

# **Lab environment setup for cybersecurity students**

**Author: Yagub Hajiyeu (Cybersecurity trainee)**  
**Contact: [yagub.hajiyeu.995@gmail.com](mailto:yagub.hajiyeu.995@gmail.com)**

# NOTES

The RHEL 9 (Red Hat Enterprise Linux 9) OS and the Vmware virtual machine has been used as the platform for lab environment, and every process has been explained step by step and supported with screenshots. It is strongly recommended to use the same OS which is used in this lab if you suffer lack of Linux knowledge.

You can setup this lab in any computer and organize a competition between your friends or students. All you need to be in the same network, then attackers (competitors) will be able to communicate with the target host (lab installed computer). It is verified with practice that when students participate as a team they solve the puzzle faster and gain much more experience than single candidates.

The whole purpose of this puzzle is to find a file which is hidden in random directory and read the code in it. You have to provide candidates with only target host's IP address (IPv4) and with the name of the file which you hide.

All steps are related with each other so if you don't setup any of services then your lab will be failed. You can modify the lab however you want, it is up to your creativity. If you face with any trouble during the setup you can contact with me.

The clear solution of the puzzle is not provided in this documentation, because the setup process itself also explains everything you need.

## **Required knowledge for the lab:**

- Active Reconnaissance
- Enumeration skills
- Encoding and hashing
- Linux skills
- NFS mounting
- Tools: Nmap, Hydra, Smbclient, FTP, SSH, MYSQL, John the Ripper

I have created a script to setup whole lab environment automatically, the link is below. Please don't forget to read the "README.txt" before you execute the script.

Link for script: <https://github.com/Yagub-Hajiyev/ETHICAL.git>

And I share my Notion notes with you, they will be useful while you are doing lab.

Link for Notion: <https://waiting-panama-ec3.notion.site/Active-Recon-Enumeration-4874aff5368742fb82c05a8558c00d62?pvs=4>

# SSH

In RHEL 9, the SSH service comes ready to use with default configuration but we need to disable the password authentication. Firstly make sure that the service is running properly like the figure below.

```
[yagub@hajiyevev ~]$ sudo systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-02-09 01:24:16 EST; 11min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 1079 (sshd)
      Tasks: 1 (limit: 10681)
     Memory: 2.1M
        CPU: 25ms
    CGroup: /system.slice/sshd.service
            └─1079 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 09 01:24:16 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Feb 09 01:24:16 localhost.localdomain sshd[1079]: Server listening on 0.0.0.0 port 22.
Feb 09 01:24:16 localhost.localdomain sshd[1079]: Server listening on :: port 22.
Feb 09 01:24:16 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[yagub@hajiyevev ~]$
```

If everything is normal then open the configuration file and change the 'PasswordAuthentication's state from 'yes' to 'no', and save the modification.

```
[root@hajiyevev ~]# vim /etc/ssh/sshd_config
```

# To disable tunneled clear text	# To disable tunneled clear text
PasswordAuthentication yes	PasswordAuthentication no
#PermitEmptyPasswords no	#PermitEmptyPasswords no

Later restart the 'sshd.service' and check the status of the service.

```
[root@hajiyevev ~]# systemctl restart sshd.service
[root@hajiyevev ~]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-02-10 06:07:06 EST; 54s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
    Main PID: 4010 (sshd)
      Tasks: 1 (limit: 10681)
     Memory: 1.4M
        CPU: 54ms
    CGroup: /system.slice/sshd.service
            └─4010 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 10 06:07:06 hajiyevev systemd[1]: Starting OpenSSH server daemon...
Feb 10 06:07:06 hajiyevev sshd[4010]: Server listening on 0.0.0.0 port 22.
Feb 10 06:07:06 hajiyevev sshd[4010]: Server listening on :: port 22.
Feb 10 06:07:06 hajiyevev systemd[1]: Started OpenSSH server daemon.
```

Candidates will be obliged to use SSH service (by creating key pair and adding to users' home folder (~/.ssh/authorized\_keys) in order to have an access to the target machine.

# FTP

Apart from the SSH, the FTP service doesn't come as default in RHEL 9, so we need to install the FTP ourselves and following services. In this lab I used the 'vsftpd' but you can use other type if you want.

The command is 'dnf install vsftpd' for the installation process.

```
[yagub@hajiyevev ~]$ sudo dnf install vsftpd
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:30:22 ago on Fri 09 Feb 2024 01:31:08 AM EST.
Dependencies resolved.
=====
Package                                Architecture    Version          Repository      Size
=====
Installing:
vsftpd                                x86_64          3.0.5-5.el9      a               172 k
=====
Transaction Summary
=====
Install 1 Package

Total size: 172 k
Installed size: 347 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : 
  Installing     : vsftpd-3.0.5-5.el9.x86_64                1/1
  Running scriptlet: vsftpd-3.0.5-5.el9.x86_64                1/1
  Verifying      : vsftpd-3.0.5-5.el9.x86_64                1/1
Installed products updated.

Installed:
vsftpd-3.0.5-5.el9.x86_64

Complete!
[yagub@hajiyevev ~]$
```

After the installation we enable and start the service with the 'systemctl enable --now vsftpd.service' command and check the status of the service.

```
[yagub@hajiyevev ~]$ sudo systemctl enable --now vsftpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[yagub@hajiyevev ~]$ sudo systemctl status vsftpd.service
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-02-09 02:02:29 EST; 10s ago
     Process: 6617 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
    Main PID: 6618 (vsftpd)
       Tasks: 1 (limit: 10681)
      Memory: 712.0K
         CPU: 5ms
    CGroup: /system.slice/vsftpd.service
           └─6618 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Feb 09 02:02:29 hajiyevev systemd[1]: Starting Vsftpd ftp daemon...
Feb 09 02:02:29 hajiyevev systemd[1]: Started Vsftpd ftp daemon.
[yagub@hajiyevev ~]$
```

Another important step is to add the installed service to the firewall list. If we don't add the service than it will not seen in the port scan. You can follow the steps below in the figure to add FTP service to the firewall. This process will be applied to all services in following sections.

```
[root@hajiyevev ~]# firewall-cmd --add-service=ftp --permanent
success
[root@hajiyevev ~]# firewall-cmd --reload
success
[root@hajiyevev ~]# firewall-cmd --list-services
cockpit dhcpv6-client ftp mysql samba ssh
[root@hajiyevev ~]#
```

Now we have to let the anonymous login in the configuration of the FTP. Open the configuration file and change the 'anonymous\_enable's state from 'NO' to 'YES'.

```
[yagub@hajiyevev ~]$ sudo vim /etc/vsftpd/vsftpd.conf
```

```
# Allow anonymous FTP?
anonymous_enable=NO
```

```
# Allow anonymous FTP?
anonymous_enable=YES
```

You can also add any information to the service's banner as I did below.

```
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
```

```
# You may fully customise the login banner string:
ftpd_banner=Welcome to Yagub's lab project.
```

After modifying the configuration file we need to restart the service.

```
[yagub@hajiyevev ~]$ sudo systemctl restart vsftpd.service
```

In default, shared files locate under the '/var/ftp' path. Go to this directory and create a user list as in the figure. Later this list will be used to make brute force attack for gaining access to the MYSQL service.

```
[yagub@hajiyevev ~]$ cd /var/ftp/
[yagub@hajiyevev ftp]$ ll
total 0
drwxr-xr-x. 2 root root 6 May 9 2023 pub
[yagub@hajiyevev ftp]$ sudo touch users.txt
[yagub@hajiyevev ftp]$ sudo vim users.txt
[yagub@hajiyevev ftp]$ ll
total 4
drwxr-xr-x. 2 root root 6 May 9 2023 pub
-rw-r--r--. 1 root root 134 Feb 9 02:22 users.txt
[yagub@hajiyevev ftp]$
```

```
Nariman
Natig
Gasim
Vusal
Farrukh
Vidadi
Yagub
Mamed
Zeynal
Kamran
Majid
Murad
Mahir
Arif
Nazrin
Orkhan
Arzu
Nihat
Rasul
Elgun
Nazar
```

# Samba (SMB)

Samba is a free software which is used for re-implementation of SMB protocol to Unix like operation systems such as Linux. In this lab we will share some files over the Samba to provide candidates with information for completing the puzzle.

