RIT | Golisano College of Computing and Information Sciences
School of Information

NSSA221 Systems Administration
Lab 05: File Services

Introduction

As a systems administrator, it may be your responsibility to set up a file-sharing server and to configure the client, or user permissions to access the remote share. Understanding the protocols used, the daemon processes, and how to configure these services is a required skill set for a competent systems administrator.

In this lab, you will also gain experience using the **`rsync`** command. **`Rsync`** is a utility that securely copies files locally and remotely, including directory hierarchies. It is a little smarter than **`scp`** (Secure Copy) because of available options, such as only copying source files that are different from the destination files. **`Rsync`** is also an excellent tool for backing up data.

Lab Summary

In this lab, you will need the storage server from Lab 04, a Windows and Linux client. Using the partitions and mounts created on the storage server, you will configure remote shares, exports, and directories. You will install several services for file transfer (rsync and FTP) and sharing services (Samba and NFS) so that clients can remotely access these storage systems.

Goals

At the end of this lab, you will…

- Have experience deploying and managing File Transfer and File Sharing services.
- Install and configure the FTP, Samba, RSYNC, and NFS services.
- Use Rsync to back up data locally and remotely.
- Mounting NFS exports and Samba shares on Linux and Windows Systems.

Please Note that any reference to a server, unless otherwise noted, means the storage/RAID server created in Lab 4.

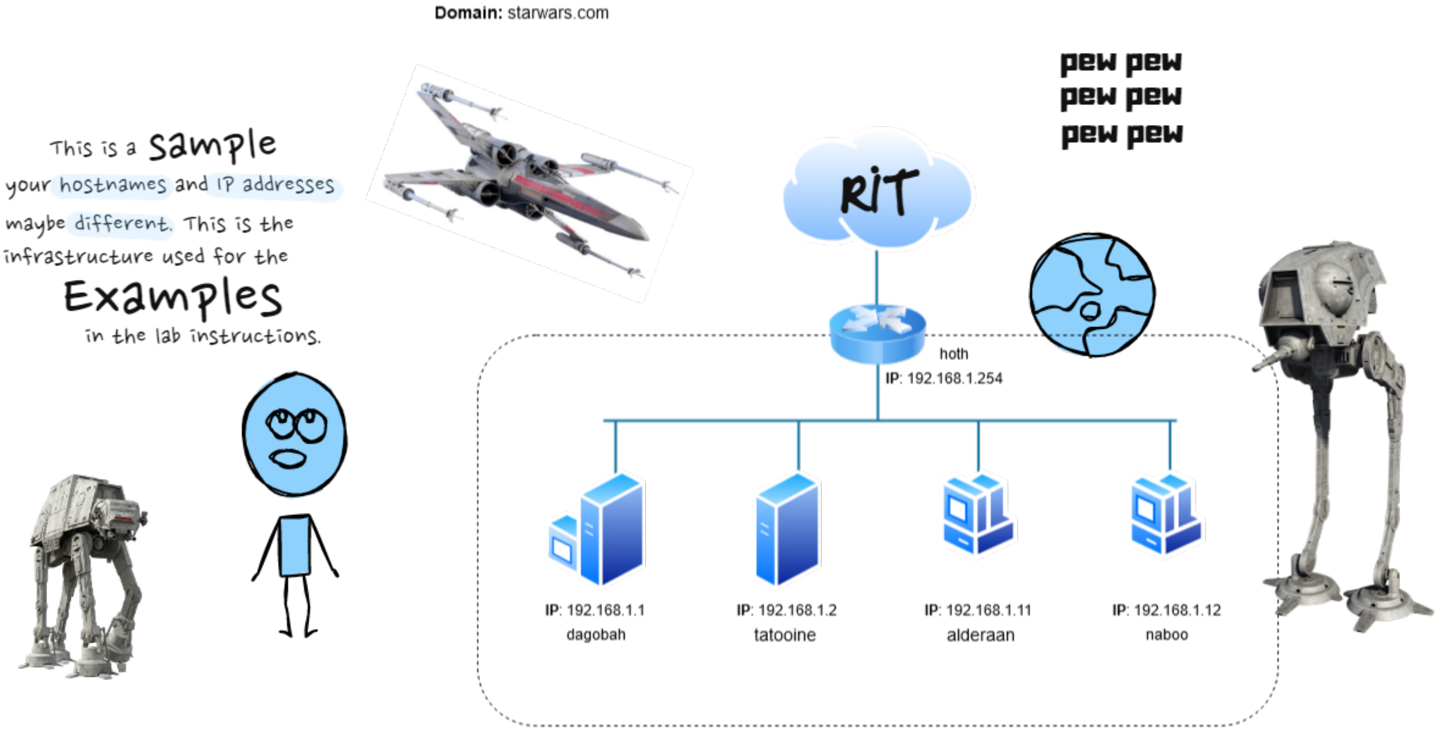
Preparation

- Read chapters 18, 19, and 20 from the Linux Bible.
- Read the rsync article posted to myCourses.
- Read the cron job article posted to myCourses.

Activity Summary

Activity 1 – Initial Set Up
Activity 2 – Rsync Basics
Activity 3 – Installing and Configuring FTP
Activity 4 – Creating a Drop Box
Activity 5 – Installing Samba and Creating Share
Activity 6 – Accessing Samba Shares from Windows
Activity 7 – Installing RSYNC and Creating an RSYNC Module
Activity 8 – Installing NFS and Creating a Simple NFS Share

Figure 1 – Lab Instructions Topology



This is a sample your hostnames and IP addresses maybe different. This is the infrastructure used for the Examples in the lab instructions.

Table 1 – Example Configuration Settings

Device	System	Hostname	FQDN	Configuration	IPv4 Address
pfSense/gateway	BSD	hoth	hoth.starwars.com	static/manual	192.168.1.254
Domain Controller	Windows	dagobah	dagobah.starwars.com	static/manual	192.168.1.1
Storage Server	Linux	tatooine	tatooine.starwars.com	static/manual	192.168.1.2
Windows Client	Windows	naboo	naboo.starwars.com	dynamic	192.168.1.11
Rocky Client	Linux	alderaan	alderaan.starwars.com	dynamic	192.168.1.12

Please note that these settings are for the lab instructions and used throughout as examples and are not indicative of your environment.

Activities

Activity 1 – Configuring DNS

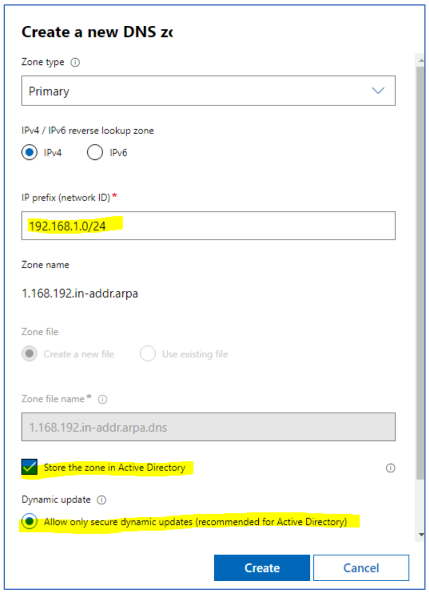
For the lab, we are going to start using hostnames instead of IP addresses for remote connections. For this activity, we will need to configure DNS on Windows Server 2022.

a. Windows Admin Center should already be installed on Windows 10 from Lab 2, go ahead and install the DNS extension. Remember, you need to connect to the server for it to show up under tools (Figure 2).

b. You may need to create a reverse zone for your network using the network ID. In the example topology, the network is 192.168.1.0/24. To create the reverse zone, click the ***"Reverse lookup zones"*** link, then select ***"Create a new DNS zone."***

c. Enter the required information (Figure 2), i.e., the network ID. Then click ***Create***, to create the reverse zone.

