

LABORATOIRE VII – CEG 4399



uOttawa

CEG 4399- Design of Secure Computer Sys.

Université d'Ottawa

Professeur : Amir H. Razavi

Noms et numéros des étudiants :
Gbegbe Decaho Jacques 300094197

Date de soumission: 07 November 2024

LDAP

Introduction

1 Overview

This lab illustrates the use of LDAP to authenticate users of Linux systems, such that multiple computers share a single repository of user and group information, including the passwords that authenticate users. This strategy allows users and administrators to manage a single set of credentials that can then be used to access multiple computers.

1.1 Background

The student is expected to have separately learned about the basic elements of Linux users, groups and authentication, e.g., the `/etc/passwd` and `/etc/shadow` files. The student is also expected to have a basic knowledge of the use of Lightweight Directory Access Protocol (LDAP).

The student is expected to have some familiarity with the Linux command line, the basics of the file system, and the ability to locate and edit a file. And some experience with the Wireshark tool is expected (e.g., the `wireshark-intro` lab).

2 Lab Environment

This lab runs in the Labtainer framework, available at <http://my.nps.edu/web/c3o/labainers>. That site includes links to a pre-built virtual machine that has Labainers installed, however Labainers can be run on any Linux host that supports Docker containers.

From your labtainer-student directory start the lab using:

```
labtainer ldap
```

A link to this lab manual will be displayed.

We loaded the “**ldap**” labtainer into the virtualbox in order to launch our working environment.

```
student@Labtainer-VirtualBox:~/labtainer/labtainer-student$ labtainer ldap
latest: Pulling from labainers/ldap.ldap.student
e726a4a269e5: Pull complete
e2e5582ec5ed: Downloading [=====] 20.29MB/22.9MB
4804c6594371: Download complete
b558ee5371d8: Download complete
d5e966fac7c1: Download complete
```

3 Network Configuration

This lab includes a client computer, two servers and an ldap server shown in Figure 1. When the lab starts, you will get one virtual terminal connected to the client, and one connected to the ldap server.

The host names of each component are per the diagram. The `/etc/hosts` files allow use of these host names instead of explicit ip addresses.

The two Linux servers have been configured to use the ldap server to authenticate users. The ldap server has been initially configured with a single user whose ID is “mike”.

The ldap server is configured for the “example.com” domain, with an ldap administrator of “admin” whose password is “adminpass”

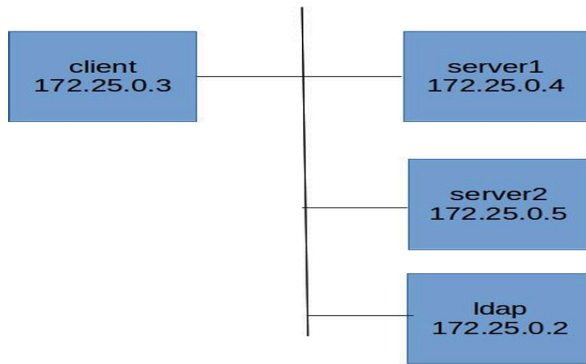


Figure 1: Network topology for the LDAP lab

The different identifiers and passwords defined by default are:

- **mike: password123**
- **admin: adminpass**

1. Lab Tasks

4.1 Explore

On the ldap server, display the ldap directory content using:

```
ldapsearch -x | less
```

and observe the entries in the directory. Note entry for “mike” and “projx”.

Start wireshark on the ldap component so that you can observe the protocol traffic.

```
wireshark &
```

Select the eth0 device. From the “client” computer, ssh to server1 as user “mike”:

```
ssh mike@server1
```