Firstly we have to install it to our Linux with the help of the 'dnf install -y samba samba-client samba-common' command.

```
[yagub@hajiyeve ~]$ sudo dnf install -y samba samba-client samba-common
[sudo] password for yagub:
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 0:00:13 ago on Fri 09 Feb 2024 02:30:41 AM EST.
Package samba-common-4.18.6-100.el9.noarch is already installed.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository         Size
=====
Installing:
samba                                  x86_64            4.18.6-100.el9    b                  984 k
samba-client                          x86_64            4.18.6-100.el9    a                  678 k
Installing dependencies:
libnetapi                             x86_64            4.18.6-100.el9    b                  143 k
samba-common-tools                    x86_64            4.18.6-100.el9    b                  461 k
samba-dcerpc                          x86_64            4.18.6-100.el9    b                  694 k
samba-ldb-ldap-modules                x86_64            4.18.6-100.el9    b                   30 k
samba-libs                            x86_64            4.18.6-100.el9    b                  128 k
=====
Transaction Summary
=====
Install 7 Packages

Total size: 3.0 M
Installed size: 10 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                                     : 1/1
  Installing : samba-libs-4.18.6-100.el9.x86_64 : 1/7
  Installing : libnetapi-4.18.6-100.el9.x86_64  : 2/7
  Installing : samba-dcerpc-4.18.6-100.el9.x86_64 : 3/7
  Installing : samba-ldb-ldap-modules-4.18.6-100.el9.x86_64 : 4/7
  Installing : samba-common-tools-4.18.6-100.el9.x86_64 : 5/7
  Installing : samba-4.18.6-100.el9.x86_64       : 6/7
  Running scriptlet: samba-4.18.6-100.el9.x86_64 : 6/7
  Installing : samba-client-4.18.6-100.el9.x86_64 : 7/7
  Running scriptlet: samba-client-4.18.6-100.el9.x86_64 : 7/7
  Verifying   : libnetapi-4.18.6-100.el9.x86_64  : 1/7
  Verifying   : samba-4.18.6-100.el9.x86_64       : 2/7
  Verifying   : samba-common-tools-4.18.6-100.el9.x86_64 : 3/7
  Verifying   : samba-dcerpc-4.18.6-100.el9.x86_64 : 4/7
  Verifying   : samba-ldb-ldap-modules-4.18.6-100.el9.x86_64 : 5/7
  Verifying   : samba-libs-4.18.6-100.el9.x86_64   : 6/7
  Verifying   : samba-client-4.18.6-100.el9.x86_64 : 7/7
Installed products updated.

Installed:
  libnetapi-4.18.6-100.el9.x86_64      samba-4.18.6-100.el9.x86_64      samba-client-4.18.6-100.el9.x86_64
```

Later we enable and start (with '--now' option) the 'smb.service'.

```
[yagub@hajiyeve ~]$ sudo systemctl enable --now smb.service
Created symlink /etc/systemd/system/multi-user.target.wants/smb.service → /usr/lib/systemd/system/smb.service.
[yagub@hajiyeve ~]$ sudo systemctl status smb.service
● smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-02-09 02:32:38 EST; 10s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 7749 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 3 (limit: 10681)
  Memory: 20.1M
    CPU: 80ms
   CGroup: /system.slice/smb.service
           └─7749 /usr/sbin/smbd --foreground --no-process-group
             └─7751 /usr/sbin/smbd --foreground --no-process-group
               └─7752 /usr/sbin/smbd --foreground --no-process-group

Feb 09 02:32:38 hajiyeve systemd[1]: Starting Samba SMB Daemon...
Feb 09 02:32:38 hajiyeve smbd[7749]: [2024/02/09 02:32:38.108799, 0] ../../source3/smbd/server.c:1746(main)
Feb 09 02:32:38 hajiyeve smbd[7749]:   smbd version 4.18.6 started.
Feb 09 02:32:38 hajiyeve smbd[7749]:   Copyright Andrew Tridgell and the Samba Team 1992-2023
Feb 09 02:32:38 hajiyeve systemd[1]: Started Samba SMB Daemon.
[yagub@hajiyeve ~]$
```

The next step is to add the service to our firewall like below in the figure.

```
[yagub@hajiyevev ~]$ sudo firewall-cmd --list-services
cockpit dhcpv6-client ssh
[yagub@hajiyevev ~]$ sudo firewall-cmd --add-service=samba --permanent
success
[yagub@hajiyevev ~]$ sudo firewall-cmd --list-services
cockpit dhcpv6-client ssh
[yagub@hajiyevev ~]$ sudo firewall-cmd --reload
success
[yagub@hajiyevev ~]$ sudo firewall-cmd --list-services
cockpit dhcpv6-client samba ssh
[yagub@hajiyevev ~]$
```

In this step we have to create a path to the shared folder. As you seen below, our shared folder is 'CS301'. Now we give fully permission to everyone with the 'chmod' command and change the folder's user owner and group owner to 'nobody'. The 'nobody' is a reserved user for such services where it doesn't require a special permission and open to everyone. The purpose of using this user it to minimize the damage for the system when it is hacked.

```
[yagub@hajiyevev ~]$ sudo mkdir -p /CodeAcademy/CS301
[yagub@hajiyevev ~]$ sudo chmod -R 0777 /CodeAcademy/CS301
[yagub@hajiyevev ~]$ sudo chown -R nobody:nobody /CodeAcademy/CS301
```

Below we change the context label for the 'CS301' folder in order to get the permission from the SELinux. We can see that it comes with 'default\_t' label but we change it to the 'samba\_share\_t' label.

```
[yagub@hajiyevev ~]$ ll -Zd /CodeAcademy/CS301
drwxrwxrwx. 2 nobody nobody unconfined_u:object_r:default_t:s0 6 Feb  9 02:41 /CodeAcademy/CS301
[yagub@hajiyevev ~]$ sudo semanage fcontext -a -t samba_share_t "/CodeAcademy(/.*)?"
[yagub@hajiyevev ~]$ sudo restorecon -R /CodeAcademy
[yagub@hajiyevev ~]$ ll -Zd /CodeAcademy/CS301
drwxrwxrwx. 2 nobody nobody unconfined_u:object_r:samba_share_t:s0 6 Feb  9 02:41 /CodeAcademy/CS301
[yagub@hajiyevev ~]$
```

The next step is to modify the Samba's configuration file, all we need to change the 'workgroup' value (in [global] section), add the 'netbios name = rhel' line (in [global] section) and add entire [anonymous] section as in the figure (right one) which doesn't come as default. This [anonymous] section allow users to enter the shared folder anonymously and they will have some permissions whatever we defined in configuration file (in [anonymous] section).

```
[yagub@hajiyevev ~]$ sudo vim /etc/samba/smb.conf
```



```

# Note:
# SMB1 is disabled by default. This me
# SMB3 are no longer able to connect t

[global]
    workgroup = SAMBA
    security = user

    passdb backend = tdbsam

    printing = cups
    printcap name = cups
    load printers = yes
    cups options = raw

[homes]
    comment = Home Directories
    valid users = %S, %D%w%S
    browseable = No
    read only = No
    inherit acls = Yes

[printers]
    comment = All Printers
    path = /var/tmp
    printable = Yes
    create mask = 0600
    browseable = No

[print$]
    comment = Printer Drivers
    path = /var/lib/samba/drivers
    write list = @printadmin root
    force group = @printadmin
    create mask = 0664
    directory mask = 0775

[global]
    workgroup = WORKGROUP
    security = user
    netbios name = rhel
    passdb backend = tdbsam

    printing = cups
    printcap name = cups
    load printers = yes
    cups options = raw

[anonymous]
    comment = Anonymous File Server Share
    path = /CodeAcademy/CS301
    browsable = yes
    writable = yes
    guest ok = yes
    read only = no
    force user = nobody

[homes]
    comment = Home Directories
    valid users = %S, %D%w%S
    browseable = No
    read only = No
    inherit acls = Yes

[printers]
    comment = All Printers
    path = /var/tmp
    printable = Yes
    create mask = 0600
    browseable = No

[print$]
    comment = Printer Drivers
    path = /var/lib/samba/drivers

```

After modifying the configuration file we test it with the Samba's default tool 'testparm'. If there is any misconfiguration in the service's configuration file then this tool will warn us.

```

[yagub@hajiyev ~]$ sudo testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)

Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions

# Global parameters
[global]
    netbios name = RHEL
    printcap name = cups
    security = USER
    idmap config * : backend = tdb
    cups options = raw

[anonymous]
    comment = Anonymous File Server Share
    force user = nobody
    guest ok = Yes
    path = /CodeAcademy/CS301
    read only = No

[homes]
    browseable = No
    comment = Home Directories

```



Like every configuration change, also in the Samba it requires to restart the service for applying new configuration.