Figure 2 – Creating a Reverse Lookup Zone



Create a new DNS zc

Zone type: Primary

IPv4 / IPv6 reverse lookup zone: ● IPv4 ○ IPv6

IP prefix (network ID)*: 192.168.1.0/24

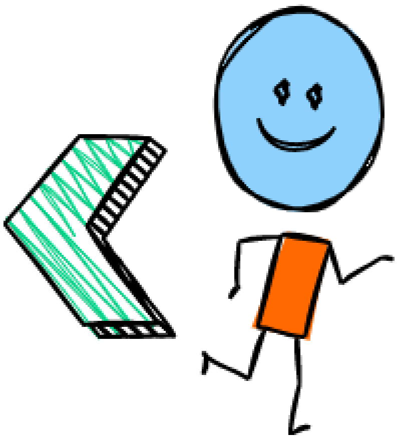
Zone name: 1.168.192.in-addr.arpa

Zone file: ● Create a new file ○ Use existing file

Zone file name*: 1.168.192.in-addr.arpa.dns

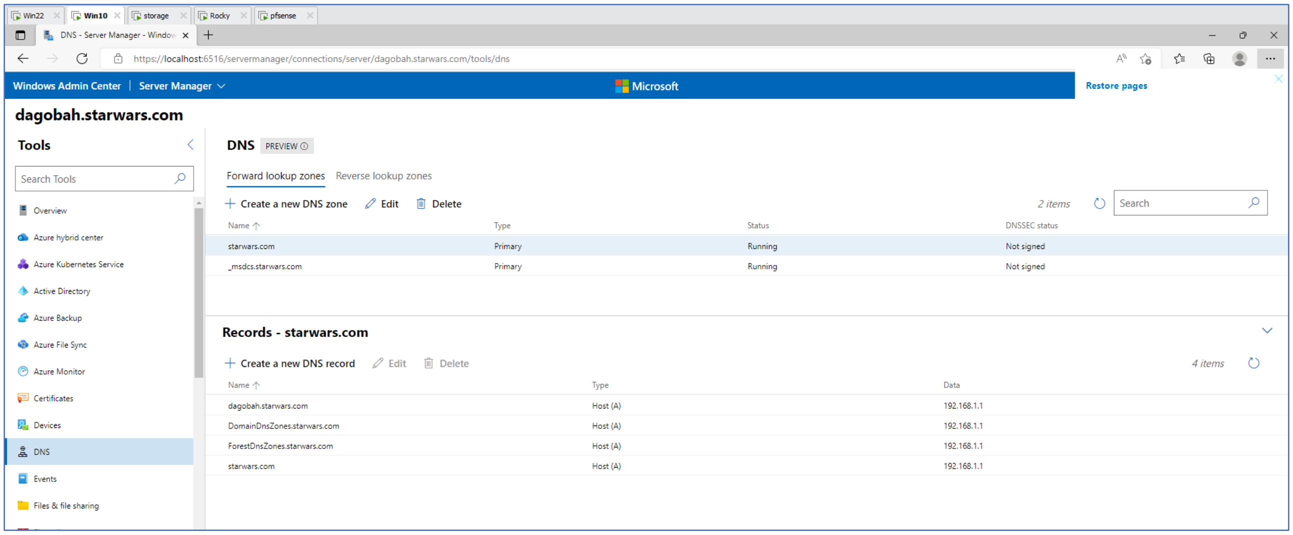
☑ Store the zone in Active Directory

Dynamic update: ● Allow only secure dynamic updates (recommended for Active Directory)



d. Referring to Figure 3, the A Resource Record for the Active Directory server is created for us, however we need to create an A record for the storage server. To do this click ***Create a new DNS record*** for your zone. In this example, the zone is ***starwars.com*** and the FQDN for the storage server is ***tatooine.starwars.com*** with an IP address *192.168.1.2*.

Figure 3 – Windows Admin Center DNS



dagobah.starwars.com

DNS

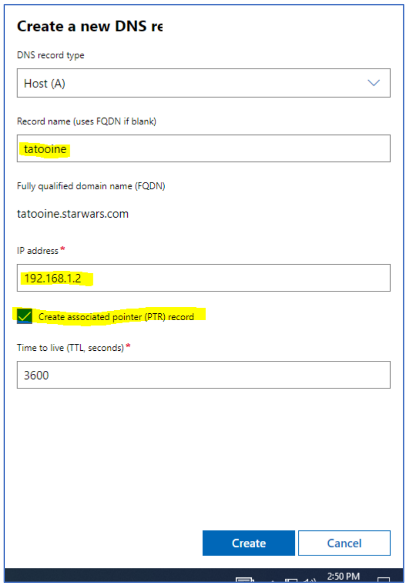
Name	Type	Status	DNSSEC status
starwars.com	Primary	Running	Not signed
_msdcs.starwars.com	Primary	Running	Not signed

Records - starwars.com

Name	Type	Data
dagobah.starwars.com	Host (A)	192.168.1.1
DomainDnsZones.starwars.com	Host (A)	192.168.1.1
ForestDnsZones.starwars.com	Host (A)	192.168.1.1
starwars.com	Host (A)	192.168.1.1

e. Fill in the required information and make sure to create the associated pointer (PTR) resource record. Figure 4. Click Create and repeat the process for all the devices in your infrastructure.

Figure 4 – Creating an A Resource Record



Create a new DNS re

DNS record type: Host (A)

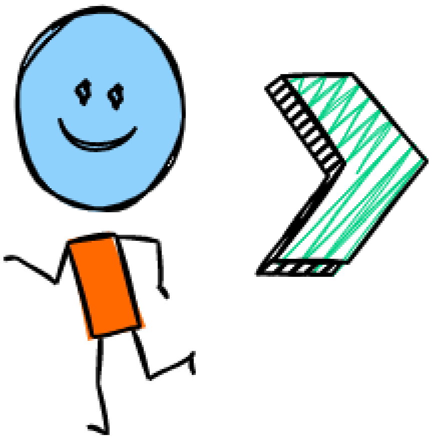
Record name (uses FQDN if blank): tatooine