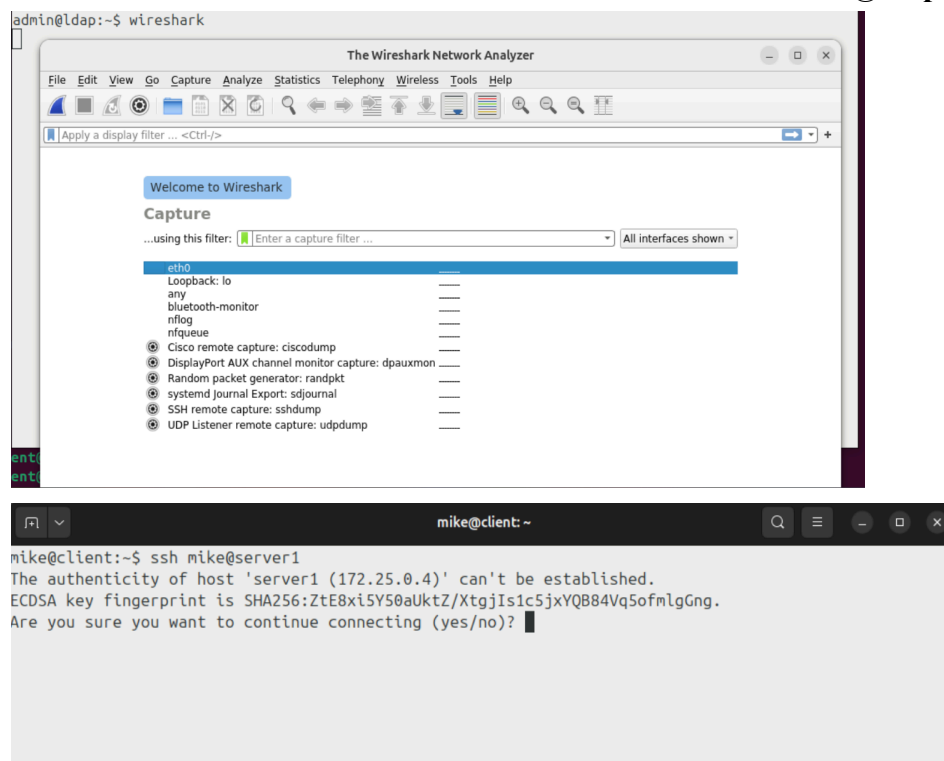
The initial password for “mike” is “password123”. The system will require that you change this password and then you will need to ssh again into server1. Change the password to whatever you like, but remember it. Use `ssh` again to login to server1 as mike, providing your new password. Use the `id` command to view your user ID and group. Then, view the `/etc/passwd` file. Do you see entries for your user or group?

After running the command “**ldapsearch -x | less**”, we observed and captured the parts in the directory content showing the details of “**projx**” and “**mike**”.

```
# projx, groups, example.com
dn: cn=projx,ou=groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
gidNumber: 1500
cn: projx
```

```
# mike, users, example.com
dn: uid=mike,ou=users,dc=example,dc=com
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: mike
uid: mike
uidNumber: 1501
gidNumber: 1500
homeDirectory: /home/mike
loginShell: /bin/bash
gecos: mike
shadowLastChange: 0
shadowMax: 0
shadowWarning: 0
```

Then we connected to the **wireshark** from the console of “**admin@ldap**”.



In the “**mike@client**” window, we executed the command “**ssh mike@server1**” in order to connect “**mike**” to server 1. Above, we entered the password “**password123**” in order to validate the authorization to connect to the server.

```
mike@client:~$ ssh mike@server1
The authenticity of host 'server1 (172.25.0.4)' can't be established.
ECDSA key fingerprint is SHA256:ZtE8xi5Y50aUktZ/XtgjIs1c5jxYQB84Vq5ofnlgGng.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1,172.25.0.4' (ECDSA) to the list of known hosts.
mike@server1's password:
You are required to change your password immediately (administrator enforced)
Creating directory '/home/mike'.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 6.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

WARNING: Your password has expired.
You must change your password now and login again!
Enter login(LDAP) password:
```

Above we see that "**mike**" is finally logged in to **server 1**.
At the bottom we see a warning asking "**mike**" to change the password because it is old.

```
WARNING: Your password has expired.
You must change your password now and login again!
Enter login(LDAP) password:
LDAP Password incorrect: try again
Enter login(LDAP) password:
New password:
Re-enter new password:
LDAP password information changed for mike
passwd: password updated successfully
Connection to server1 closed.
mike@client:~$
```

Above, we have changed the password of "**mike**" to "**mike123**".
We have checked the values of the account of "**mike**" present in the directory. As shown in the screen below.

```
mike@server1:~$ id
uid=1501(mike) gid=1500(projx) groups=1500(projx)
mike@server1:~$

mike@client:~$ ssh mike@server1
mike@server1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 6.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Tue Oct 29 16:05:56 2024 from 172.25.0.3
mike@server1:~$
```

Now in order to verify that everything is “OK” we reconnected to **server 1** as “mike”.