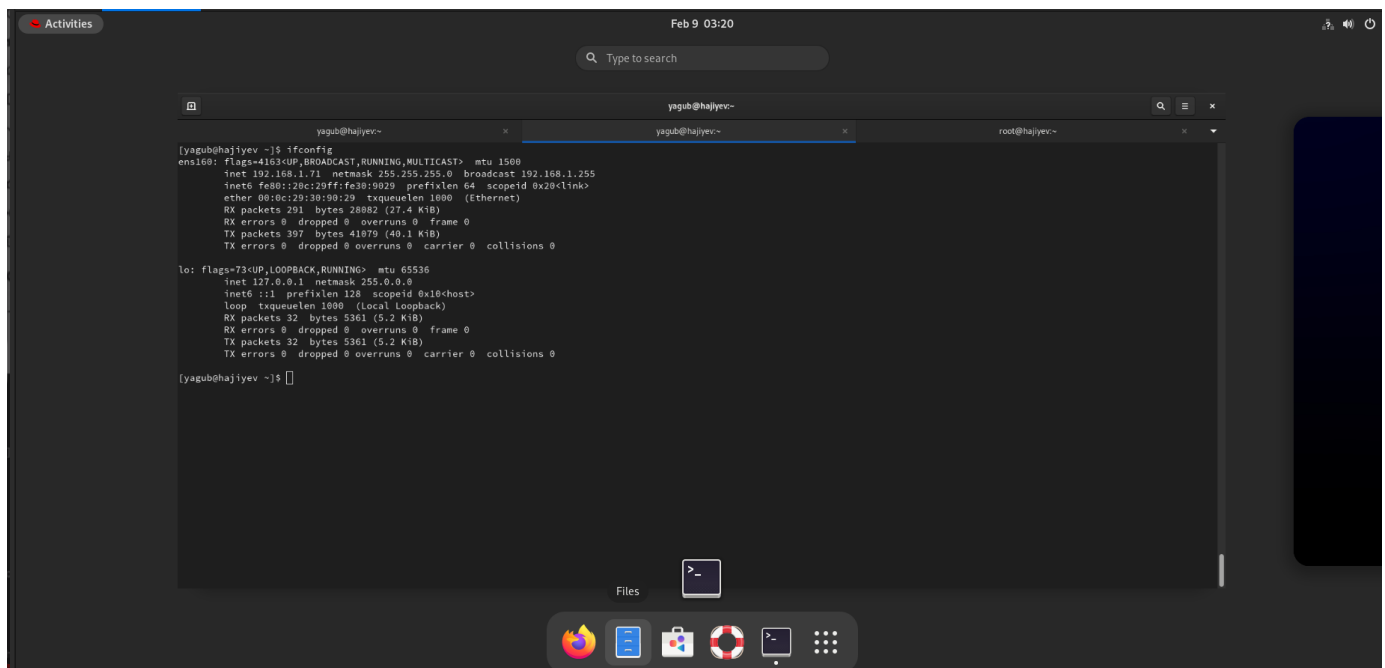
```
[yagub@hajiyeve ~]$ sudo systemctl restart smb.service
[yagub@hajiyeve ~]$ sudo systemctl status smb.service
● smb.service - Samba SMB Daemon
   Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-02-09 03:17:03 EST; 2s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 8194 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 3 (limit: 10681)
    Memory: 7.6M
       CPU: 49ms
    CGroup: /system.slice/smb.service
            └─8194 /usr/sbin/smbd --foreground --no-process-group
              └─8197 /usr/sbin/smbd --foreground --no-process-group
                └─8198 /usr/sbin/smbd --foreground --no-process-group

Feb 09 03:17:03 hajiyeve systemd[1]: Starting Samba SMB Daemon...
Feb 09 03:17:03 hajiyeve smbd[8194]: [2024/02/09 03:17:03.802444, 0] ../../source3/smbd/server.c:1746(main)
Feb 09 03:17:03 hajiyeve smbd[8194]: smbd version 4.18.6 started.
Feb 09 03:17:03 hajiyeve smbd[8194]: Copyright Andrew Tridgell and the Samba Team 1992-2023
Feb 09 03:17:03 hajiyeve systemd[1]: Started Samba SMB Daemon.
[yagub@hajiyeve ~]$
```

