrate_common.ph
$DEFAULT_MAIL_DOMAIN = "empresa.com.br";
$DEFAULT_BASE = "dc=empresa,dc=com,dc=br";
```

2. Entre no diretório `/usr/share/migrationtools` e execute os *scripts* `migrate_base.pl`, `migrate_passwd.pl` e `migrate_group.pl` para exportar as bases (respectivamente) geral, de usuários/senhas e de grupos. Atente-se para a sintaxe de uso de cada *script*.

```
# cd /usr/share/migrationtools
# ./migrate_base.pl > /root/base.ldif
# ./migrate_passwd.pl /etc/passwd /root/passwd.ldif
# ./migrate_group.pl /etc/group /root/group.ldif
```

3. Remova os registros `dc=com,dc=br` e `dc=empresa,dc=com,dc=br` do topo do arquivo gerado pelo *script* `migrate_base.pl`, que já foram incluídos no diretório LDAP na primeira atividade.

```
# sed -i '/dn: dc=com,dc=br/,/^$/d' /root/base.ldif
# sed -i '/dn: dc=empresa,dc=com,dc=br/,/^$/d' /root/base.ldif

# head -n1 /root/base.ldif
dn: ou=Networks,dc=empresa,dc=com,dc=br
```

4. Adicione os arquivos `.ldif` gerados anteriormente à base LDAP usando o comando `ldapadd`. Consulte sua página de manual para descobrir as opções apropriadas a passar para o comando. Lembre-se, apenas, que o diretório LDAP está utilizando autenticação simples, não SASL, e que é necessário informar um DN administrativo e senha para inserção de dados.

```
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /root/base.ldif
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /root/passwd.ldif
# ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' < /root/group.ldif
```

5. Use o comando `ldapsearch` juntamente com um filtro de pesquisa apropriado para listar todos os grupos que foram adicionados ao diretório LDAP pelos arquivos `.ldif` incluídos no passo anterior.

```
# ldapsearch -x -b 'dc=empresa,dc=com,dc=br' '(&(cn=*)(objectClass=posixGroup))'
# extended LDIF
#
# LDAPv3
# base <dc=empresa,dc=com,dc=br> with scope subtree
# filter: (&(cn=*)(objectClass=posixGroup))
# requesting: ALL
#

# root, Group, empresa.com.br
dn: cn=root,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: root
gidNumber: 0

(...)

# openldap, Group, empresa.com.br
dn: cn=openldap,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: openldap
gidNumber: 117

# search result
search: 2
result: 0 Success

# numResponses: 58
# numEntries: 57
```

3) Configurança do cliente Linux para uso do LDAP

Esta configuração será realizada na máquina virtual *Client_Linux*.

Para que as clientes Linux possam se autenticar na base de dados do LDAP, é necessário configurar o PAM (*Pluggable Authentication Modules*) e NSS (*Name Service Switch*) para consultarem logins junto ao servidor LDAP.

Configure a máquina *Client_Linux* (192.168.0.20) para se autenticar na base LDAP que está instalada na máquina *Server_Linux* (192.168.0.10). Você deverá instalar os seguintes pacotes: `ldap-utils`, `libpam-ldap`, `libnss-ldap`, `nscd`.

Tabela 3. Configuração libpam-ldap e libnss-ldap no **Client_Linux**

Parâmetro	**Valor**
LDAP URI	`ldap://192.168.0.10`

Parâmetro	Valor
Search base	`dc=empresa,dc=com,dc=br`
LDAP Admin	`cn=admin,dc=empresa,dc=com,dc=br`
LDAP Admin como usuário `root` local	Sim
LDAP requer autenticação	Não
Versão do LDAP	3

Não se esqueça de editar os arquivos `/etc/ldap/ldap.conf` e `/etc/nsswitch.conf` para habilitar consulta às bases do LDAP durante procedimentos de login.

Se desejar que diretórios *home* sejam criados automaticamente para usuários LDAP inexistentes na máquina local, insira a linha a seguir ao final do arquivo `/etc/pam.d/common-password`:

```
session optional pam_mkhomedir.so skel=/etc/skel/ umask=022
```

Finalmente, para reiniciar a *cache* de usuários e grupos do LDAP, execute `# systemctl restart nscd`. Se houver algum registro de erro nos arquivos de log quanto à inexistência do arquivo `/etc/netgroup`, crie-o manualmente.

1. Instale os *plugins* LDAP para as bibliotes PAM e NSS, bem como programas auxiliares que serão utilizados:

   ```
   # apt-get install ldap-utils libpam-ldap libnss-ldap nscd
   ```

2. Verifique se os arquivos das bibliotecas PAM e NSS foram configurados automaticamente de forma correta pelo gerenciador de pacotes:

   - `/etc/libnss-ldap.conf`:

     ```
     # cat /etc/libnss-ldap.conf | grep -v '^#' | sed '/^$/d'
     base dc=empresa,dc=com,dc=br
     uri ldap://192.168.0.10
     ldap_version 3
     rootbinddn cn=admin,dc=empresa,dc=com,dc=br
     ```

   - `/etc/libnss-ldap.secret`:

     ```
     # cat /etc/libnss-ldap.secret
     rnpesr
     ```

   - `/etc/pam_ldap.conf`:

```
# cat /etc/pam_ldap.conf | grep -v '^#' | sed '/^$/d'
base dc=empresa,dc=com,dc=br
uri ldap://192.168.0.10
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
pam_password crypt
```

- `/etc/pam_ldap.secret`:

```
# cat /etc/pam_ldap.secret
rnpesr
```

3. Insira as informações sobre o servidor LDAP no arquivo `/etc/ldap/ldap.conf`:

```
# cat /etc/ldap/ldap.conf | grep -v '^#' | sed '/^$/d'
BASE    dc=empresa,dc=com,dc=br
URI     ldap://192.168.0.10
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt
```

4. Configure o `nsswitch` para consultar as bases LDAP em adição às bases locais de usuários e senhas. Se desejar que as bases do LDAP tenham preferência, coloque a palavra-chave `ldap` à frente da palavra `compat`.

```
# cat /etc/nsswitch.conf | grep -v '^#'

passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
gshadow:        files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

5. Para que diretórios *home* sejam criados de forma automática se usuários LDAP inexistentes na máquina local logarem no sistema, insira a linha a seguir ao final do arquivo `/etc/pam.d/common-password`:

```
# tail -n1 /etc/pam.d/common-session
session optional pam_mkhomedir.so skel=/etc/skel/ umask=022
```

6. Adiantando-se ao problema futuro que em que o *daemon* `nscd` irá apontar a inexistência do arquivo `/etc/netgroup`, crie-o, vazio:

```
# touch /etc/netgroup
```

7. Finalmente, reinicie o *daemon* `nscd` e verifique se os usuários e grupos remotos do servidor LDAP estão sendo utilizados pelos subsistemas de autenticação local:

```
# systemctl restart nscd

# grep '^openldap:' /etc/passwd
# getent passwd | grep '^openldap:'
openldap:x:111:117:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false

# grep '^openldap:' /etc/group
# getent group | grep '^openldap:'
openldap:x:117:
```

   Observe, acima, que tanto o usuário quanto o grupo `openldap` estão disponíveis na máquina local, muito embora não existam nos arquivos `/etc/passwd` e `/etc/group`. Eles estão sendo obtidos, remotamente, no servidor LDAP *Server_Linux*.

4) Configurança do servidor Linux para uso do LDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

Agora que a máquina *Client_Linux* (192.168.0.20) está configurada para se autenticar na base LDAP remota localizada na máquina *Server_Linux* (192.168.0.10), faça com que o próprio servidor *Server_Linux* autentique-se usando sua base LDAP local.

1. Essencialmente, basta repetir os passos do exercício anterior, desta vez na máquina *Server_Linux*:

```
# hostname
servidor

# cat /etc/libnss-ldap.conf | grep -v '^#' | sed '/^$/d'
base dc=empresa,dc=com,dc=br
uri ldap://127.0.0.1
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
```

```
# cat /etc/libnss-ldap.secret
rnpesr

# cat /etc/pam_ldap.conf | grep -v '^#' | sed '/^$/d'
base dc=empresa,dc=com,dc=br
uri ldap://127.0.0.1
ldap_version 3
rootbinddn cn=admin,dc=empresa,dc=com,dc=br
pam_password crypt

# cat /etc/pam_ldap.secret
rnpesr

# cat /etc/ldap/ldap.conf | grep -v '^#' | sed '/^$/d'
BASE    dc=empresa,dc=com,dc=br
URI     ldap://127.0.0.1
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt

# cat /etc/nsswitch.conf | grep -v '^#'

passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
gshadow:        files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis

# tail -n1 /etc/pam.d/common-session
session optional pam_mkhomedir.so skel=/etc/skel/ umask=022

# ls -ld /etc/netgroup
-rw-r--r-- 1 root root 0 Ago 12 17:20 /etc/netgroup

# systemctl restart nscd
```