No.	Time	Source	Destination	Protocol	Length	Info
316	566.291360793	172.25.0.2	172.25.0.4	LDAP	80	searchResDone(7) success [2 results]
317	566.291400286	172.25.0.4	172.25.0.2	TCP	66	34188 → 389 [ACK] Seq=881 Ack=637 Win=31872 Len=0 T
318	566.291515031	172.25.0.4	172.25.0.2	LDAP	208	searchRequest(8) "dc=example,dc=com" wholeSubtree
319	566.292133326	172.25.0.2	172.25.0.4	LDAP	117	searchResDone(8) success [2 results]
320	566.332972076	172.25.0.4	172.25.0.2	TCP	66	34188 → 389 [ACK] Seq=943 Ack=688 Win=31872 Len=0 T
321	571.284005459	02:42:ac:19:00:02	02:42:ac:19:00:05	ARP	42	Who has 172.25.0.5? Tell 172.25.0.2
322	571.284156665	02:42:ac:19:00:05	02:42:ac:19:00:02	ARP	42	172.25.0.5 is at 02:42:ac:19:00:05
323	571.796861712	02:42:ac:19:00:02	02:42:ac:19:00:04	ARP	42	Who has 172.25.0.4? Tell 172.25.0.2
324	571.797094119	02:42:ac:19:00:04	02:42:ac:19:00:02	ARP	42	Who has 172.25.0.2? Tell 172.25.0.4
325	571.797021894	02:42:ac:19:00:02	02:42:ac:19:00:04	ARP	42	172.25.0.2 is at 02:42:ac:19:00:02
326	571.797024458	02:42:ac:19:00:04	02:42:ac:19:00:02	ARP	42	172.25.0.4 is at 02:42:ac:19:00:04
327	817.557349268	fe80::44c3:b2ff:fe7... ff02::2		ICMPv6	70	Router Solicitation from 46:c3:b2:7a:9b:5e

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
Ethernet II, Src: 02:42:ac:19:00:03 (02:42:ac:19:00:03), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

0000 ff ff ff ff ff ff 02 42 ac 19 00 03 08 06 00 01B
0010 08 00 06 04 00 01 02 42 ac 19 00 03 ac 19 00 03B
0020 00 00 00 00 00 00 ac 19 00 04

In **wireshark**, we observe the **different threads** active on the **network**.

4.2 View protocol traffic

Go to the wireshark window, and stop capturing packets (e.g., the red stop button). Enter a display filter of “ldap”, i.e., near the top where it says “Apply a display filter...”. Review the LDAP traffic. Which components are exchanging packets? Locate the packet that changed mike’s password and use File / Export Specified Packets to save that packet in a file named password.pcapng

