Jesua Gonzalez

Ethical Hacking

Lightweight Directory Access Protocol

Sat Oct 11

Executive Summary

This report covers my hands-on work in the Labtainers LDAP lab, where I explored how the Lightweight Directory Access Protocol(LDAP) centralizes user and group authentication for Linux systems. In a controlled lab environment, I will connect a client to the shared LDAP repository, enumerate and inspect directory entries, create or modify test user records as allowed, and verify that multiple hosts can authenticate against the same credential set. All while taking the necessary screenshots and keeping vital output results to demonstrate my findings and provide context to my learning experience. My goals are to showcase the flow of authentication, observe and analyze encrypted versus unencrypted LDAP traffic, and produce clear, reproducible steps and remediation advice based on my findings.

Throughout the lab, I used Wireshark to capture and inspect network traffic, allowing me to observe LDAP authentication exchanges and determine whether credentials were transmitted securely. I used SSH to perform encrypted logins from the client and to demonstrate what a secure session looks like when authentication is backed by the directory. I also modified LDAP entries to see how changes to user and group records affected authentication. While performing these actions, I saved packet captures, console output, and screenshots so each step can be documented and audited.

Lab LDAP:

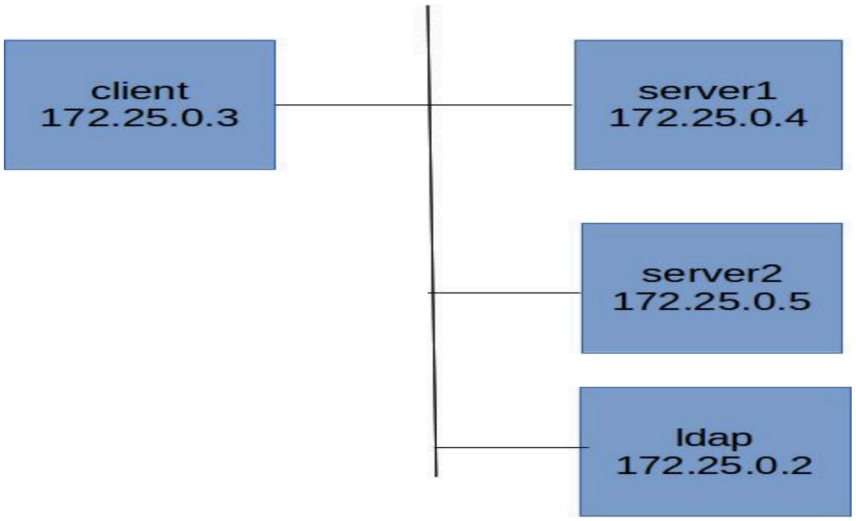
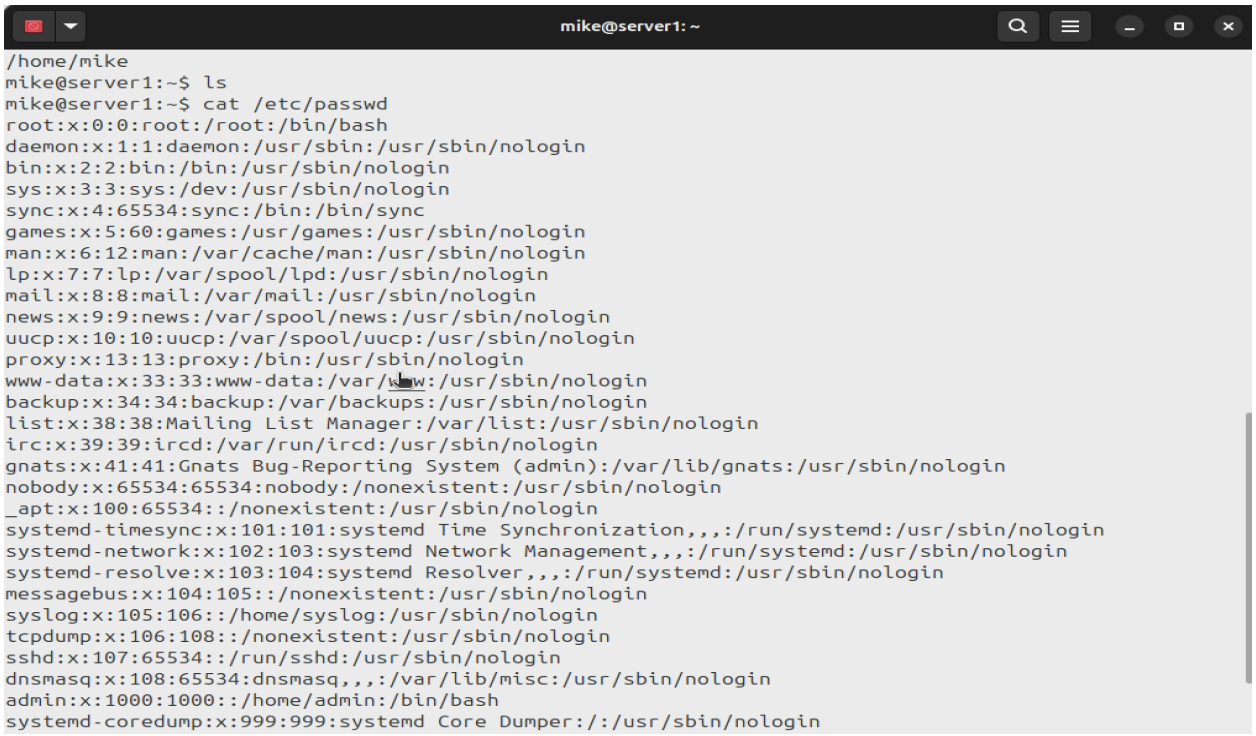