Fully qualified domain name (FQDN): tatooine.starwars.com

IP address *: 192.168.1.2

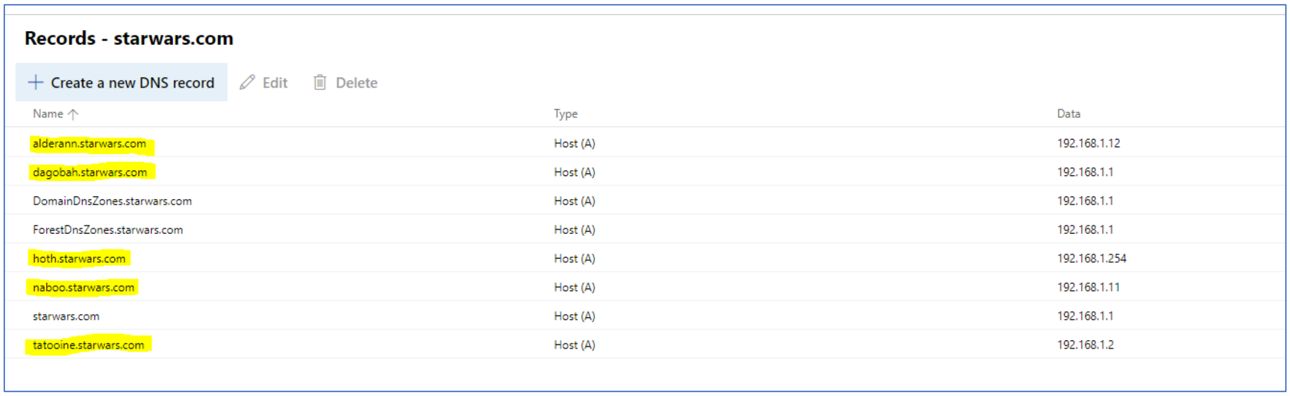
☑ Create associated pointer (PTR) record

Time to live (TTL, seconds) *: 3600



f. Using the information from Table 1, when you are done, you should have A Resource Records (Figure 5) for all devices/virtual machines in your domain and the associated reverse lookups and pointer records (Figure 6).

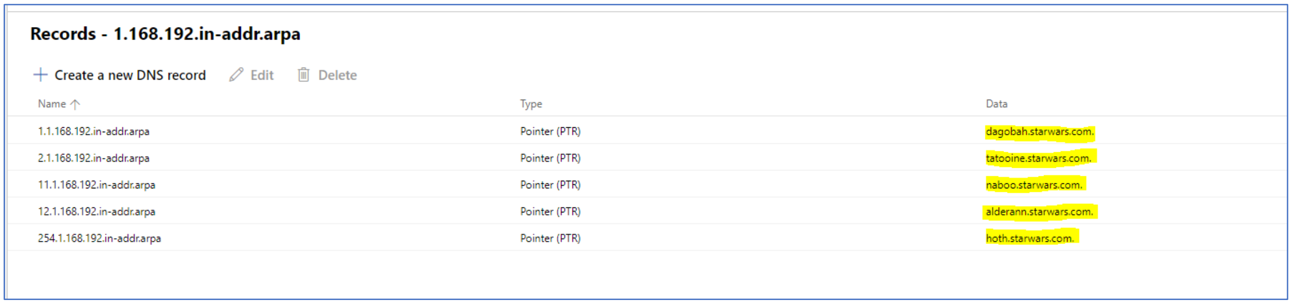
Figure 5 – Completed Forward Lookup Zone



Records - starwars.com

Name ↑	Type	Data
alderann.starwars.com	Host (A)	192.168.1.12
dagobah.starwars.com	Host (A)	192.168.1.1
DomainDnsZones.starwars.com	Host (A)	192.168.1.1
ForestDnsZones.starwars.com	Host (A)	192.168.1.1
hoth.starwars.com	Host (A)	192.168.1.254
naboo.starwars.com	Host (A)	192.168.1.11
starwars.com	Host (A)	192.168.1.1
tatooine.starwars.com	Host (A)	192.168.1.2

Figure 6 – Completed Reverse Lookup Zone



Records - 1.168.192.in-addr.arpa

Name ↑	Type	Data
1.1.168.192.in-addr.arpa	Pointer (PTR)	dagobah.starwars.com.
2.1.168.192.in-addr.arpa	Pointer (PTR)	tatooine.starwars.com.
11.1.168.192.in-addr.arpa	Pointer (PTR)	naboo.starwars.com.
12.1.168.192.in-addr.arpa	Pointer (PTR)	alderann.starwars.com.
254.1.168.192.in-addr.arpa	Pointer (PTR)	hoth.starwars.com.

IMPORTANT!

✓ For the Report

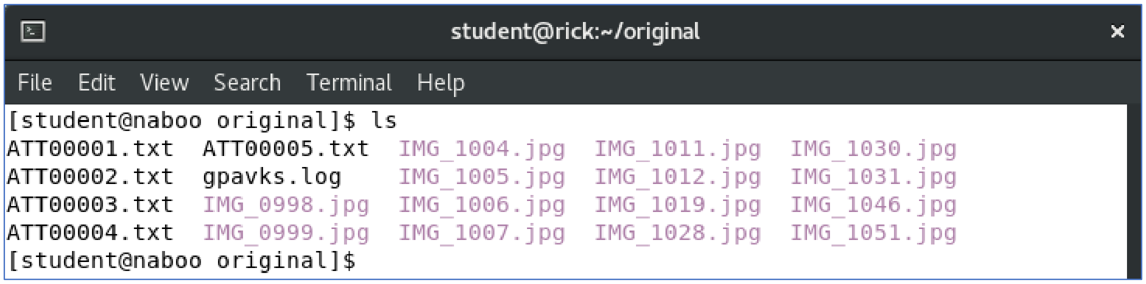
For the report, you will need two screenshots of the forward and reverse lookup zone configurations similar to Figures 5 and 6.

Activity 2 – Rsync Basics

Backing up files is a vital system administrator responsibility. In Lab 3, you tracked changes to files using Git and pushed them to a repository, which, in a roundabout way, backed those files up, and if you ever need to restore them, you can clone or pull the repository. In this activity, we will take a more direct approach by backing up files on the Linux client using the **`rsync`** utility and some of its more common options.

a. On the Linux client, create two directories titled "original" and "backup" in your home directory.
b. In myCourses, download the zipped file located in the "Lab Materials" section in the module titled "Sample Data." This zipped file contains a mix of jpeg and txt files (Figure 7) that are needed for the activity.

Figure 7 – Sample Data



```
[student@naboo original]$ ls
ATT00001.txt  ATT00005.txt  IMG_1004.jpg  IMG_1011.jpg  IMG_1030.jpg
ATT00002.txt  gpavks.log    IMG_1005.jpg  IMG_1012.jpg  IMG_1031.jpg
ATT00003.txt  IMG_0998.jpg  IMG_1006.jpg  IMG_1019.jpg  IMG_1046.jpg
ATT00004.txt  IMG_0999.jpg  IMG_1007.jpg  IMG_1028.jpg  IMG_1051.jpg
[student@naboo original]$
```

c. Extract the downloaded zipped files to the "original" directory (Figure 7). For information on how to unzip compressed files in Linux refer to the unzip man pages.
d. Rsync is installed on the Rocky images by default, if you are curious check the version by entering **`rsync --version`** in the terminal.
e. To perform local backups the syntax of the **`rsync`** command is **`rsync {options} {source} {Destination}`**.
f. Back up the files in the original directory to the backup directory. Using the following command, where "abc1234," is your RIT login ID. This command assumes you are currently in your home directory.