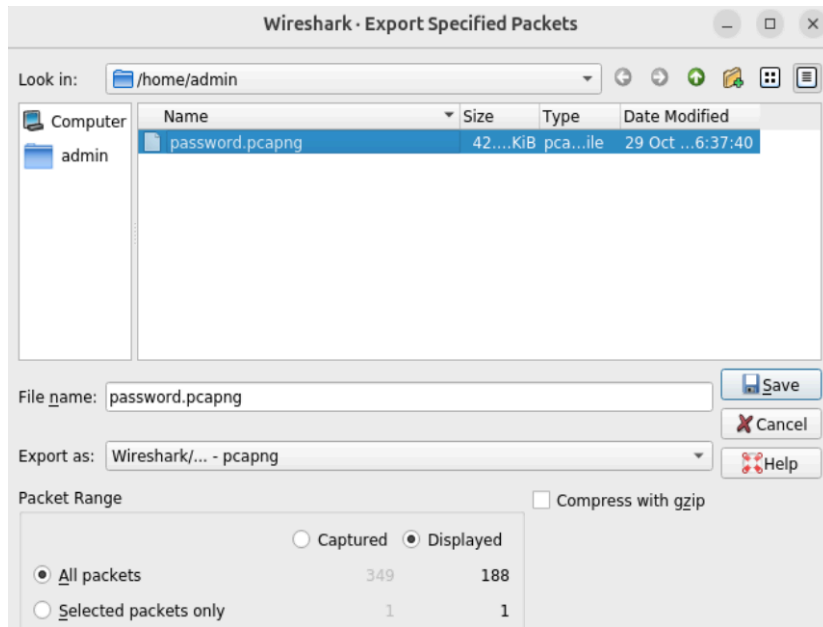
We applied a filter to the wireshark to identify only the threads linked to the “ldap”.

No.	Time	Source	Destination	Protocol	Length	Info
161	143.788718091	172.25.0.4	172.25.0.2	LDAP	115	bindRequest(4) "cn=admin,dc=example,dc=com" simple
162	143.790773771	172.25.0.2	172.25.0.4	LDAP	80	bindResponse(4) success
168	153.931918741	172.25.0.4	172.25.0.2	LDAP	126	bindRequest(5) "uid=mike,ou=users,dc=example,dc=com" simple
169	153.932615002	172.25.0.2	172.25.0.4	LDAP	80	bindResponse(5) success
171	153.933696434	172.25.0.4	172.25.0.2	LDAP	115	bindRequest(6) "cn=admin,dc=example,dc=com" simple
172	153.935415746	172.25.0.2	172.25.0.4	LDAP	80	bindResponse(6) success
174	165.082473899	172.25.0.4	172.25.0.2	LDAP	178	modifyRequest(7) "uid=mike,ou=users,dc=example,dc=com"
175	165.091827944	172.25.0.2	172.25.0.4	LDAP	80	modifyResponse(7) success
177	165.091724730	172.25.0.4	172.25.0.2	LDAP	146	modifyRequest(8) "uid=mike,ou=users,dc=example,dc=com"
178	165.102486938	172.25.0.2	172.25.0.4	LDAP	80	modifyResponse(8) success
179	165.103896080	172.25.0.4	172.25.0.2	LDAP	73	unbindRequest(9)
186	165.130042549	172.25.0.4	172.25.0.2	LDAP	260	searchRequest(18) "dc=example,dc=com" wholeSubtree
187	165.137212255	172.25.0.2	172.25.0.4	LDAP	385	searchResEntry(18) "uid=mike,ou=users,dc=example,dc=com"
188	165.138190824	172.25.0.2	172.25.0.4	LDAP	80	searchResDone(18) success [18 results]

Internet Protocol Version 4, Src: 172.25.0.4, Dst: 172.25.0.2
Transmission Control Protocol, Src Port: 34216, Dst Port: 389, Seq: 568, Ack: 232, Len: 80
Lightweight Directory Access Protocol
LDAPMessage modifyRequest(8) "uid=mike,ou=users,dc=example,dc=com"
messageID: 8
protocolOp: modifyRequest (6)
modifyRequest
[Response In: 178]

0000 02 42 ac 19 00 02 02 42 ac 19 00 04 08 00 45 00 ..B....B.....E..
0010 00 84 fb 0e 40 00 40 06 e7 2c ac 19 00 04 ac 19@.....
0020 00 02 85 a8 01 85 1d 69 26 0c b1 e3 ac 7e 80 18i&.....
0030 00 fa 58 af 00 00 01 01 08 8a 18 b2 43 ee 64 31 ..X.....C.dl..
0040 f6 05 30 4a 02 61 08 66 49 04 23 75 69 02 20 06 ..OW..f.L.....
0050 69 6b 85 2c 6f 75 3d 75 73 65 72 73 2c 64 63 3d ..ke,ou=users,dc=
0060 65 78 61 6d 70 6c 65 2c 64 63 3d 63 6f 6a 30 22 ..example,dc=com"
0070 30 20 0a 01 02 30 1b 04 10 73 68 61 64 6f 77 4c 0...0...shadowL..
0080 61 73 74 43 68 61 6e 67 65 31 07 04 05 32 30 30 61 73 74 43 68 61 6e 67 65 31 07 04 05 32 30 30 ..astChange el...200
0090 32 35 25

We have identified the “**ldap**” thread that was responsible for changing the password in the wireshark.