Figure 1: Network topology for the LDAP lab

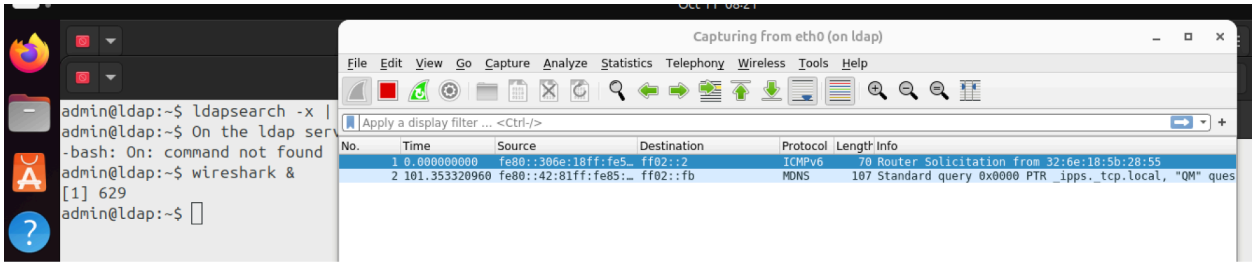
- The lab provides a client, two servers, and an LDAP server, as shown in the following topology. I used the supplied client and LDAP terminal windows to run commands to authenticate Mike and other users to the LDAP-connected servers.

- Starting this lab, I first checked the current directory contents using the command **'ldapsearch -x | less'**. This allowed me to observe the already configured users/groups on the LDAP server, in this case, 'Mike' and 'Projx'.

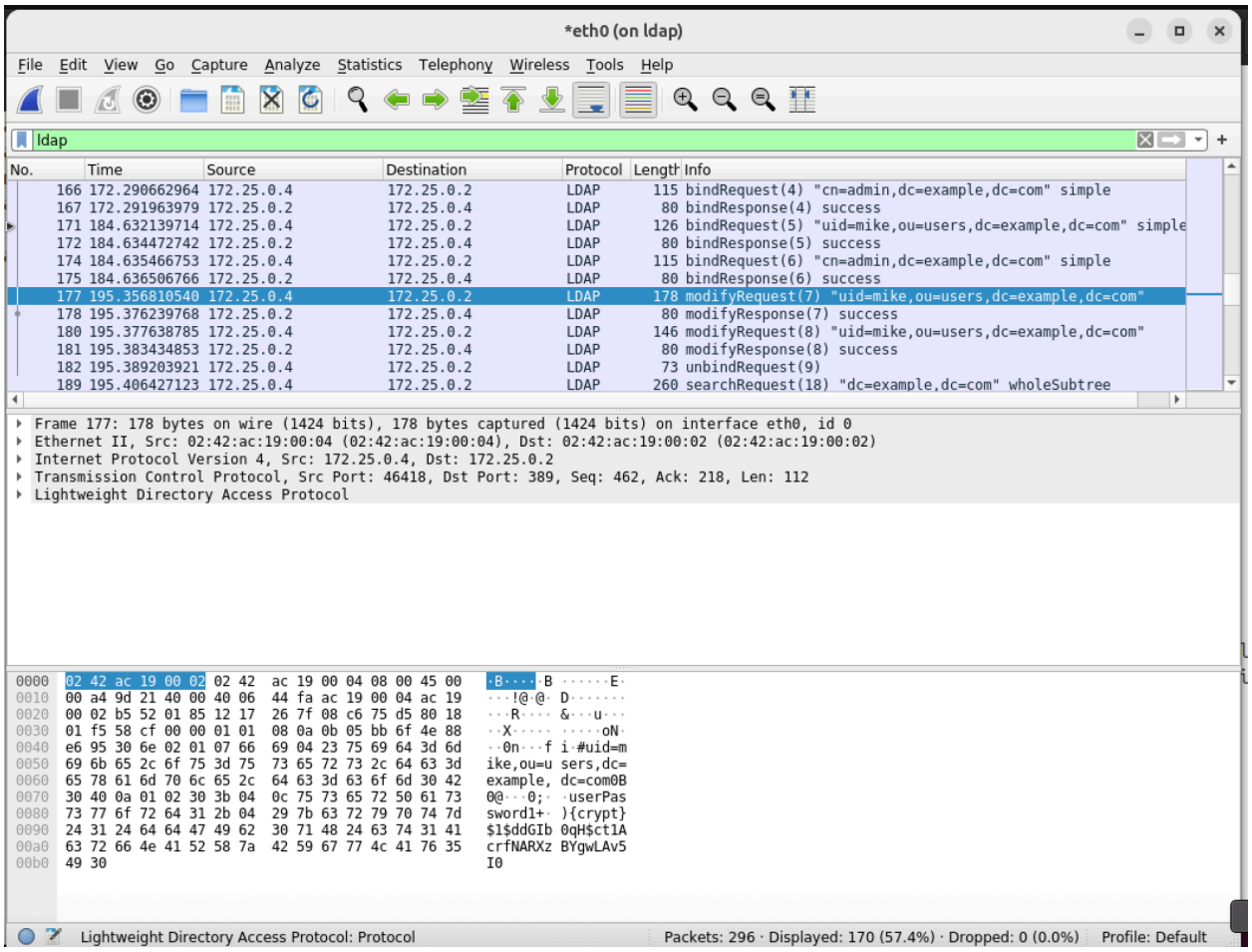
- Using the SSH server login as Mike, I was prompted to change my password: ‘password123’. Usually, this occurs in the real world due to rules such as a password policy, which keeps things in rotation and up to date, to minimize attack surfaces.