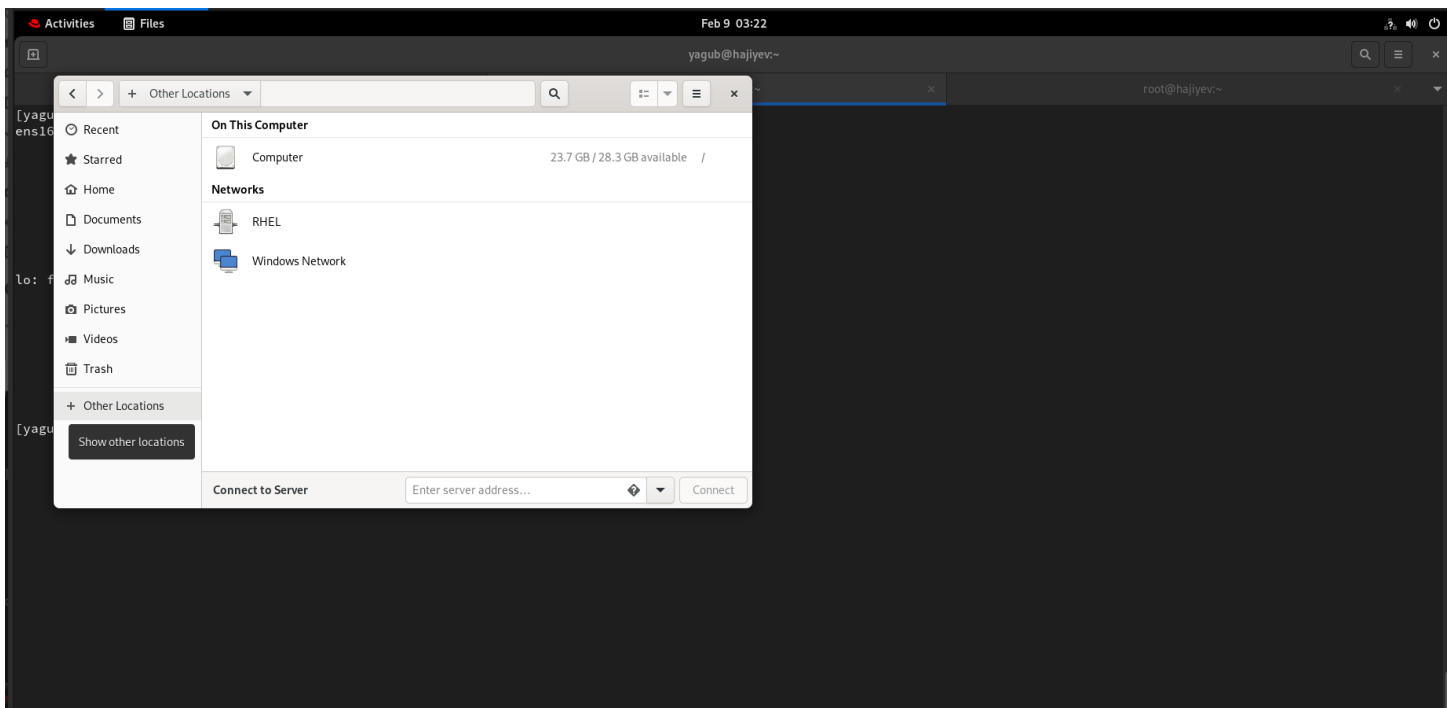
Now we are ready to use Samba shared folder in our system. Firstly we get our machine's IP address (IPv4).

```
[yagub@hajiyeve ~]$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.71 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::20c:29ff:fe30:9029 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:30:90:29 txqueuelen 1000 (Ethernet)
    RX packets 291 bytes 28082 (27.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 397 bytes 41079 (40.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

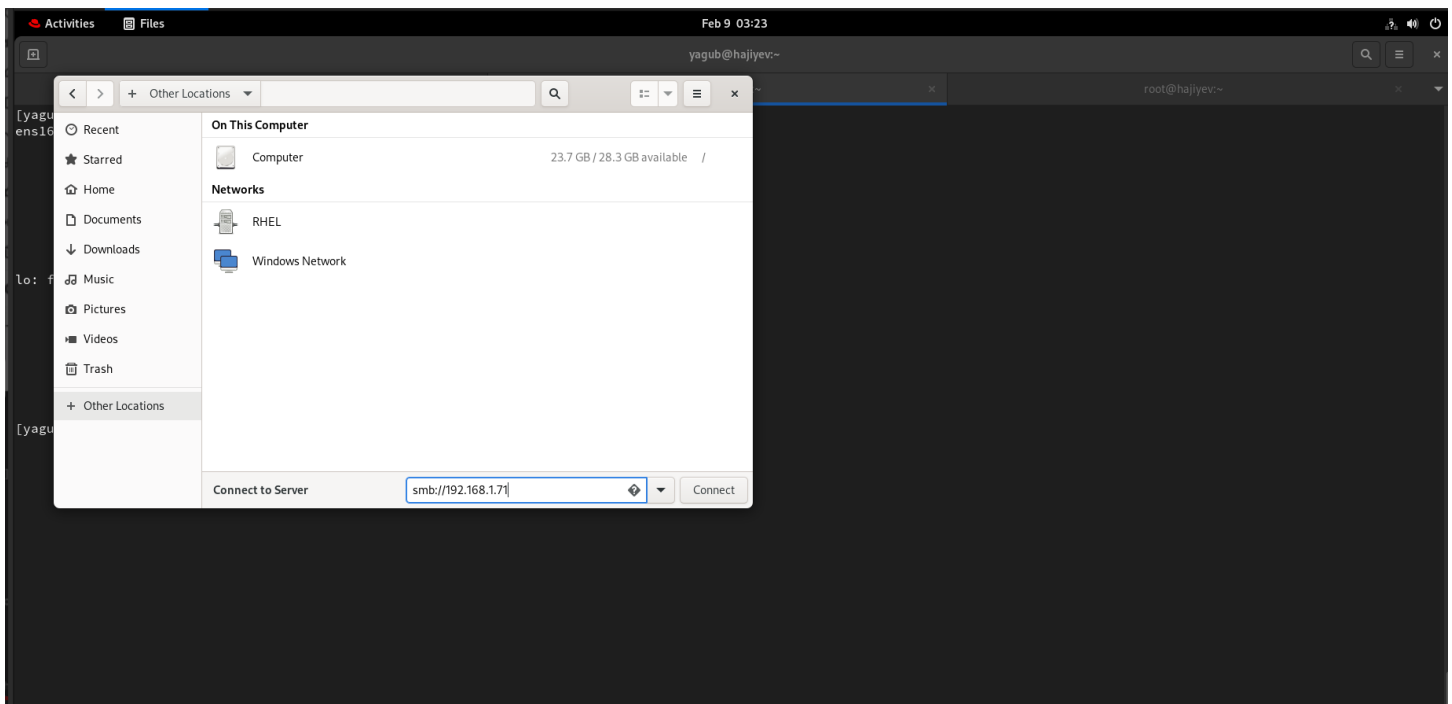
Next we enter the file manager in our Linux as shown below in the figure.



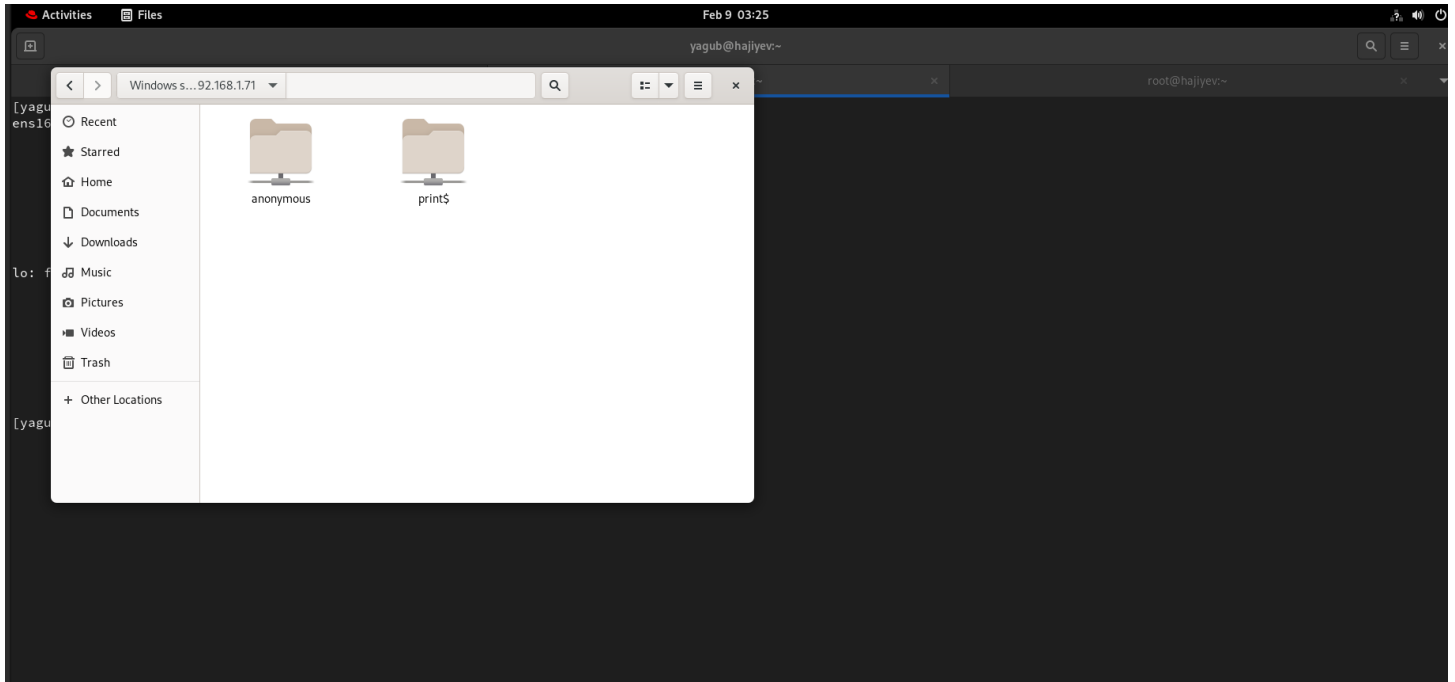
In the popped window, enter to the 'Other Locations' on the left of the window.