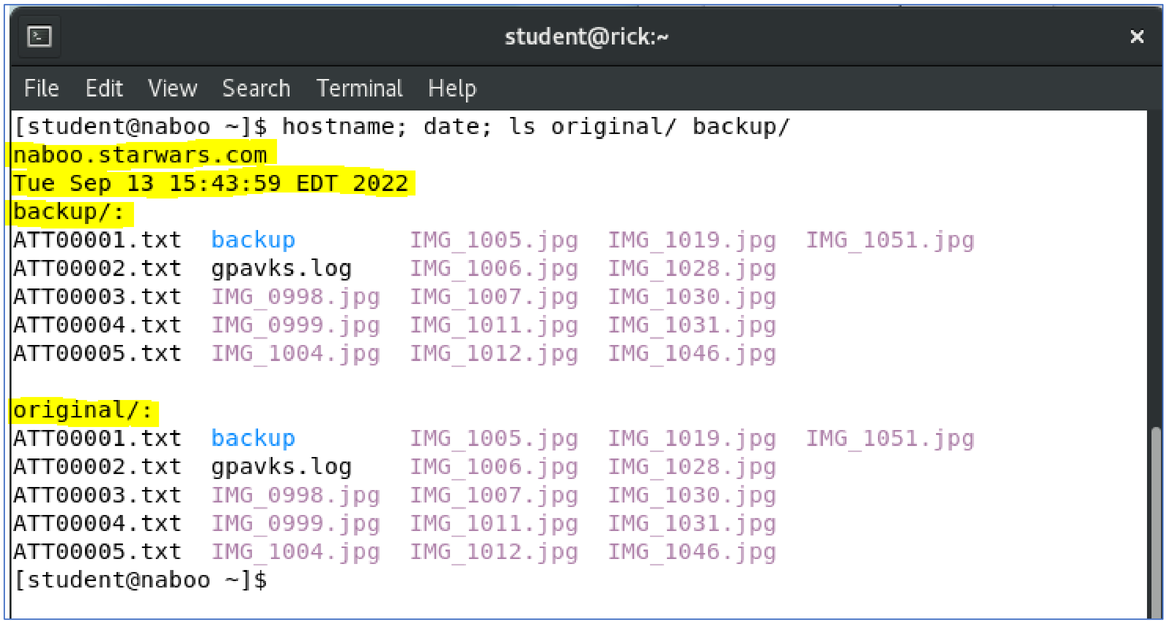
```
# rsync -av original/ backup --log-file=abc1234.log
```

g. Examine the contents of the log file. Look for information on the total bytes sent and received, and the total size of the data.

IMPORTANT NOTICE

For the report, include a screenshot showing the output of the **`hostname`**, **`date`**, **`ls original/ backup/`** commands.

Figure 8 – Example Screenshot



```
[student@naboo ~]$ hostname; date; ls original/ backup/
naboo.starwars.com
Tue Sep 13 15:43:59 EDT 2022
backup/:
ATT00001.txt  backup        IMG_1005.jpg  IMG_1019.jpg  IMG_1051.jpg
ATT00002.txt  gpavks.log    IMG_1006.jpg  IMG_1028.jpg
ATT00003.txt  IMG_0998.jpg  IMG_1007.jpg  IMG_1030.jpg
ATT00004.txt  IMG_0999.jpg  IMG_1011.jpg  IMG_1031.jpg
ATT00005.txt  IMG_1004.jpg  IMG_1012.jpg  IMG_1046.jpg

original/:
ATT00001.txt  backup        IMG_1005.jpg  IMG_1019.jpg  IMG_1051.jpg
ATT00002.txt  gpavks.log    IMG_1006.jpg  IMG_1028.jpg
ATT00003.txt  IMG_0998.jpg  IMG_1007.jpg  IMG_1030.jpg
ATT00004.txt  IMG_0999.jpg  IMG_1011.jpg  IMG_1031.jpg
ATT00005.txt  IMG_1004.jpg  IMG_1012.jpg  IMG_1046.jpg
[student@naboo ~]$
```

g. Delete the files in the backup directory and run the following command. For steps f through j after running each command examine the contents of the backup and destination directories and observe the behavior. **You will need to answer questions about your observations in the report.**

```
# rsync -av --exclude '*.jpg' original/ backup
```

h. Delete the current files in the backup directory and create some test files using the following command.

```
# touch test{1..9}.txt
```

i. Verify that the files have been created and examine the contents of the backup directory. Then run the following command.

```
# rsync -av --delete original/ backup
```

j. Run the same command again but omit the "/" after the "original" directory.

```
# rsync -av --delete original backup
```

k. In your home directory create another directory call “destination” and run the following command. Examine the contents of the ***destination*** directory. Was anything backed up?

```
# rsync --dry-run --remove-source-files -av original/ destination/
```

l. Run the command again without the **`--dry-run`** argument. Exam the contents of the “original” and “destination” directories and record what you observe.

```
# rsync --remove-source-files -av original/ destination/
```

IMPORTANT NOTICE

There is a question about **`rsync`** and the various arguments used in this activity, now would be a good time to answer it while the exercise is fresh in your mind.

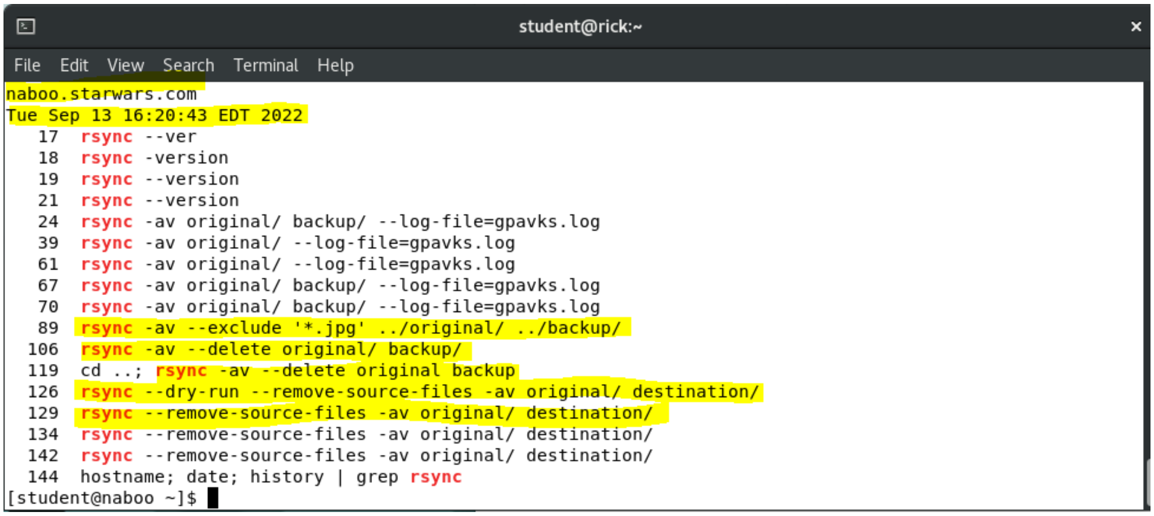
Explain in your own words what the **`rsync`** command and associative arguments did in this activity. What behavior did you observe?

IMPORTANT!

✓ For the Report

For the report, you will need to show the output of the **`hostname`**, **`date`**, and terminal **`history`** showing the **`rsync`** commands used in this activity. Refer to Figure 9 for reference.