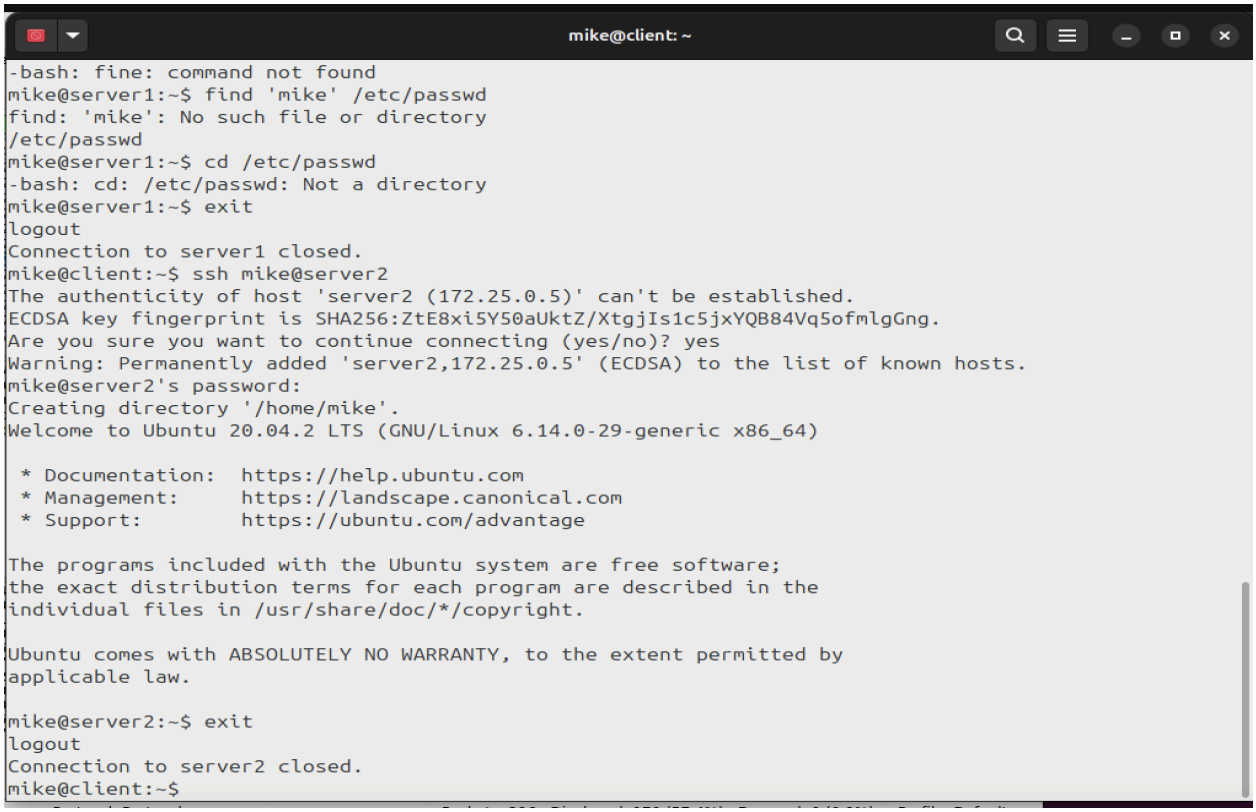
- After successfully logging in as Mike with a new password, I was then able to use the ‘ls’ command. This allowed me to display files, and I was able to read files such as ‘/etc/passwd’. I was not able to locate any files with the names tagged under user ‘Mike' or group ‘Projx’, so I side-noted this and needed to further research file topics.



- On the other hand, while I tried logging into Mike through the SSH server login, I had used **'wireshark &'** to listen to protocol traffic flow on the eth0 device.



- After logging in to server 1 through the client window, I stopped my packet sniffing and used the packet display filter for LDAP results only. This displayed the connections using LDAP on port 389. I was then able to track communication between the LDAP server and server1 through IP addresses 172.25.0.4 ← → 172.25.0.2. Packet 177 shows a request to modify. In this case, it was the time when I was prompted to change my default password, and I was able to save this packet in a file called **'password.pcapng'** for future reference.