5) Criação e remoção de usuários e grupos LDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

Agora que a máquina *Client_Linux* está conectada ao servidor LDAP, adicione um novo usuário e grupo associado, ambos com o mesmo nome, e faça login com o usuário. Para realizar essa tarefa,

crie arquivos LDIF manualmente e adicione-os via `ldapadd`. Não esqueça de definir a senha através do comando `ldappasswd`.

Observação: Para evitar confusões entre a base de usuários do LDAP e a base local dos clientes, é recomendável adotar um *buffer* numérico entre os usuários locais e os usuários do diretório. Faça com que o UID e GID dos novos usuários/grupos comece a partir de 5000.

1. Primeiro, vamos verificar o formato da entrada de diretório LDAP para um usuário existente:

```
# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br' '(uid=aluno)'
dn: uid=aluno,ou=People,dc=empresa,dc=com,dc=br
uid: aluno
cn: aluno
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/aluno
gecos: aluno,,,
```

2. Vamos adicionar o usuário `esr`. Para isso, basta processar a saída do comando acima, substituir com os dados do novo usuário, e enviar como entrada para o comando `ldapadd`, como se segue:

```
# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br' '(uid=aluno)' | sed 's/aluno/esr/
; s/1000/5000/' | ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br'
Enter LDAP Password:
adding new entry "uid=esr,ou=People,dc=empresa,dc=com,dc=br"

# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br' '(uid=esr)'
dn: uid=esr,ou=People,dc=empresa,dc=com,dc=br
uid: esr
cn: esr
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 5000
gidNumber: 5000
homeDirectory: /home/esr
gecos: esr,,,
```

3. Excelente! Vamos fazer o mesmo para o grupo:

```
# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br'
'(&(cn=aluno)(objectClass=posixGroup))' | sed 's/aluno/esr/ ; s/1000/5000/' |
ldapadd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br'
Enter LDAP Password:
adding new entry "cn=esr,ou=Group,dc=empresa,dc=com,dc=br"

# ldapsearch -x -LLL -b 'dc=empresa,dc=com,dc=br'
'(&(cn=esr)(objectClass=posixGroup))'
dn: cn=esr,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: esr
gidNumber: 5000
```

4. Ainda falta configurar a senha do novo usuário. Vamos fazer isso através do comando `ldappasswd`:

```
# ldappasswd -x -W -D 'cn=admin,dc=empresa,dc=com,dc=br' -S
"uid=esr,ou=People,dc=empresa,dc=com,dc=br"
New password:
Re-enter new password:
Enter LDAP Password:

# ldapsearch -x -LLL -W -D 'cn=admin,dc=empresa,dc=com,dc=br' -b
'dc=empresa,dc=com,dc=br' '(uid=esr)' userPassword
Enter LDAP Password:
dn: uid=esr,ou=People,dc=empresa,dc=com,dc=br
userPassword:: e1NTSEF9ZW1YWkFBQVNWWEh1a2kwQmVRbzdMdkNzSGp0cm9VOEc=
```

5. De volta à máquina *Client_Linux*, vamos verificar se o novo usuário está sendo importado corretamente:

```
# hostname
cliente

# getent passwd | grep '^esr:'
esr:x:5000:5000:esr,,,:/home/esr:/bin/bash
# getent group | grep '^esr:'
esr:*:5000:
```

6. Agora, basta fazer login com o usuário e testar se a criação automática de diretório *home* está funcionando:

```
$ ssh esr@192.168.0.20
esr@192.168.0.20's password:
Creating directory '/home/esr'.

$ hostname
cliente

$ whoami
esr

$ pwd
/home/esr
```

6) Criação e deleção automática de usuários LDAP

Esta configuração será realizada na máquina virtual *Server_Linux*.

O esquema de criação de usuários manualmente acima funcionou, como visto. Não é, no entanto, muito conveniente do ponto de vista de manutenção do sistema proceder dessa forma. Seria mais interessante, se possível, automatizar essa tarefa para facilitar sua execução no dia-a-dia.