We saved this thread on the virtual machine with the name “**password.pcapng**”

```
mike@server2:~$ exit
logout
Connection to server2 closed.
mike@server1:~$
```

Then we terminated the “**mike**” connection on **server 2**.

Then we reconnected to **server 1** with the command “**ssh mike@server1**”

4.3 Use the mike credentials to access another server

Exit your ssh session from server1. Then ssh to server2:

```
ssh mike@server2
```

What password do you expect to use to authenticate to server2? After logging into server2, exit that ssh session.

The expected password is the one established at the base “**adminpass**”.

```
mike@server1:~$ exit
logout
Connection to server1 closed.
mike@server1:~$ ssh mike@server2
The authenticity of host 'server2 (172.25.0.5)' can't be established.
ECDSA key fingerprint is SHA256:ZtE8xi5Y50aUktZ/XtgjIs1c5jxYQ884VqSofmlgGng.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server2,172.25.0.5' (ECDSA) to the list of known hosts.
mike@server2's password:
Creating directory '/home/mike'.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 6.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

mike@server2:~$
```

After which we broke the connection with server 1 and logged into server 2 to verify that we have access to **server 2**.

```
mike@server2:~$ exit
logout
Connection to server2 closed.
```

Then we terminated the connection on **server 2**.

4.4 Add an LDAP user

Go to the ldap virtual terminal and use `ls` to see a directory listing. View the file named `mike.ldif`, it was used to define the user named “mike”. Then view the `projx.ldif` file. The LDAP command that was used to add the entry defined in `mike.ldif` is:

```
ldapadd -x -W -D "cn=admin,dc=example,dc=com" -f mike.ldif
```

Note how the `-D` option names the administrator on whose behalf the LDAP addition is to be made. Use `man ldapadd` to learn more about the syntax of that command. The initial password for the mike user was created with this command:

```
ldappasswd -s password123 -W -D "cn=admin,dc=example,dc=com" \
-x "uid=mike,ou=users,dc=example,dc=com"
```

Create ldif files to define a new group named “qa” and a new user having an ID of “mary”. Assign mary to the qa group. Take care to adjust the `uidNumber` and `gidNumber` values. Use the `ldapadd` command to add the new group and the new user. Use the `ldappasswd` command to assign an initial password to mary. Again, the password for the LDAP administrator is “adminpass”.

Then go to the client computer and test your ability to ssh as mary to both server1 and server2.

```
admin@ldap:~$ ls
mike.ldif  password.pcapng  projx.ldif
admin@ldap:~$
```

We checked the files present in the **ldap folder**.


```
admin@ldap:~$ cat mike.ldif
dn: uid=mike,ou=users,dc=example,dc=com
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: mike
uid: mike
uidNumber: 1501
gidNumber: 1500
homeDirectory: /home/mike
loginShell: /bin/bash
gecos: mike
userPassword: {crypt}x
shadowLastChange: 0
shadowMax: 0
shadowWarning: 0
```