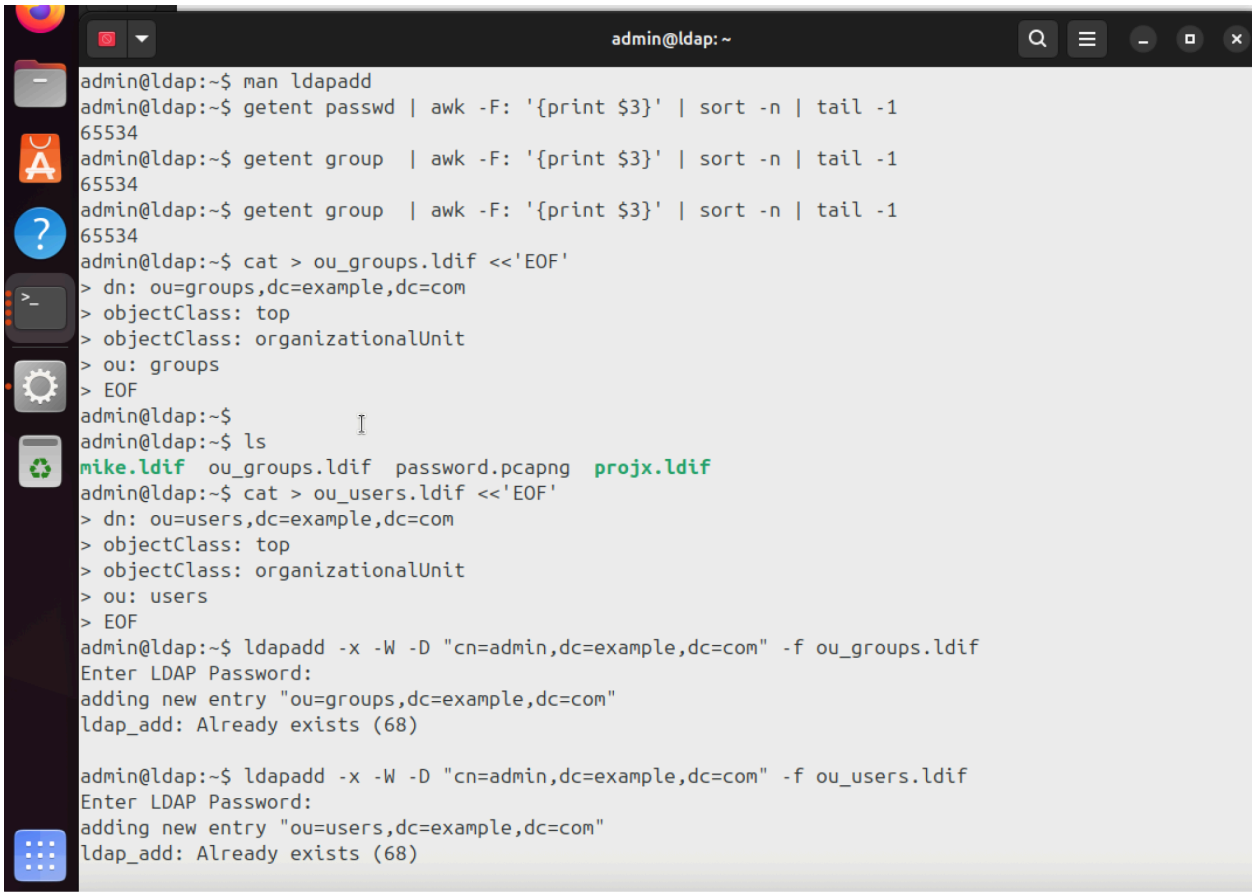


- After exiting the first session with server1 using the **'Exit'** command, I then switched to logging into server2. What I noticed was that I was able to enter server2 using the new credentials I changed after I first tried server1. This leads me to confirm that LDAP stores credentials to be able to match at later logins to authenticate users in a network.