Crie um *script* que faça a adição e deleção automática de usuários na base LDAP. Atente-se para o fato de que os UIDs e GIDs desses usuários não devem se confundir com o dos sistemas locais. Use o valor mínimo de 5000 para ambos.

1. O *script shell* abaixo mostra um exemplo de solução para o problema proposto:

```
#!/bin/bash


tldap_user() {
  qlu=$( ldapsearch -x -LLL -b "dc=empresa,dc=com,dc=br" "(uid=${1})" | grep
"^uid:" | awk '{print $2}' )
  [ ! -z $qlu ] && return 1 || return 0
}


# $1 ldap_admin, $2 ldap_password, $3: user, $4: pass
r_adduser() {
  if ! tldap_user $3; then
    echo "  [*] LDAP user exists!"
    exit 1
  fi

  lastuid=$( ldapsearch -x -LLL
'(&(objectClass=posixAccount)(uid=*)(!(uid=nobody)))' uidNumber | grep
'^uidNumber:' | awk '{print $2}' | sort -n | tail -n1 )
```

```
  lastgid=$( ldapsearch -x -LLL '(&(objectClass=posixGroup)(cn=*)(!(cn=nogroup)))'
gidNumber | grep  '^gidNumber:' | awk '{print $2}' | sort -n | tail -n1 )

  ((lastuid++))
  ((lastgid++))

  ldapadd -x -D $1 -w $2 << EOF
dn: uid=$3,ou=People,dc=empresa,dc=com,dc=br
uid: $3
cn: $3
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: $lastuid
gidNumber: $lastgid
homeDirectory: /home/$3
gecos: $3,,,

EOF

  ldapadd -x -D $1 -w $2 << EOF
dn: cn=$3,ou=Group,dc=empresa,dc=com,dc=br
objectClass: posixGroup
objectClass: top
cn: $3
gidNumber: $lastgid

EOF

  ldappasswd -x -D $1 -w $2 -s $4 "uid=$3,ou=People,dc=empresa,dc=com,dc=br"
}


# $1 ldap_admin, $2 ldap_password, $3: user
r_deluser() {
  if tldap_user $3; then
    echo "  [*] LDAP user does not exist!"
    exit 1
  fi

  ldapdelete -x -D $1 -w $2 "uid=$3,ou=People,dc=empresa,dc=com,dc=br"
  ldapdelete -x -D $1 -w $2 "cn=$3,ou=Group,dc=empresa,dc=com,dc=br"
}


usage() {
  echo "  Usage: $0 -l LDAP_ADMIN -w LDAP_PASSWD -u USER [-a|-d] [-p PASSWD]"
```

```
  exit 1
}


# - - - main() - - -


if [[ $EUID -ne 0 ]]; then
  echo "  [*] Not root!" 1>&2
  exit 1
fi

while getopts ":adu:p:l:w:" opt; do
  case "$opt" in
    l)
      ladmin=${OPTARG}
      ;;
    w)
      lpass=${OPTARG}
      ;;
    u)
      user=${OPTARG}
      ;;
    p)
      pass=${OPTARG}
      ;;
    a)
      uadd=1
      ;;
    d)
      udel=1
      ;;
    *)
      usage
      ;;
  esac
done

[ -z $ladmin ] && { echo "  [*] No LDAP admin?"; usage; }
[ -z $lpass ] && { echo "  [*] No LDAP password?"; usage; }
[ -z $user ] && { echo "  [*] No user?"; usage; }

if [ -z $uadd ] && [ -z $udel ]; then
  echo "  [*] Choose '-a' (add) or '-d' (delete)."
  usage
elif (($uadd)) && (($udel)); then
  echo "  [*] Do not use '-a' (add) and '-d' (delete) simultaneously."
  usage
fi

if (($uadd)) && [ -z $pass ]; then
```

```
  echo "  [*] '-p' (password) mandatory with '-a' (add)."
  usage
fi

(($uadd)) && r_adduser $ladmin $lpass $user $pass
(($udel)) && r_deluser $ladmin $lpass $user
```

Observe que apesar de o *script* ser relativamente complexo, ele ainda não está completo — falta tratar a adição de usuários a grupos secundários no LDAP, bem como sua remoção desses grupos quando de sua deleção.