Figure 9 – Sample Screen Shot of Rsync Commands



```
naboo.starwars.com
Tue Sep 13 16:20:43 EDT 2022
   17  rsync --ver
   18  rsync -version
   19  rsync --version
   21  rsync --version
   24  rsync -av original/ backup/ --log-file=gpavks.log
   39  rsync -av original/ --log-file=gpavks.log
   61  rsync -av original/ --log-file=gpavks.log
   67  rsync -av original/ backup/ --log-file=gpavks.log
   70  rsync -av original/ backup/ --log-file=gpavks.log
   89  rsync -av --exclude '*.jpg' ../original/ ../backup/
  106  rsync -av --delete original/ backup/
  119  cd ..; rsync -av --delete original backup
  126  rsync --dry-run --remove-source-files -av original/ destination/
  129  rsync --remove-source-files -av original/ destination/
  134  rsync --remove-source-files -av original/ destination/
  142  rsync --remove-source-files -av original/ destination/
  144  hostname; date; history | grep rsync
[student@naboo ~]$ 
```

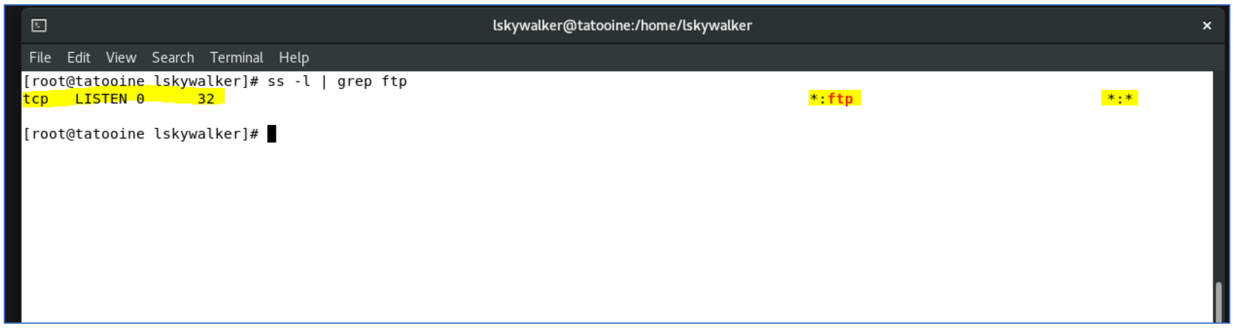
Activity 3 – Installing and Configuring FTP

For this activity, you will install the required packages for the Very Secure File Transport Protocol Service, or vsftpd. You should be familiar with the **dnf** command; if not, refer to the dnf cheat sheet in myCourses.

a. Make sure the system is updated! Always update!
b. Install the vsftpd package using **dnf** on the storage server created in Lab 4.
c. Once the installation is complete, enable the service to start when the system boots, and then start the service. Again, if you are not sure how to do this, refer to the systemd cheat sheet in myCourses.
d. Verify that the server is listening for FTP traffic by entering the following command. The output will be similar to Figure 10.

```
# ss -l | grep ftp
```

Figure 10 – Verification that the server is "Listening" FTP traffic


lskywalker@tatooine:/home/lskywalker
File Edit View Search Terminal Help

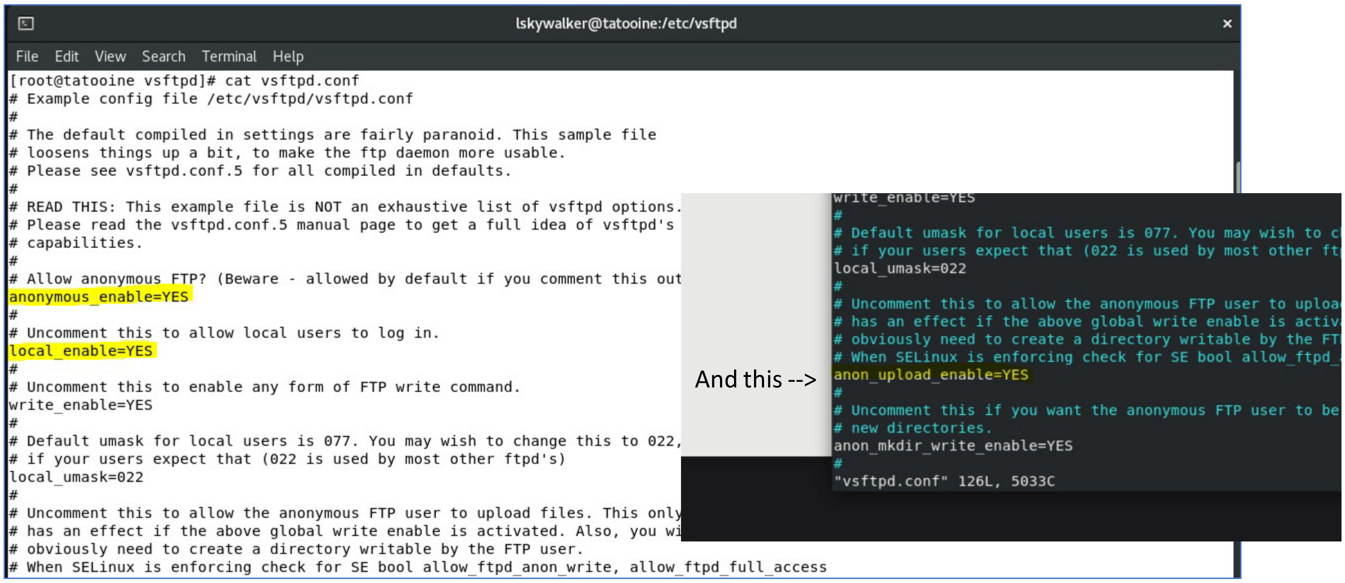
```
[root@tatooine lskywalker]# ss -l | grep ftp
tcp    LISTEN 0      32                                                                  *:ftp                                   *:*

[root@tatooine lskywalker]# 
```

e. Next, create the firewall rule to allow for incoming traffic on port 21 and FTP, then reload **`firewalld`**.

```
# firewall-cmd --permanent --add-port=21/tcp
# firewall-cmd --permanent --add-service=ftp
# firewall-cmd --reload
```

f. Examine the VSFTPD configuration file located in the ***/etc/vsftpd/*** directory. Take note of how, by default, the ***vsftpd.conf*** file allows local user and anonymous access (Figure 11). Examine so of the other default settings and read the comments in the file to understand what they do.

Figure 11 – Default Parameter Settings


lskywalker@tatooine:/etc/vsftpd
File Edit View Search Terminal Help

```
[root@tatooine vsftpd]# cat vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you wi
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_full_access
```

And this -->

```
write_enable=YES
#
# Default umask for local users is 077. You may wish to ch
# if your users expect that (022 is used by most other ft
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to uploa
# has an effect if the above global write enable is activ
# obviously need to create a directory writable by the FT
# When SELinux is enforcing check for SE bool allow_ftpd_
anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be
# new directories.
anon_mkdir_write_enable=YES
#
"vsftpd.conf" 126L, 5033C
```

g. Next, create a local user account (remember the **`adduser`** command?) that will be used to FTP from the client. Set a password for the account.