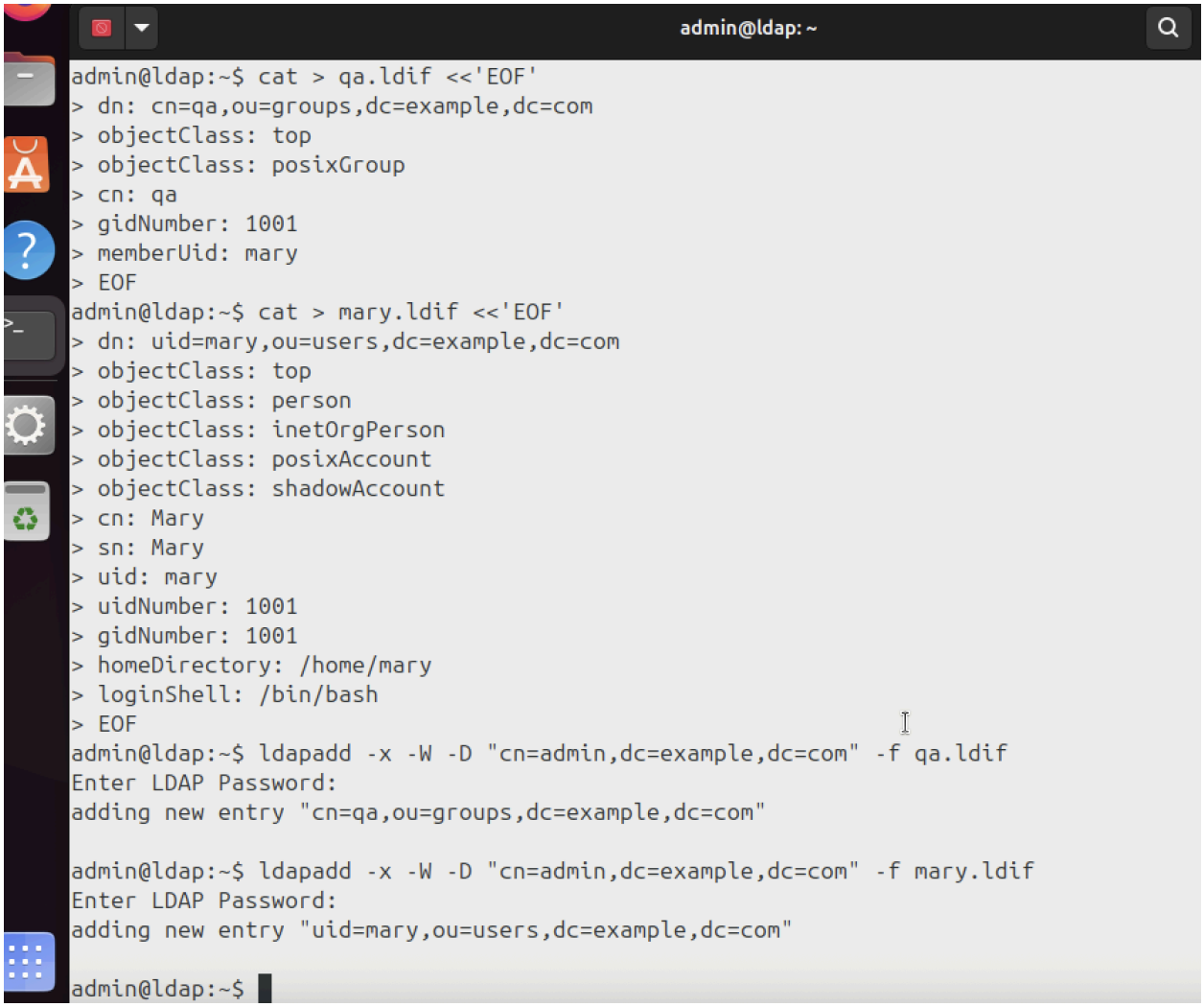
```
admin@ldap:~$ ls
mike.ldif  password.pcapng  projx.ldif
admin@ldap:~$ cat mike.ldif
dn: uid=mike,ou=users,dc=example,dc=com
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: mike
uid: mike
uidNumber: 1501
gidNumber: 1500
homeDirectory: /home/mike
loginShell: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

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

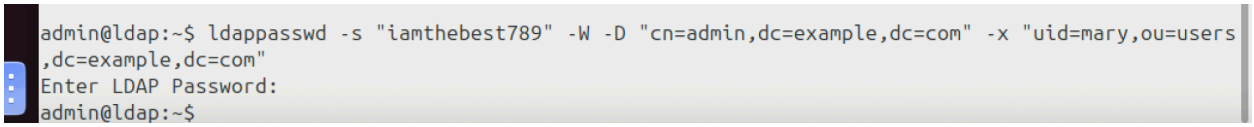
- Back to the LDAP terminal, I viewed the files of the group and users using a simple **'cat'** command to get a better understanding of what components went into defining these accounts.



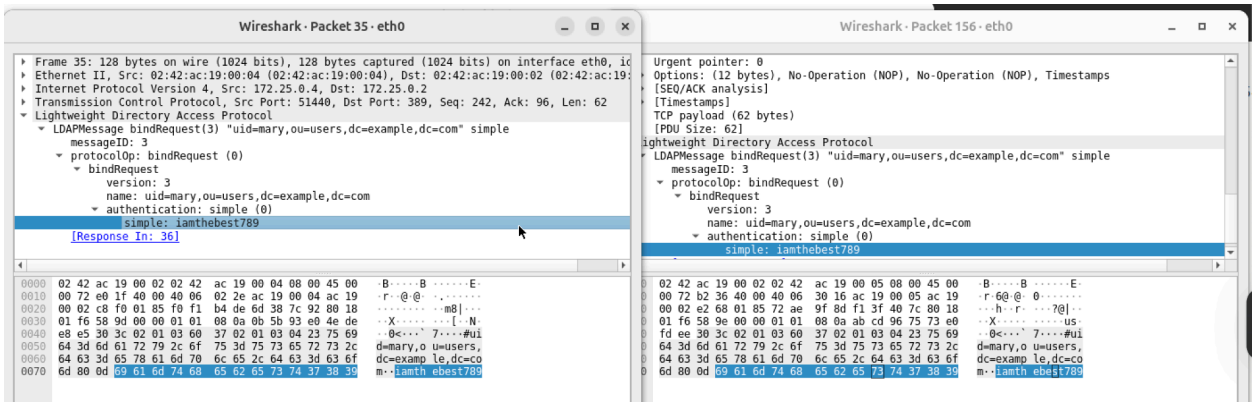
- I now needed to create two new profiles, one for a user named 'Mary' and one for a group named 'qa'. But first, parent files needed to be created for LDAP to add any new users or groups to a coherent state. I was able to figure out more information using the AWS Directory Service: Create an LDIF guide.



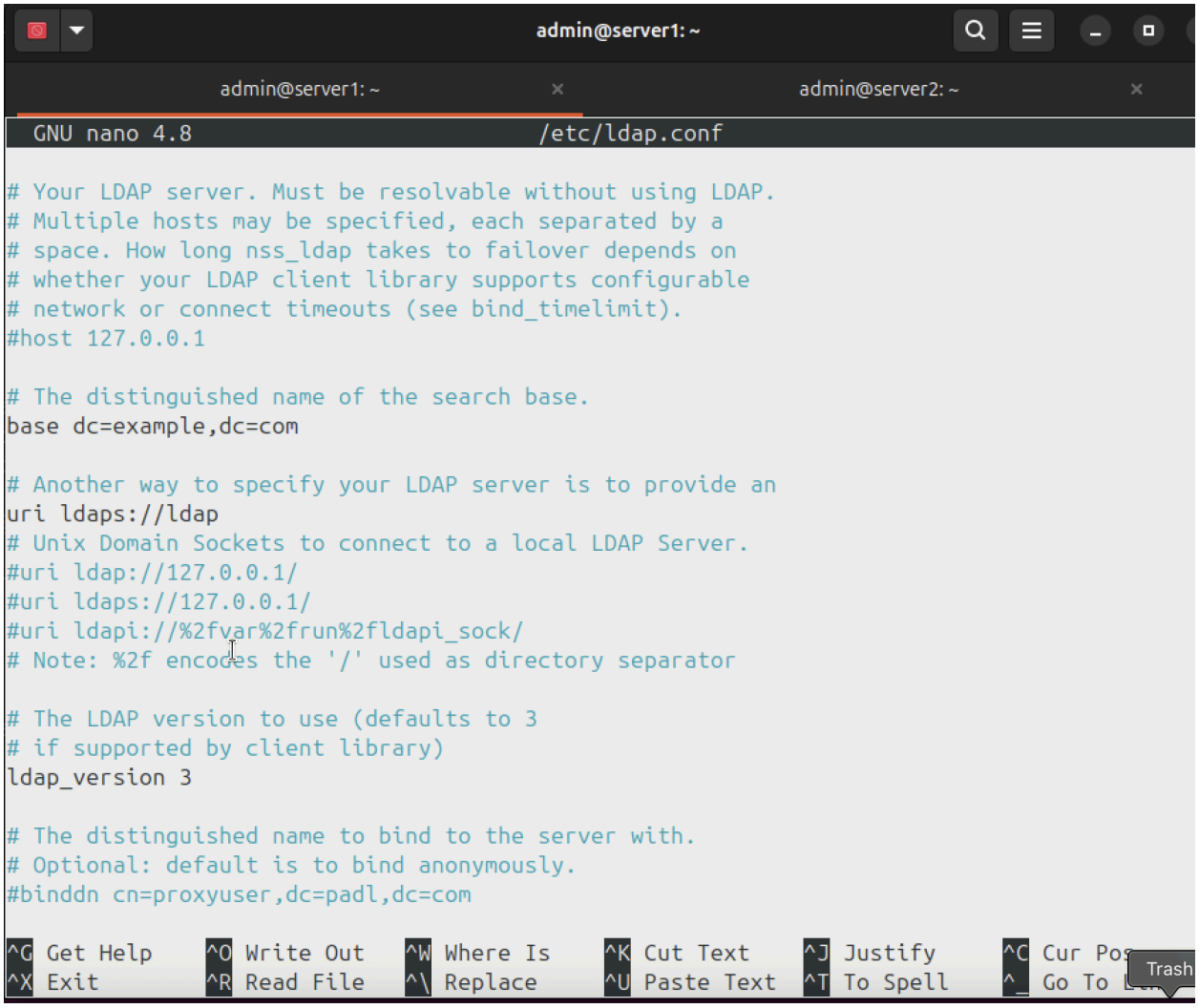