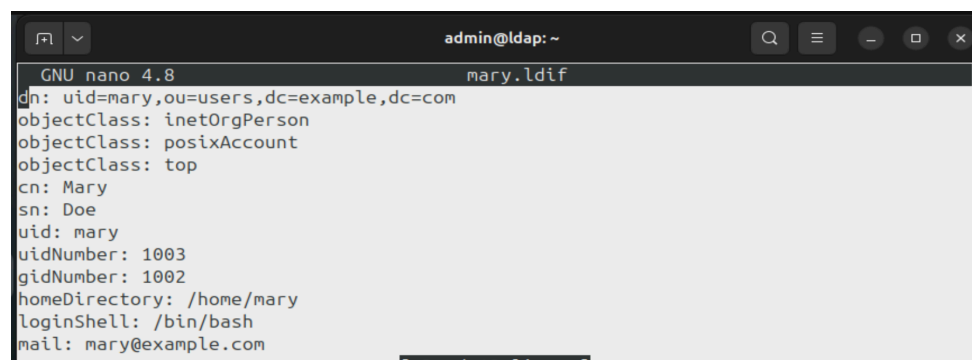
Above we have displayed the components of the “**mike.ldif**” file with the “**cat**” command.

```
admin@ldap:~$ cat projx.ldif
dn: cn=projx,ou=groups,dc=example,dc=com
objectClass: top
objectClass: posixGroup
gidNumber: 1500
admin@ldap:~$
```

Above, we have displayed the components of the “**projx.ldif**” file with the “**cat**” command.

```
admin@ldap:~$ ldapadd -x -W -D "cn=admin,dc=example,dc=com" -f mike.ldif
Enter LDAP Password:
adding new entry "uid=mike,ou=users,dc=example,dc=com"
ldap_add: Already exists (68)
```

In the “**admin@ldap**” window, using the command “**ldapadd -x -W -D**
“**cn=admin,dc=example,dc=com**” -f **mike.ldif**”, we defined mike as an administrator and specified the file to import containing his information.



```
admin@ldap: ~
GNU nano 4.8      mary.ldif
dn: uid=mary,ou=users,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: Mary
sn: Doe
uid: mary
uidNumber: 1003
gidNumber: 1002
homeDirectory: /home/mary
loginShell: /bin/bash
mail: mary@example.com
```

With “**nano**” we verified that the information was correct by adding it to the file.

```
admin@ldap: ~  
GNU nano 4.8 qa.ldif  
dn: cn=qa,ou=groups,dc=example,dc=com  
objectClass: posixGroup  
cn: qa  
gidNumber: 1002
```

Similarly, we checked and added the information in the group file “**qa.ldif**”.

```
admin@ldap:~$ nano qa.ldif  
admin@ldap:~$ nano mary.ldif  
admin@ldap:~$ ldapadd -x -W -D "cn=admin,dc=example,dc=com" -f qa.ldif  
Enter LDAP Password:  
adding new entry "cn=qa,ou=groups,dc=example,dc=com"  
  
admin@ldap:~$ ldapadd -x -W -D "cn=admin,dc=example,dc=com" -f mary.ldif  
Enter LDAP Password:  
adding new entry "uid=mary,ou=users,dc=example,dc=com"  
  
admin@ldap:~$ ldapasswd -s "mary123" -W -D "cn=admin,dc=example,dc=com" -x "uid=mary,ou=users,dc=example,dc=com"  
Enter LDAP Password:  
admin@ldap:~$
```

We created and defined a new admin “**mary**” and added her using the “**mary.ldif**” file containing her information.

We then changed the password of “**mary**” in the file to that of “**mary123**”

```
mike@server1:~$ ssh mary@server1  
mary@server1's password:  
Creating directory '/home/mary'.  
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 6.8.0-36-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
mary@server1:~$
```

We then managed to connect it to **server 1**

```
mary@server1:~$ exit  
logout  
Connection to server1 closed.
```

We have terminated his connection to this server.

```
mike@server1:~$ ssh mary@server2
mary@server2's password:
Creating directory '/home/mary'.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 6.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

mary@server2:~$
```

We then managed to connect it to **server 2**.

```
mary@server2:~$ exit
logout
Connection to server2 closed.
```

We have terminated his connection to this server.

5 Submission

After finishing the lab, go to the terminal on your Linux system that was used to start the lab and type:

```
stoplab
```

When you stop the lab, the system will display a path to the zipped lab results on your Linux system. Provide that file to your instructor, e.g., via the Sakai site.

We finally stopped the labtainer with the “**stoplab**” command.

```
student@Labtainer-VirtualBox:~/labtainer/labtainer-student$ stoplab
Results stored in directory: /home/student/labtainer_xfer/ldap
student@Labtainer-VirtualBox:~/labtainer/labtainer-student$
```