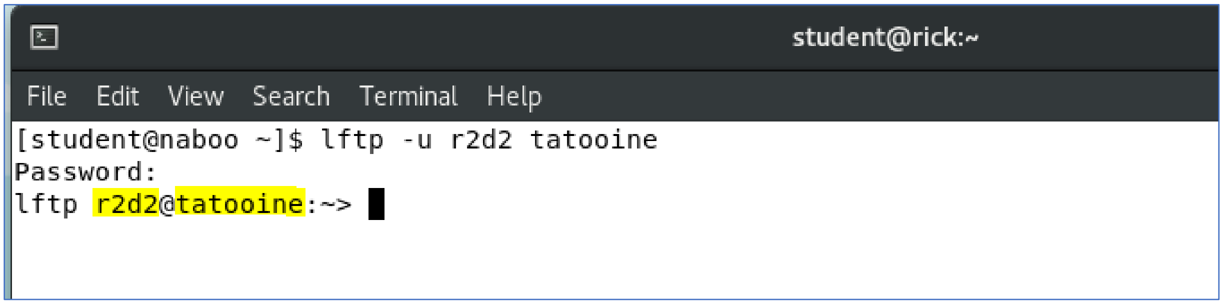
h. On the Linux client virtual machine, install the "Sophisticated File Transfer Program," or "lftp," using the dnf package manager. Did you update?

i. Once the user has been created, log into the server from the client using the following command. In the example command below the user logging in is, "r2d2" and the server hostname is "tatooine."

```
$ lftp -u r2d2 tatooine
```

j. When prompted, enter the password for the user. Once you have logged in successfully, the prompt will change from the shell prompt to the FTP prompt (Figure 12), showing the user logged in and the hostname. Type **`help`**, to see the many commands that FTP provides. To disconnect, type **`exit`**.

Figure 12 – FTP Prompt

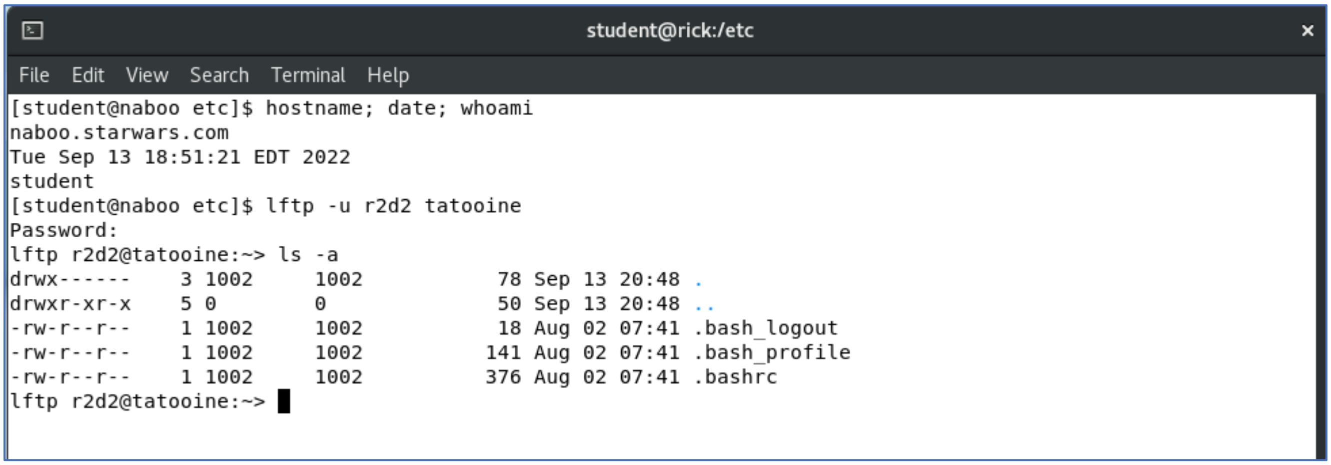


```
[student@naboo ~]$ lftp -u r2d2 tatooine
Password:
lftp r2d2@tatooine:~>
```

IMPORTANT NOTICE

For the report, include a single screenshot showing the output from the **`hostname`**, **`whoami`**, and **`date`** commands. Show the user you created logging into the FTP server from the client. Once you have logged in, enter the **`ls -a`** command. Refer to Figure 13 for an example.

Figure 13 – Successful FTP Login



```
[student@naboo etc]$ hostname; date; whoami
naboo.starwars.com
Tue Sep 13 18:51:21 EDT 2022
student
[student@naboo etc]$ lftp -u r2d2 tatooine
Password:
lftp r2d2@tatooine:~> ls -a
drwx------    3 1002     1002           78 Sep 13 20:48 .
drwxr-xr-x    5 0        0              50 Sep 13 20:48 ..
-rw-r--r--    1 1002     1002           18 Aug 02 07:41 .bash_logout
-rw-r--r--    1 1002     1002          141 Aug 02 07:41 .bash_profile
-rw-r--r--    1 1002     1002          376 Aug 02 07:41 .bashrc
lftp r2d2@tatooine:~>
```

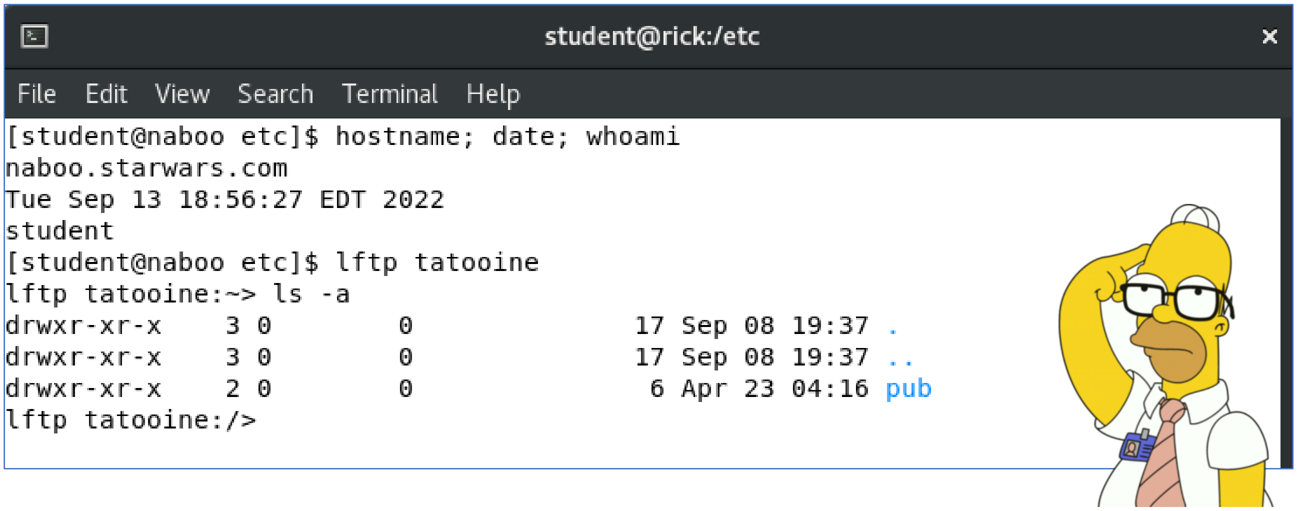
j. Next, log in anonymously by entering the server's hostname only.

```
$ lftp tatooine
```

IMPORTANT NOTICE

For the report, include a single screenshot showing the output from the **`hostname`**, **`whoami`**, and **`date`** commands. Show the anonymous login from the client to the server. Once you have logged in, enter the **`ls -a`** command. Refer to Figure 14 for an example.