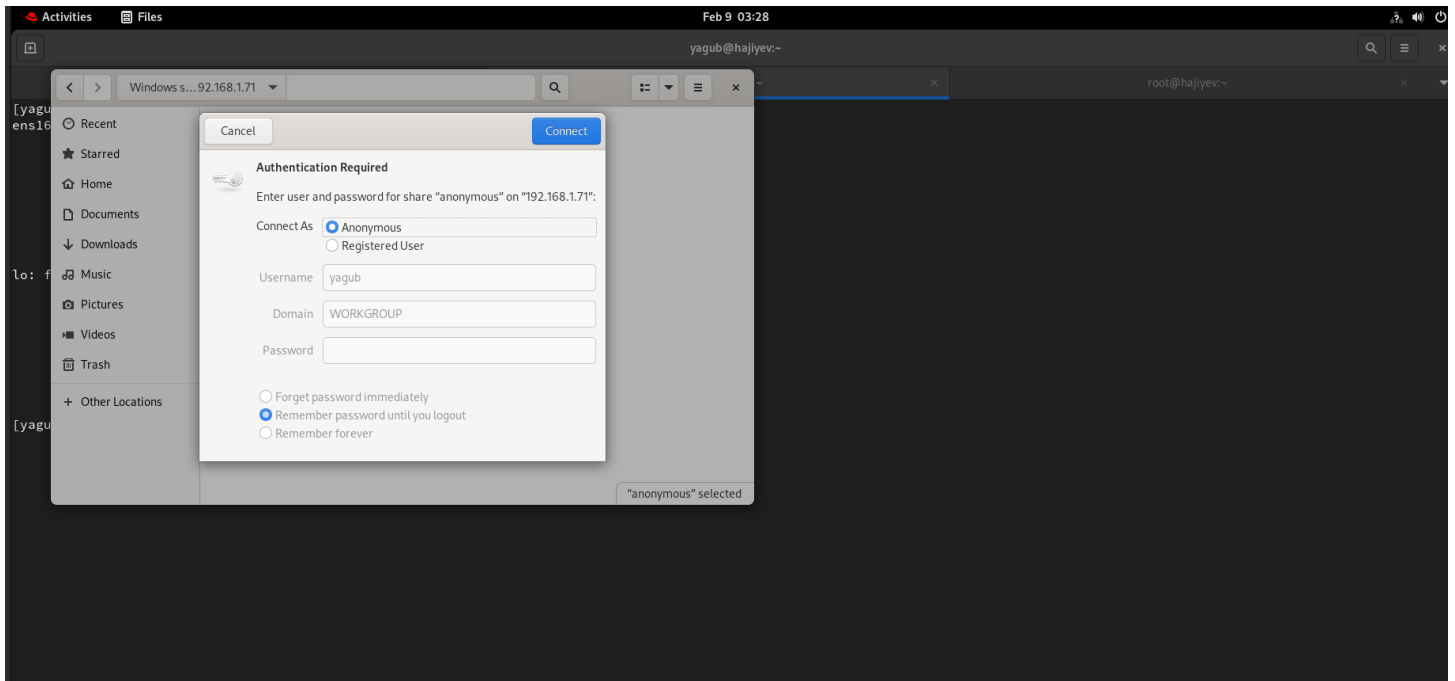
Later enter the 'smb://<your IP>' to the 'Enter server address...' bar and press Enter.



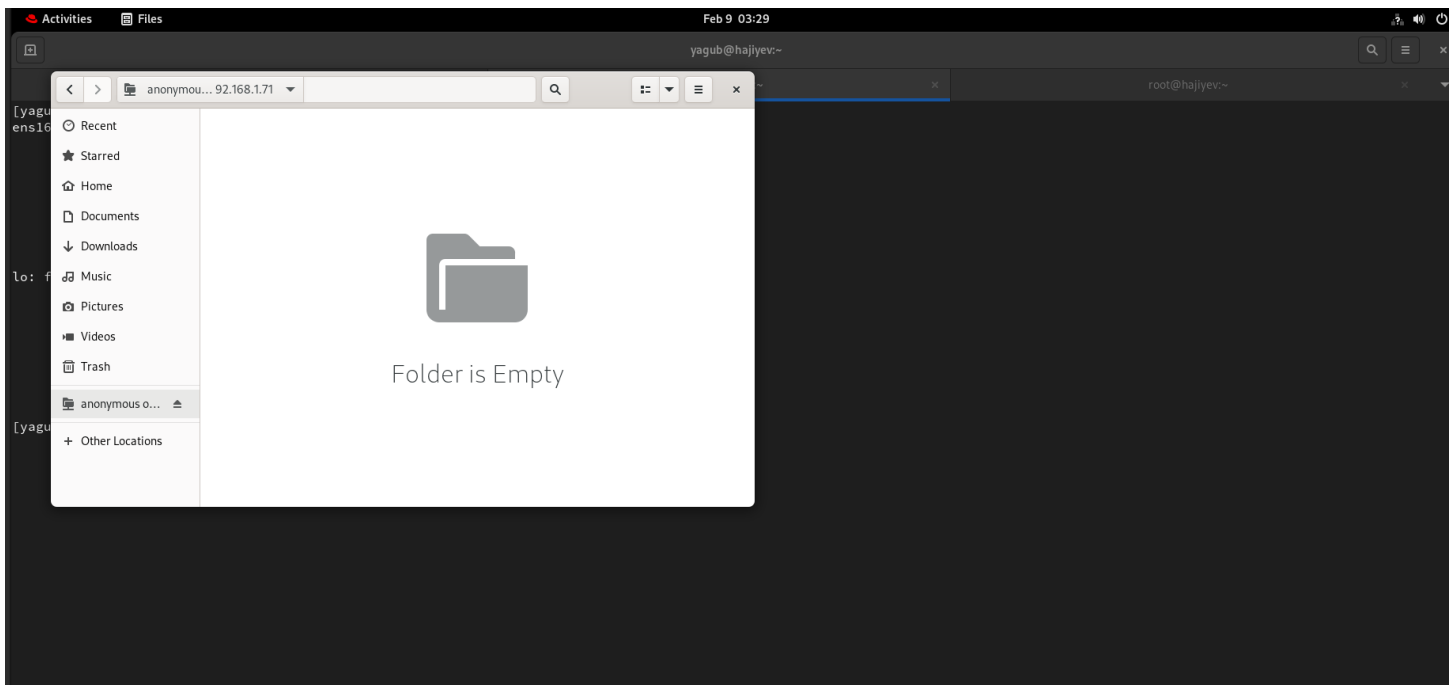
This command will take us to the shared folders where the 'anonymous' is our anonymously shared folder. Click on the 'anonymous' folder.



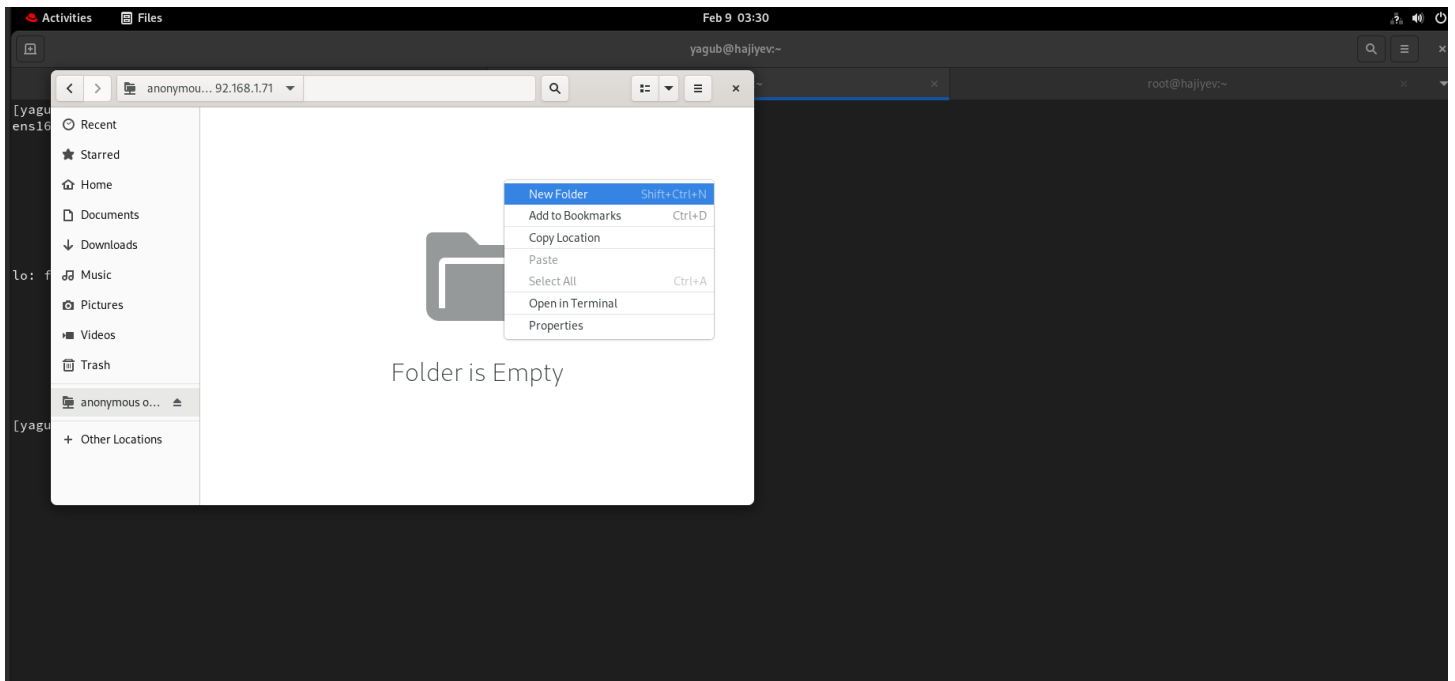
Click to the 'connect' button on the popped window.

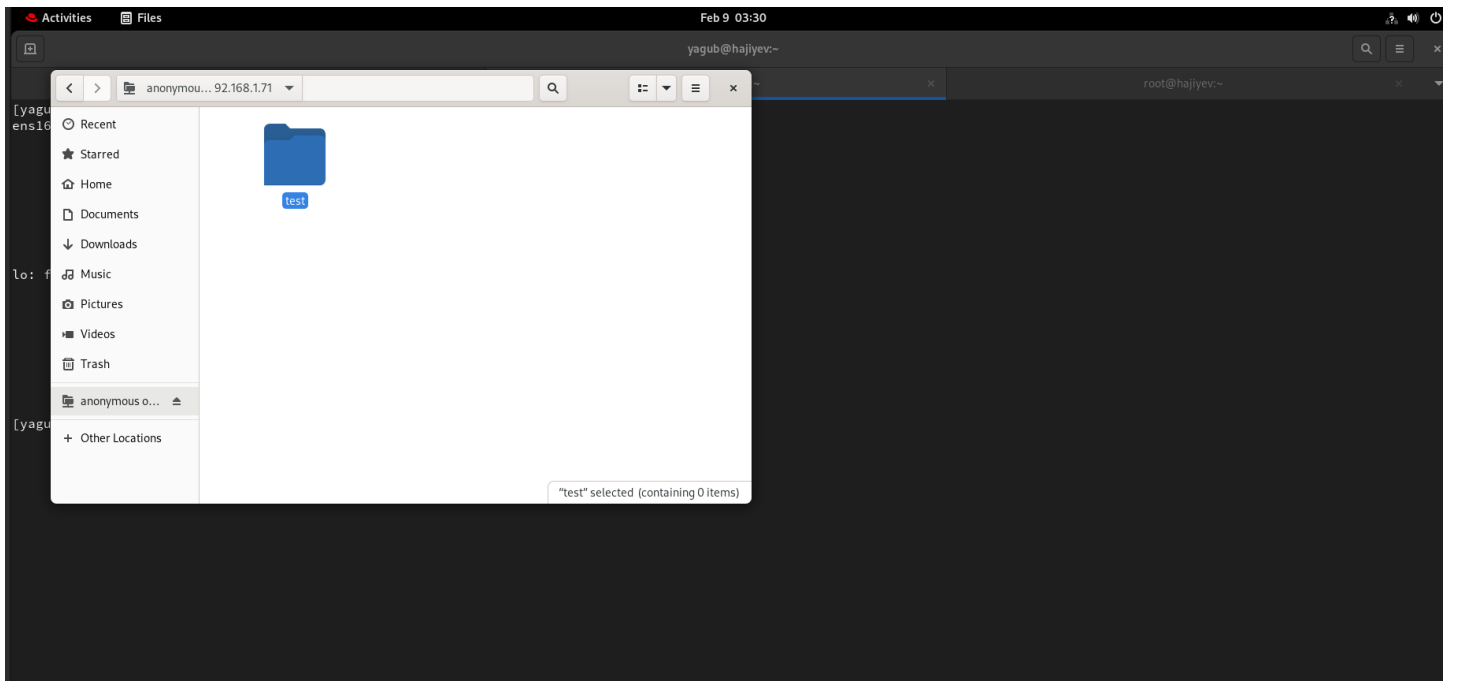
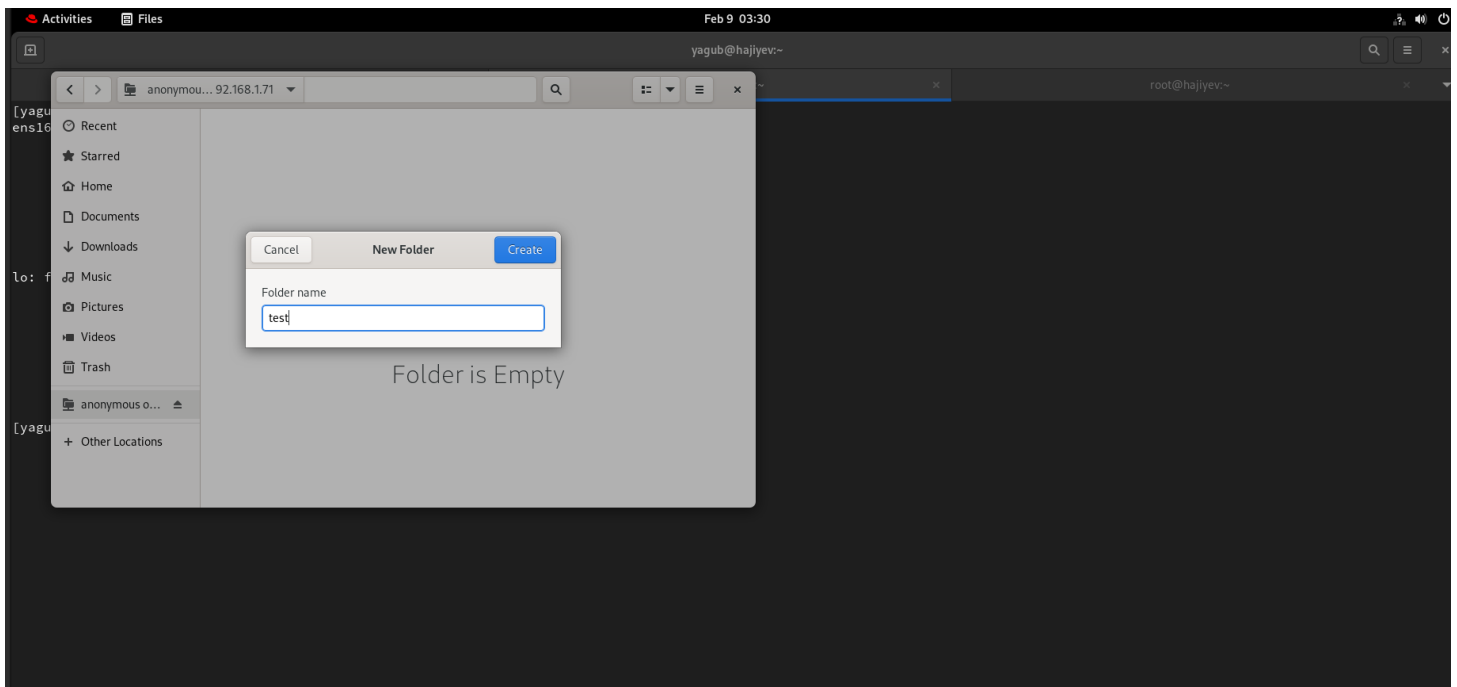


This empty folder is our 'CS301' folder which we created above.



In order to see where we are, just right click in the folder and add new folder. In this example I named 'test' this new folder.





Now back to command line in the terminal and list the content of the 'CS301' directory.