- Once the parent files have been created, we can move forward with the actual users we want to add. I have created LDIF files to input static information and used the command **'ldapadd -x -W -D "cn=admin,dc=example,dc=com" -f mike.ldif',** which was used to create Mike's user profile.

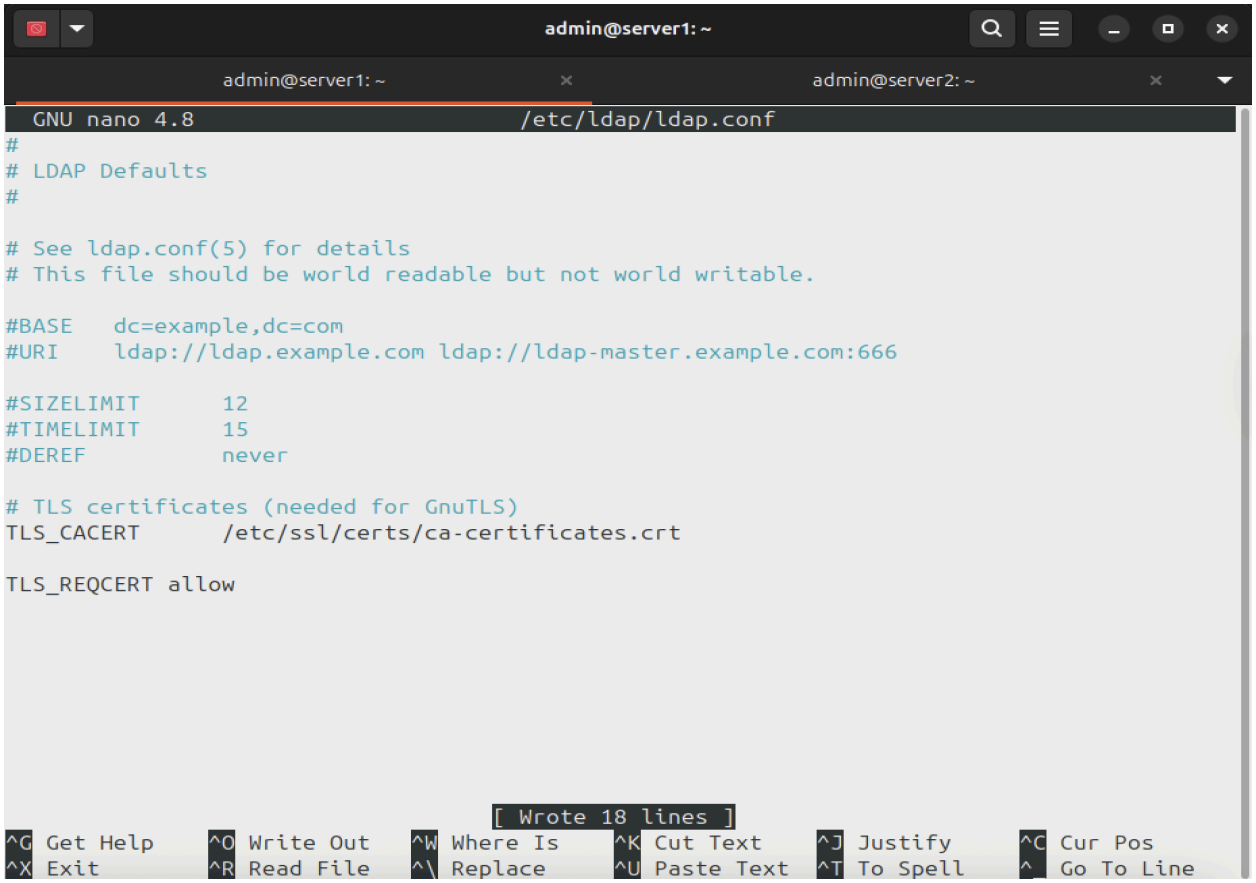


- The final touch was to give Mary's user account a password so they can log in to servers 1 and 2. This was done with the command **'ldappasswd -s password123 -W -D "cn=admin,dc=example,dc=com" \ -x "uid=mike,ou=users,dc=example,dc=com" '.**

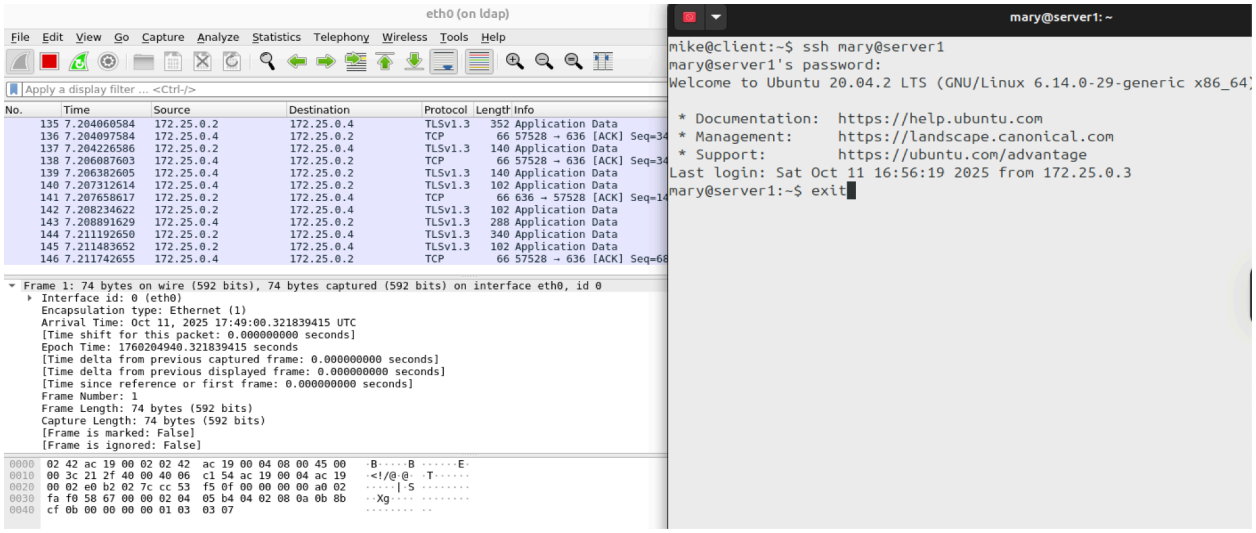


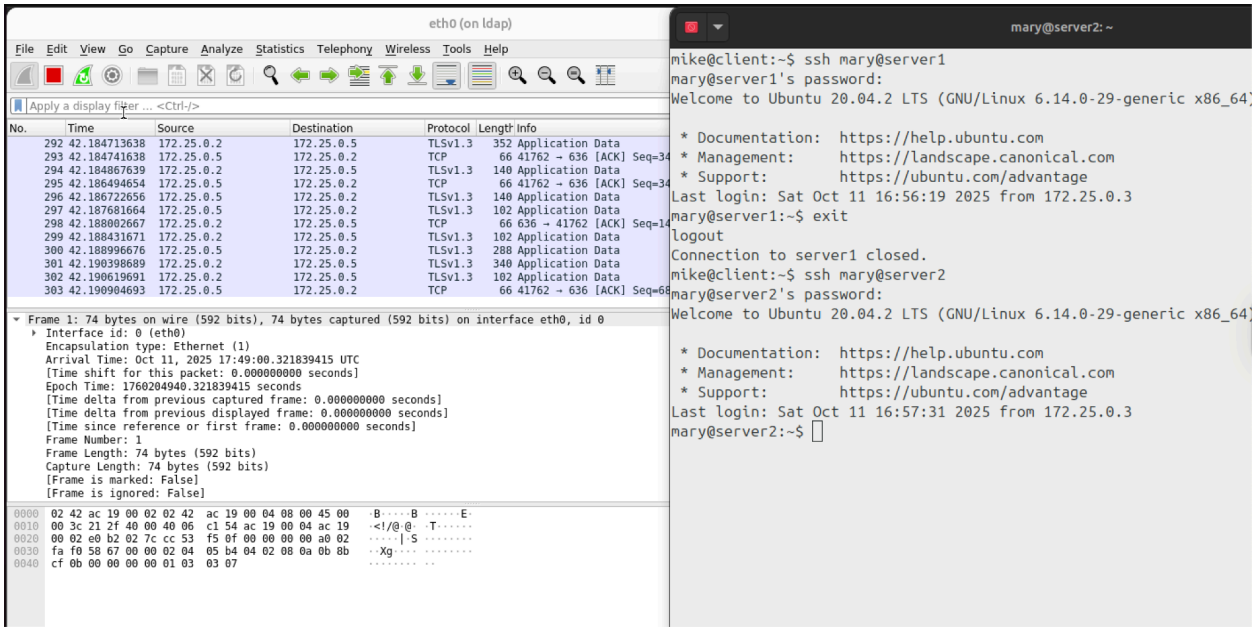
- I went ahead and tried Mary's new account by logging into both servers, which was a success. Simultaneously, I ran Wireshark to capture packets and analyze communication, and to my surprise, Mary's credentials were being shared over the network via plaintext. This poses a huge risk to unauthorized access if a real threat actor were to intercept these credentials through this LDAP vulnerability.



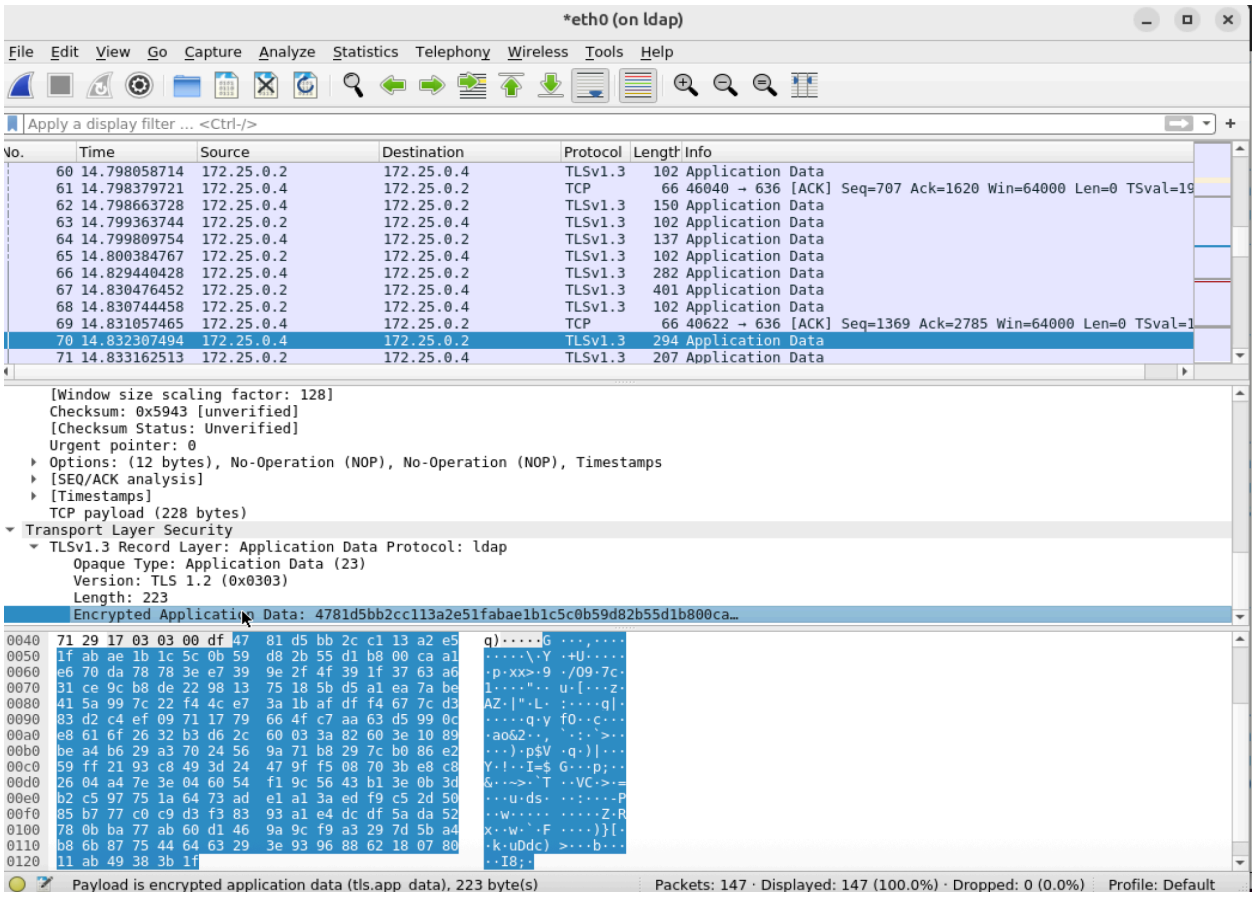