Figure 14 – Anonymous FTP Login



```
[student@naboo etc]$ hostname; date; whoami
naboo.starwars.com
Tue Sep 13 18:56:27 EDT 2022
student
[student@naboo etc]$ lftp tatooine
lftp tatooine:~> ls -a
drwxr-xr-x    3 0        0              17 Sep 08 19:37 .
drwxr-xr-x    3 0        0              17 Sep 08 19:37 ..
drwxr-xr-x    2 0        0               6 Apr 23 04:16 pub
lftp tatooine:/>
```

Time to think. What are the differences between a local user logging into the server and someone logging in anonymously?

Activity 4 – Creating a DropBox

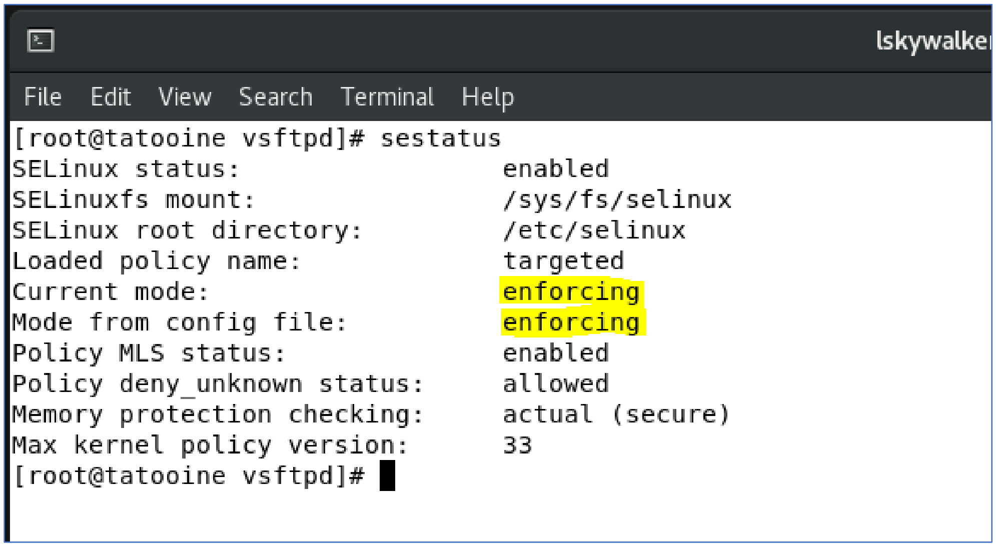
This activity will walk you through the process of configuring an FTP anonymous user drop box and give you some idea of how the drop boxes work in myCourses. As a student, you can upload files to the myCourses drop boxes, but once they have been uploaded, you no longer have the ability to download them. Additionally, this activity will also provide a brief introduction to SELinux. For this to work, SELinux will need to be set to enforcing.

a. Install the SELinux Troubleshooter **on the server** using the following command.

```
# dnf install -y setroubleshoot setools
```

b. On the server edit the /etc/selinux/config file, so it is enforcing, or you may temporarily set it to enforcing by entering the command **`setenforce 1`**. You can set it to permissive by entering the command **`setenforce 0`**. If you edit the configuration file you will need to **reboot** the system for the changes to take effect. To check the current status, use the **getenforce**, **sestatus** commands. For this activity SELinux needs to be "**Enforcing,**" on the server (Figure 15). For later labs you may want to disable it, or set it to permissive.

Figure 15 – Sestatus Output



```
[root@tatooine vsftpd]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[root@tatooine vsftpd]# 
```

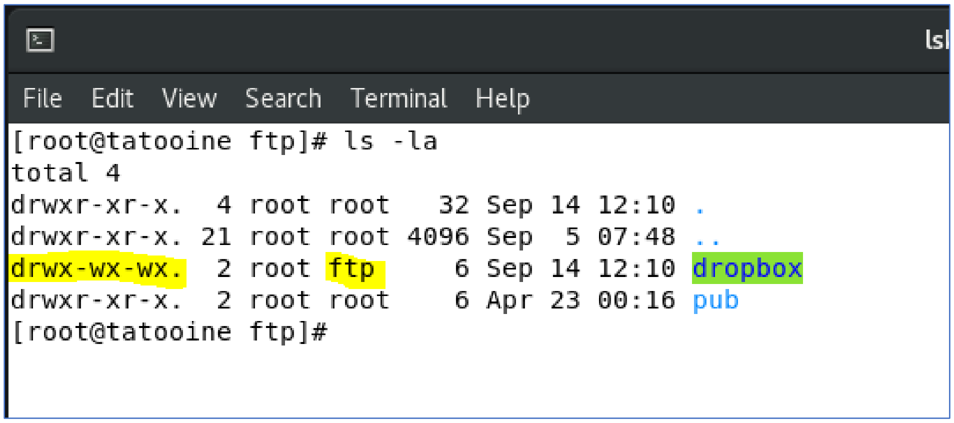
c. Create a directory in the FTP default document root, /var/ftp, called "dropbox".

d. Set the permissions, so that the owner can read, write, and execute. Change permissions so that the group and others have write and execute permissions. The octal values for the directory will be **0733**. For the drop box we want the anonymous user to be able to write to it but not read from it.

e. Next, create an **ftp** group for the directory by entering the following command. The permissions should look like the permissions in Figure 16, for the directory. Notice that everyone has write permissions.

```
# chgrp ftp /var/ftp/dropbox
```

Figure 16 – Directory Permissions



```
[root@tatooine ftp]# ls -la
total 4
drwxr-xr-x.  4 root root   32 Sep 14 12:10 .
drwxr-xr-x. 21 root root 4096 Sep  5 07:48 ..
drwx-wx-wx.  2 root ftp     6 Sep 14 12:10 dropbox
drwxr-xr-x.  2 root root    6 Apr 23 00:16 pub
[root@tatooine ftp]#
```

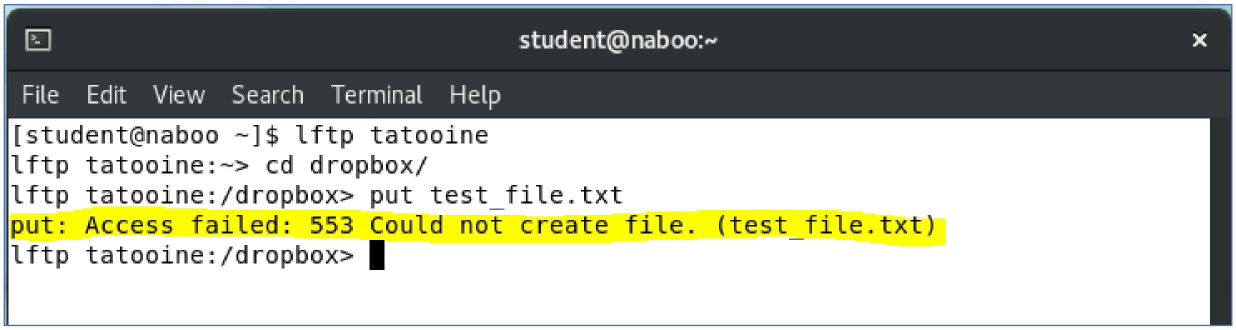
f. Edit the vsftpd configuration file so that the following statements are created or uncommented.

```
anon_upload_enable=yes
anon_mkdir_write_enable=yes
chown_uploads = yes
chown_username = root
```

g. Restart the vsftpd service.

h. On the client create a file to transfer to the server. Anonymously log into the server and navigate to the dropbox directory. Use the **put** command to transfer the file to the dropbox directory. It will fail, see Figure 17. This is because we have not created the correct SELinux context label for the "dropbox" directory.