```
[yagub@hajiyevev ~]$ cd /CodeAcademy/CS301/  
[yagub@hajiyevev CS301]$ ll  
total 0  
drwxr-xr-x. 2 nobody nobody 6 Feb  9 03:33 test  
[yagub@hajiyevev CS301]$
```

As you see our test folder is in the 'CS301' directory.

## Adding users

In this step we will add some users and assign passwords to these users. Candidates will must use one of them in order to establish the SSH connection. This is the only way to enumerate into the target host.

```
[yagub@hajiyevev ~]$ sudo useradd Nariman
Creating mailbox file: File exists
[yagub@hajiyevev ~]$ echo HeadOfCS |sudo passwd --stdin Nariman
Changing password for user Nariman.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$ sudo useradd Natig
[yagub@hajiyevev ~]$ echo Network_King|sudo passwd --stdin Natig
Changing password for user Natig.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$ sudo useradd Gasim
[yagub@hajiyevev ~]$ echo RedHat |sudo passwd --stdin Gasim
Changing password for user Gasim.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$ sudo useradd Farrukh
[yagub@hajiyevev ~]$ echo WhiteHat |sudo passwd --stdin Farrukh
Changing password for user Farrukh.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$ sudo useradd Vidadı
[yagub@hajiyevev ~]$ echo Anonymous |sudo passwd --stdin Vidadı
Changing password for user Vidadı.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$ sudo useradd Vusal
[yagub@hajiyevev ~]$ echo CyberSoldier |sudo passwd --stdin Vusal
Changing password for user Vusal.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$
```

```
[yagub@hajiyevev ~]$ ll /home/
total 4
drwx-----. 3 Farrukh Farrukh    78 Feb  9 04:54 Farrukh
drwx-----. 3 Gasim    Gasim      78 Feb  9 04:53 Gasim
drwx-----. 3 Nariman  Nariman   78 Feb  9 04:52 Nariman
drwx-----. 3 Natig    Natig      78 Feb  9 04:53 Natig
drwx-----. 3 Vidadı   Vidadı    78 Feb  9 04:55 Vidadı
drwx-----. 3 Vusal    Vusal      78 Feb  9 04:55 Vusal
drwx-----. 14 yagub    yagub     4096 Feb  9 02:39 yagub
[yagub@hajiyevev ~]$
```

# MYSQL

The next service is the MYSQL, we are going to install. The MYSQL is a database service, in this lab we will use it for storing usernames and passwords (passwords will be hashed).

To install the service, type 'dnf install -y mysql-server' command and run it.

```
[yagub@hajiyeve ~]$ sudo dnf install -y mysql-server
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 2:38:01 ago on Fri 09 Feb 2024 02:30:41 AM EST.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
-----
Installing:
mysql-server                           x86_64            8.0.32-1.el9_2     a                   17 M
Installing dependencies:
mariadb-connector-c-config             noarch            3.2.6-1.el9_0     a                   11 k
mecab                                   x86_64            0.996-3.el9_4     a                   360 k
mysql                                  x86_64            8.0.32-1.el9_2     a                   2.8 M
mysql-common                           x86_64            8.0.32-1.el9_2     a                   79 k
mysql-errmsg                           x86_64            8.0.32-1.el9_2     a                   493 k
mysql-selinux                           noarch            1.0.5-1.el9_0     a                   37 k
protobuf-lite                           x86_64            3.14.0-13.el9     a                   235 k
=====
Transaction Summary
-----
Install 8 Packages

Total size: 21 M
Installed size: 179 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Installing               : mariadb-connector-c-config-3.2.6-1.el9_0.noarch 1/8
  Installing               : mysql-common-8.0.32-1.el9_2.x86_64 2/8
  Installing               : mysql-8.0.32-1.el9_2.x86_64 3/8
  Installing               : mysql-errmsg-8.0.32-1.el9_2.x86_64 4/8
  Installing               : protobuf-lite-3.14.0-13.el9.x86_64 5/8
Running scriptlet: mysql-selinux-1.0.5-1.el9_0.noarch 6/8
Installing               : mysql-selinux-1.0.5-1.el9_0.noarch 6/8
Running scriptlet: mysql-selinux-1.0.5-1.el9_0.noarch 6/8
libsemanage.semanage_direct_install_info: Overriding mysql module at lower priority 100 with module at priority 200.
Installing               : mecab-0.996-3.el9_4.x86_64 7/8
Running scriptlet: mysql-server-8.0.32-1.el9_2.x86_64 8/8
Installing               : mysql-server-8.0.32-1.el9_2.x86_64 8/8
Running scriptlet: mysql-server-8.0.32-1.el9_2.x86_64 8/8
Running scriptlet: mysql-selinux-1.0.5-1.el9_0.noarch 8/8
Running scriptlet: mysql-server-8.0.32-1.el9_2.x86_64 8/8
Verifying                 : mariadb-connector-c-config-3.2.6-1.el9_0.noarch 1/8
Verifying                 : mecab-0.996-3.el9_4.x86_64 2/8
Verifying                 : mysql-8.0.32-1.el9_2.x86_64 3/8
Verifying                 : mysql-common-8.0.32-1.el9_2.x86_64 4/8
```

Like other services, after installing the MYSQL service we need to enable and start it manually.

```
[yagub@hajiyeve ~]$ sudo systemctl enable --now mysqld.service
Created symlink /etc/systemd/system/multi-user.target.wants/mysqld.service → /usr/lib/systemd/system/mysqld.service.
[yagub@hajiyeve ~]$ sudo systemctl status mysqld.service
● mysqld.service - MySQL 8.0 database server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-02-09 05:10:28 EST; 13s ago
     Process: 10321 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
     Process: 10343 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mysqld.service (code=exited, status=0/SUCCESS)
    Main PID: 10418 (mysqld)
      Status: "Server is operational"
     Tasks: 39 (limit: 10681)
    Memory: 413.5M
       CPU: 5.028s
    CGroup: /system.slice/mysqld.service
           └─10418 /usr/libexec/mysqld --basedir=/usr

Feb 09 05:10:22 hajiyeve systemd[1]: Starting MySQL 8.0 database server...
Feb 09 05:10:22 hajiyeve mysql-prepare-db-dir[10343]: Initializing MySQL database
Feb 09 05:10:28 hajiyeve systemd[1]: Started MySQL 8.0 database server.
[yagub@hajiyeve ~]$
```

When we make sure that our service is running properly, we need to add it into the firewall configuration.

Don't forget that if we don't add the service to firewall then the firewall will block our traffic with outside, this is why it is have to done each time after installing new service or changing the service's default port.



```
[root@hajiyevev ~]# firewall-cmd --add-service=mysql --permanent
success
[root@hajiyevev ~]# firewall-cmd --reload
success
[root@hajiyevev ~]# firewall-cmd --list-services
cockpit dhcpv6-client mysql samba ssh
[root@hajiyevev ~]#
```

In this step we will add two information into the MYSQL service's configuration file, the 'bind-address' which is used to allowing only entered IPs to establish a connection with the service (default is 'localhost' or loopback IP but we entered 0.0.0.0 which allow all networks) and the 'skip-networking' line what is used for defining which TCP/IP connections will be allowed (0- listen all clients, 1- listen only local clients).

```
[yagub@hajiyevev ~]$ sudo vim /etc/my.cnf
```

```
#
# This group is read both both by the client and the server
# use it for options that affect everything
#
[client-server]
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
~
~
```

```
#
# This group is read both both by the client and the server
# use it for options that affect everything
#
[client-server]
bind-address = 0.0.0.0
skip-networking = 0
#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
~
~
```

After modifying the service's configuration file, we restart the MYSQL and check the status to make sure everything is ok.