- It was time to upgrade LDAP communication to our servers, so on the server terminals, I went into the LDAP configuration files using a simple text editor, **nano**, to add SSL encryption to LDAP. I changed the uri to display ldaps instead of ldap and removed the # in front of the line that had 'SSL on'. Lastly, I was able to write out **'TLS_REQCERT allow'** on the /etc/ldap/ldap.conf file path to finish the secure LDAP configuration for both servers.





- Now it was time to test the newly configured LDAP with SSL by logging in to the servers with a user account, and with a new Wireshark session, I was able to keep track of the time of packets being sent out and in.



- Taking a closer look at the packets, we can see a few changes, starting with the protocol being displayed with the name ‘TLSv1.3’ instead of the old LDAP protocol. Also, when inspecting a single packet, there is new information saying Encrypted Application Data, in which packet information is no longer sent in plaintext, and we are only given a bunch of random characters showcasing the encryption over this network connection.

Unencrypted vs. encrypted LDAP traffic

In the Labtainers LDAP lab, I saw why encrypting LDAP matters. Unencrypted LDAP on port 389 sends binds, searches, and modify operations in cleartext, so anyone sniffing the network with tools like Wireshark can read domain names, attributes, and even credential data. Encrypted LDAP using SSL(LDAPS) protects those same operations with TLS; therefore, credentials and directory contents are not easily intercepted or replayed. Good security steps are simple and effective. Future businesses should require LDAPS for all clients and servers, make clients validate the server certificate, and install a trusted CA. Apart from encryption, there should be strict ACLs so non-admins cannot read sensitive information, as well as enforce strong password policies, and limit which hosts can reach the LDAP port with firewalls or VLANs. Enabling logs and monitoring of LDAP activity, rotating certificates regularly, and testing changes in a controlled environment before rolling them out can provide multiple layers of security control. Documenting my experience during the Labtainers LDAP exercise will benefit future assessments of similar situations and establish a foundation for Lightweight Directory Access Protocol security.