```
[yagub@hajiyevev ~]$ sudo systemctl restart mysqld.service
[yagub@hajiyevev ~]$ sudo systemctl status mysqld.service
● mysqld.service - MySQL 8.0 database server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
   Active: active (running) since Fri 2024-02-09 05:20:36 EST; 10s ago
     Process: 10808 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
     Process: 10830 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mysqld.service (code=exited, status=0/SUCCESS)
    Main PID: 10864 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 10681)
       Memory: 365.8M
          CPU: 738ms
       CGroup: /system.slice/mysqld.service
               └─10864 /usr/libexec/mysqld --basedir=/usr

Feb 09 05:20:35 hajiyevev systemd[1]: Starting MySQL 8.0 database server...
Feb 09 05:20:36 hajiyevev systemd[1]: Started MySQL 8.0 database server.
[yagub@hajiyevev ~]$
```

Now we are ready to enter our server and create new databases.

In the MYSQL service, we can empty password authenticateto the server with the 'root' user, that means there is no need to enter the password. All we need to do is type the 'mysql -u root' command and just press Enter key whenever it requires a

password. If you login successfully you will meet the mysql shell environment as shown in the figure below.

After entering the server we create a new user who will be used by candidates for authenticating to our server. The command syntax is "CREATE USER '<username>'@<IP or domain>' IDENTIFIED BY '<password>' ". In our example, the percent sign (%) represent all IP address.

```
[root@hajiyevev ~]# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 899
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'Mamed'@'%' IDENTIFIED BY 'wrestler';
Query OK, 0 rows affected (0.02 sec)

mysql> █
```

We can check the userlist with the help of 'SELECT USER, host FROM mysql.user' command in the server. We can see that only the 'Mamed' user has permission to login server from different host, and rest of the users have permissions only in local host.

```
mysql> SELECT USER, host FROM mysql.user;
+-----+-----+
| USER                | host          |
+-----+-----+
| Mamed                | %             |
| mysql.infoschema     | localhost    |
| mysql.session        | localhost    |
| mysql.sys            | localhost    |
| root                 | localhost    |
+-----+-----+
5 rows in set (0.00 sec)

mysql> █
```

In this step we quit from the 'root' user and login with the 'Mamed' user.

```
[root@hajiyevev ~]# mysql -u Mamed -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 900
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

After successful login, we create the new database named 'Admins' with the 'CREATE DATABASE <database name>;' command and list databases.

```
mysql> CREATE DATABASE Admins;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| Admins   |
| information_schema |
| mysql    |
| performance_schema |
| sys      |
+-----+
5 rows in set (0.00 sec)
```

The 'USE <database name>;' command let us to enter the any databases which we used to enter the 'Admins' database, created new table named the 'Passwords' with two columns, 'Users' and 'Passwords', where we will store usernames and passwords. The 'varchar(20)' and the 'varchar(500)' mean that in the 'Users' column values' max length can be 20 characters and 500 characters length for the 'Passwords' column, and the 'NOT NULL' define that you can't enter null value into the 'Passwords' column.

**Note:** after typing 'CREATE TABLE Passwords(' press Enter.

```
mysql> USE Admins;
Database changed
mysql> CREATE TABLE Passwords(
  -> Users varchar(20),
  -> Passwords varchar(500) NOT NULL
  -> );
Query OK, 0 rows affected (0.03 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_Admins |
+-----+
| Passwords         |
+-----+
1 row in set (0.00 sec)

mysql>
```

In the figure, we have to enter values into the table, the general syntax is "INSERT INTO <table name> (column1, column2) VALUES ('<username>', <password>'; ". In this lab example I hashed passwords with the help of website. You can see the example website in the second figure.

```
mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Administrator', 'd03805fbfa4b41f1c6f01afba246156a');
Query OK, 1 row affected (0.26 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('root', 'f94198fa6094ac2dd1ef66566296927a');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Gasim', '3e96e648d12e5cd923fb3a689e979ad1');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Natig', 'e47ae20475213c4a5460031d53856973');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Farrukh', '70e0ea26808d782a12b5d3a6a10c61d5');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Vidadi', 'f91ab02f90892f57c51de112d2f544d8');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Vusal', 'e08d3b12011146058ef0bcf552b41be9');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Yagub', '04bf7f361d5bb352bb69c22715d3d41c');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Passwords (Users, Passwords) VALUES ('Nariman', '2d1e50d4f80a9a6d7e02abbfc271b645');
Query OK, 1 row affected (0.01 sec)

mysql>
```

The screenshot shows a web browser window with the URL <https://www.md5hashgenerator.com>. The page has a navigation bar with links like "Dan's Tools", "Web Dev", "Conversion", "Encoders / Decoders", "Formatters", "Internet", and "English". The main content area is titled "MD5 Hash Generator" and includes a text input field containing "CS301\_Student". Below the input field is a "Generate" button. A table displays the results of the hash generation:

Your String	
CS301_Student	
MD5 Hash	04bf7f361d5bb352bb69c22715d3d41c <a href="#">Copy</a>
SHA1 Hash	213103bce8cc3348dd2321e983856f0fdb1b86e2 <a href="#">Copy</a>

Below there are 8 users and only Administrator is the fake user and the 'Yagub' will be created in the following sections. We will assign the password 'CS301\_Student' to the 'Yagub' user and only its password is valid because others hashes are just random words.

Later we will create a wordlist and put it into the FTP server for candidates to use for cracking hashes to get the password of the 'Yagub' user.

```
mysql> SELECT * FROM Passwords;
```

Users	Passwords
Administrator	d03805fbfa4b41f1c6f01afba246156a
root	f94198fa6094ac2dd1ef66566296927a
Gasim	3e96e648d12e5cd923fb3a689e979ad1
Natig	e47ae20475213c4a5460031d53856973
Farrukh	70e0ea26808d782a12b5d3a6a10c61d5
Vidadi	f91ab02f90892f57c51de112d2f544d8
Vusal	e08d3b12011146058ef0bcf552b41be9
Yagub	04bf7f361d5bb352bb69c22715d3d41c
Nariman	2d1e50d4f80a9a6d7e02abbfc271b645

```
9 rows in set (0.00 sec)
```

```
mysql> █
```

# NFS

The last service for the lab is the NFS, what we can install with the 'dnf install -y nfs-utils' command.

```
[root@hajiyeve ~]# dnf install nfs-utils -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Last metadata expiration check: 3:41:08 ago on Fri 09 Feb 2024 05:52:31 AM EST.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
nfs-utils                               x86_64            1:2.5.4-20.el9   b                458 k
Installing dependencies:
gssproxy                               x86_64            0.8.4-6.el9      b                114 k
keyutils                               x86_64            1.6.3-1.el9      b                78 k
libev                                   x86_64            4.33-5.el9       b                56 k
libnfsidmap                            x86_64            1:2.5.4-20.el9   b                66 k
libverto-libev                         x86_64            0.3.2-3.el9      b                15 k
rpcbind                                x86_64            1.2.6-5.el9      b                62 k
sssd-nfs-idmap                         x86_64            2.9.1-2.el9      b                45 k
=====
Transaction Summary
=====
Install 8 Packages

Total size: 893 k
Installed size: 2.1 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                :
  Installing                : libnfsidmap-1:2.5.4-20.el9.x86_64
  Running scriptlet: rpcbind-1.2.6-5.el9.x86_64
  Installing                : rpcbind-1.2.6-5.el9.x86_64
  Running scriptlet: rpcbind-1.2.6-5.el9.x86_64
Created symlink /etc/systemd/system/multi-user.target.wants/rpcbind.service → /usr/lib/systemd/system/rpcbind.service.
Created symlink /etc/systemd/system/sockets.target.wants/rpcbind.socket → /usr/lib/systemd/system/rpcbind.socket.
  Installing                : libev-4.33-5.el9.x86_64
  Installing                : libverto-libev-0.3.2-3.el9.x86_64
1/1
1/8
2/8
2/8
2/8
3/8
4/8
```

This service also have to be enabled and started after the installation.

```
[root@hajiyeve ~]# systemctl enable --now nfs-server.service
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /usr/lib/systemd/system/nfs-server.service.
[root@hajiyeve ~]# systemctl status nfs-server.service
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; preset: disabled)
   Active: active (exited) since Fri 2024-02-09 09:34:51 EST; 12s ago
     Process: 4369 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
     Process: 4370 ExecStart=/usr/sbin/rpc.nfsd (code=exited, status=0/SUCCESS)
     Process: 4388 ExecStart=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload gssproxy ; fi (code=exited, status=0/SUCCESS)
    Main PID: 4388 (code=exited, status=0/SUCCESS)
      CPU: 27ms

Feb 09 09:34:51 hajiyeve systemd[1]: Starting NFS server and services...
Feb 09 09:34:51 hajiyeve systemd[1]: Finished NFS server and services.
[root@hajiyeve ~]#
```

Apart from other services here we have to add the NFS into the firewall with another 2 services, 'mountd' and 'rpc-bind', as demonstrated below.

```
[root@hajiyeve ~]# firewall-cmd --add-service={mountd,nfs,rpc-bind} --permanent
success
[root@hajiyeve ~]# firewall-cmd --reload
success
[root@hajiyeve ~]# firewall-cmd --list-services
cockpit dhcpv6-client ftp mountd mysql nfs rpc-bind samba ssh
[root@hajiyeve ~]#
```

In our lab we will share the entire '/home' directory for candidates to use in order to establish the SSH connection (we know there are users home directories in the '/home' folder, so they can use these directories for copyin their public keys to the any user and get access over that user).

For this purpose we have to add the '/home \*(rw)' line into the 'exports' file. The '\*' sign represent that the NFS service will share the directory to everyone and everyone will have read and write (rw) permissions.

```
[root@hajiyevev ~]# vim /etc/exports
[root@hajiyevev ~]# cat /etc/exports
/home *(rw)
[root@hajiyevev ~]#
```

After modifying the 'exports' file we have to restart the service to apply changes. Below the 'showmount -e' command display shared directories ( '/home' in this case).

```
[root@hajiyevev ~]# systemctl restart nfs-server.service
[root@hajiyevev ~]# showmount -e
Export list for hajiyevev:
/home *
[root@hajiyevev ~]#
```

We can check the shared folders in other machine with this command 'showmount -e <target host's IP>'.



## Final steps

In this section we will complete our lab by creating wordlist for cracking hashed passwords (look MYSQL section), creating the 'Yagub' user, creating encoded password for brute force attack and creating final file which will contain the code to finish the puzzle. Lets begin with creating wordlist.

In the figure we enter to the '/var/ftp' folder (what is shared with the FTP) and create the file named 'wordlist.txt'. Candidates will need this file to crack the hashed passwords.

```
[yagub@hajiyevev ~]$ cd /var/ftp/
[yagub@hajiyevev ftp]$ sudo touch wordlist.txt
[yagub@hajiyevev ftp]$ vim wordlist.txt
[yagub@hajiyevev ftp]$ ll
total 8
drwxr-xr-x. 2 root root    6 May  9  2023 pub
-rw-r--r--. 1 root root 134 Feb  9 02:22 users.txt
-rw-r--r--. 1 yagub yagub 448 Feb  9 06:58 wordlist.txt
[yagub@hajiyevev ftp]$
```

Here is the content of the 'wordlist.txt' file, where I placed the 'Yagub' user's password 'CS301\_Student'.

```
Redhat_king
networker
Whitehat
anonymous_cs
cs_301
man_in-themiddle
csstudent
Code_Student
Student_Cs301
301-Student_cs
Cyber_Stundet
Attacker
war_inthepiece
Porsche_911
Life_is_hard
StudentOfAcademy
CyberSoldier
CS301-Student
WhoAmI
CS301_Student
FinalExam
Work_hard123
HelloWorld!
Salam123
Adminadmin
rootismine
KharabagIsAzerbaijan
BakuTheCapital
CaspianSea
CS-Code_301
number
apple
glasses2398
mcals
saxioi
-- INSERT --
```

We had been talked about the user 'Yagub' above, now it is time to create this user. Below with the help of 'useradd -M Yagub' command I created the user and assign it the password 'CS301\_Student'. The '-M' option lets us to add new user without creating its home directory (if this user has a home directory under the '/home' folder then candidates would use it to establish direct SSH connection but we want force them to crack the hashed passwords in MYSQL and use it)

```
[yagub@hajiyevev ~]$ sudo useradd -M Yagub
Creating mailbox file: File exists
[yagub@hajiyevev ~]$ echo CS301_Student | sudo passwd --stdin Yagub
Changing password for user Yagub.
passwd: all authentication tokens updated successfully.
[yagub@hajiyevev ~]$ su - Yagub
Password:
su: warning: cannot change directory to /home/Yagub: No such file or directory
[Yagub@hajiyevev yagub]$
```

Below I append the user 'Yagub' into the 'wheel' group to give him root privileges and removed the user 'yagub' from the 'wheel' group.

```
[yagub@hajiyevev ~]$ sudo usermod -aG wheel Yagub
[yagub@hajiyevev ~]$ id Yagub
uid=1007(Yagub) gid=1007(Yagub) groups=1007(Yagub),10(wheel)
[yagub@hajiyevev ~]$ su - Yagub
Password:
su: warning: cannot change directory to /home/Yagub: No such file or directory
[Yagub@hajiyevev yagub]$ sudo gpasswd -d yagub wheel
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

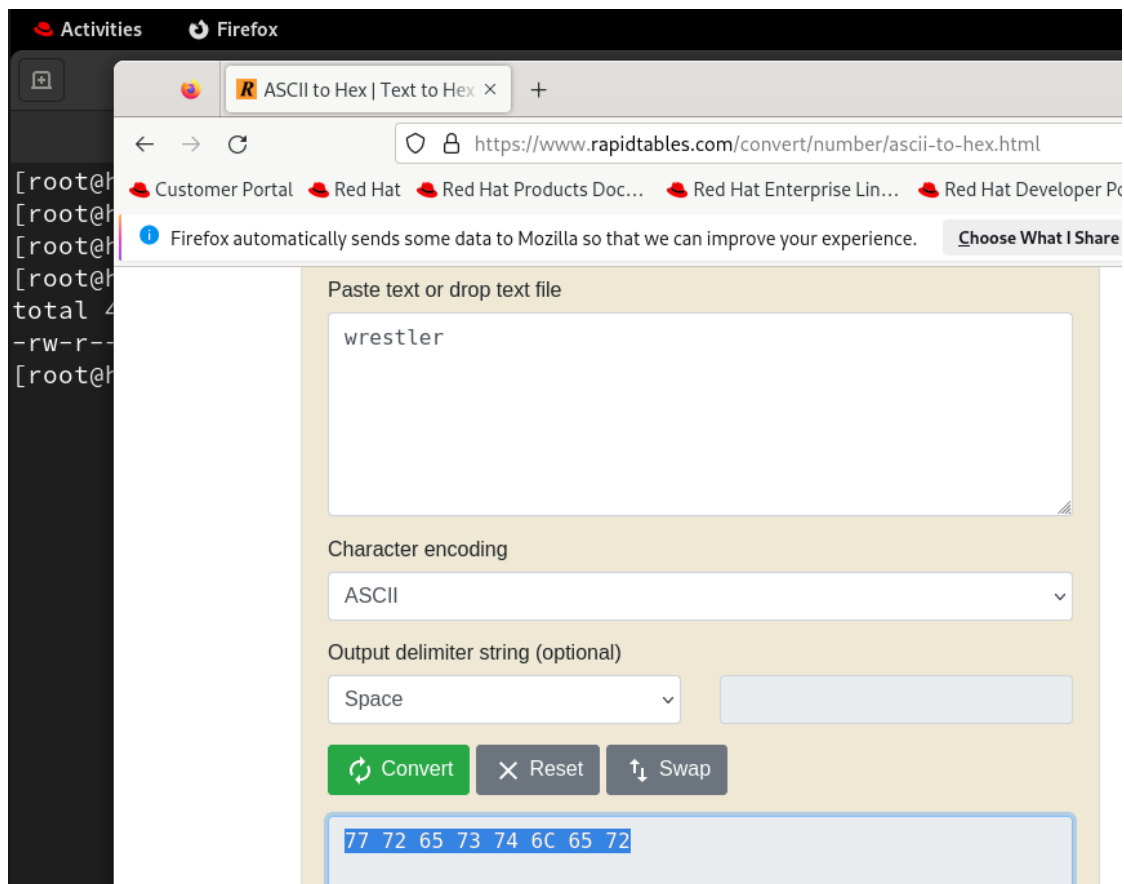
```
[sudo] password for Yagub:
Removing user yagub from group wheel
[Yagub@hajiyevev yagub]$ id yagub
uid=1000(yagub) gid=1000(yagub) groups=1000(yagub)
[Yagub@hajiyevev yagub]$
```

The next step is to create encoded passwords and add them into the list named 'passwords.txt'. For this purpose we must create the the in the 'test' folder what we created in the Samba section.

In order to encode passwords also this time I used a website on the internet and encoded them into the 'hexadecimal' codes whcih can be easily noticed by candidates. I created just 4 encoded passwords and put them in the 'passwords.txt' file but only

one them is the valid passowrd (the last one which is 'wrestler'). After decoding these passwords, candidates will use them to make brute force attack on the MYSQL server (you remember we created the 'Mamed' user and assign it the password 'wrestler').

```
[root@hajiyevev ~]# cd /CodeAcademy/CS301/test/
[root@hajiyevev test]# touch passwords.txt
[root@hajiyevev test]# vim passwords.txt
[root@hajiyevev test]# ll
total 4
-rw-r--r--. 1 root root 117 Feb  9 07:31 passwords.txt
[root@hajiyevev test]#
```



Below figure display the content of the 'passwords.txt' file and above figure is the website where I encoded the 'wrestler' password and some random words.

```
50 6F 72 73 63 68 65 5F 39 31 31
61 70 70 6C 65
59 61 67 75 62 5F 54 68 65 5F 4B 69 6E 67
77 72 65 73 74 6C 65 72
```

In the final step we create the file named 'cs301' and put in a random directory ('/usr/share/info' in this case). After creating the file we change its user owner and group owner to 'Yagub' and take all permissions from other users (to make sure that no one will read the content of the 'cs301' file except 'Yagub').

```
[root@hajiyevev ~]# cd /usr/share/info/  
[root@hajiyevev info]# touch cs301  
[root@hajiyevev info]# vim cs301  
[root@hajiyevev info]# chown Yagub.Yagub cs301  
[root@hajiyevev info]# chmod 0770 cs301
```

You can type whatever you want in the 'cs301' file, I wrote the code '1918' which is the foundation year of the Azerbaijan Republic.

```
[root@hajiyevev info]# cat cs301  
Congratulations!
```

```
-----  
-                                     -  
-                                     -  
-   code:1918                       -  
-                                     -  
-                                     -  
-----
```