Figure 17 – Transfer Fail

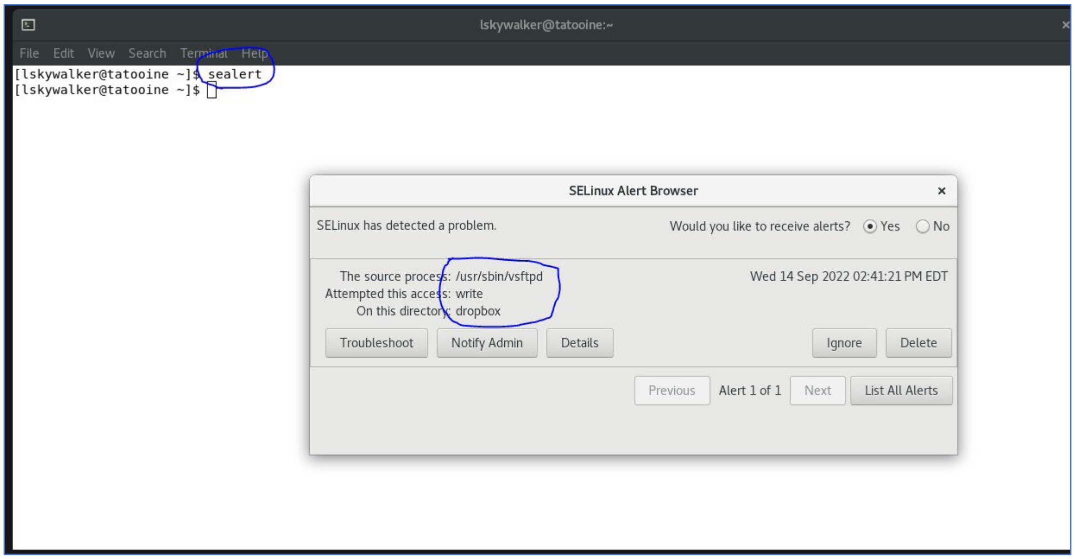


```
[student@naboo ~]$ lftp tatooine
lftp tatooine:~> cd dropbox/
lftp tatooine:/dropbox> put test_file.txt
put: Access failed: 553 Could not create file. (test_file.txt)
lftp tatooine:/dropbox>
```

```
# dnf install -y setroubleshoot setools
```

i. How do we know the "dropbox" directory does not have the correct SELinux context label? Open the SELinux Alert Browser by typing **sealert** in a terminal. The message displayed will be similar to Figure 18. Examining the alert gives us some idea of what is going on, the vsftpd process, attempted to write to the "dropbox" directory and was denied.

Figure 18 – SELinux Alert Message



```
[lskywalker@tatooine ~]$ sealert
[lskywalker@tatooine ~]$
```

SELinux has detected a problem.

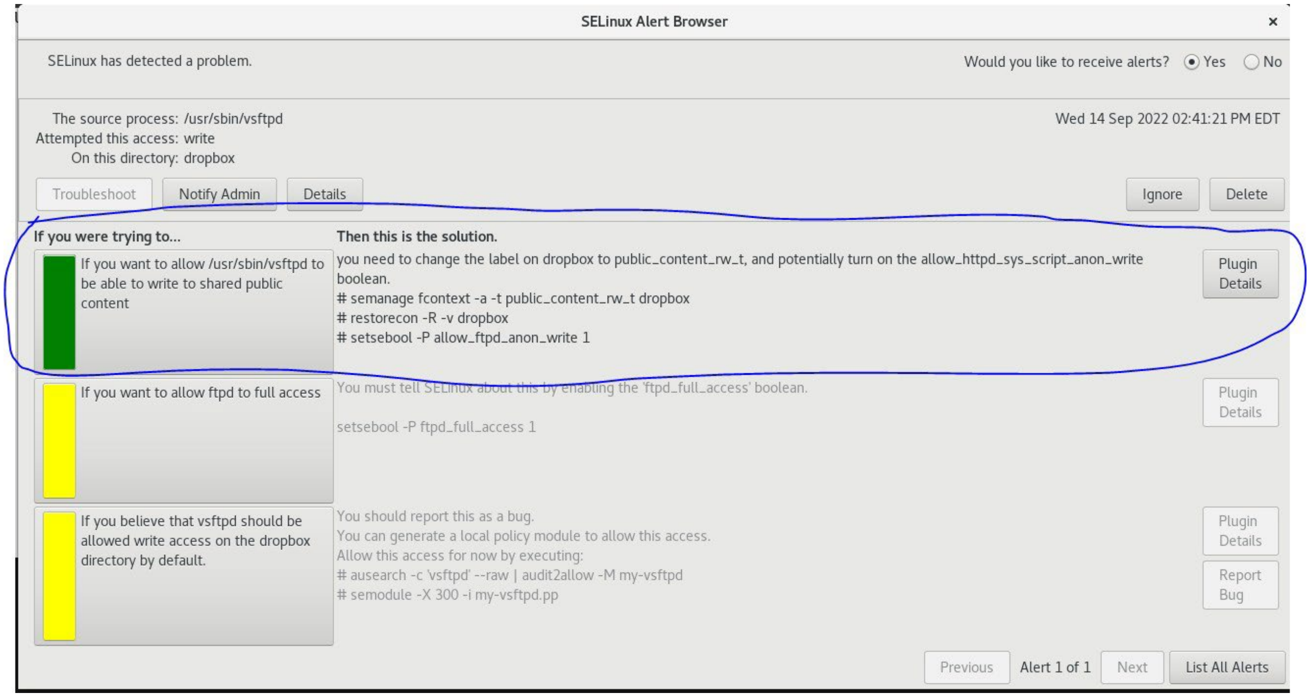
Would you like to receive alerts? ● Yes ○ No

The source process: /usr/sbin/vsftpd
Attempted this access: write
On this directory: dropbox

Wed 14 Sep 2022 02:41:21 PM EDT

j. For more information, you can click the "Troubleshoot" button, and are given several options to address the problem (Figure 19). What we are trying to do is allow the vsftpd service to write to the "dropbox" directory. The troubleshooter provides us with a solution and the commands to use.

Figure 19 – SELinux Troubleshooting Solution



SELinux has detected a problem.

Would you like to receive alerts? ● Yes ○ No

The source process: /usr/sbin/vsftpd
Attempted this access: write
On this directory: dropbox

Wed 14 Sep 2022 02:41:21 PM EDT

If you were trying to...	Then this is the solution.
🟩 If you want to allow /usr/sbin/vsftpd to be able to write to shared public content	you need to change the label on dropbox to public_content_rw_t, and potentially turn on the allow_httpd_sys_script_anon_write boolean. # semanage fcontext -a -t public_content_rw_t dropbox # restorecon -R -v dropbox # setsebool -P allow_ftpd_anon_write 1
🟨 If you want to allow ftpd to full access	You must tell SELinux about this by enabling the 'ftpd_full_access' boolean. setsebool -P ftpd_full_access 1
🟨 If you believe that vsftpd should be allowed write access on the dropbox directory by default.	You should report this as a bug. You can generate a local policy module to allow this access. Allow this access for now by executing: # ausearch -c 'vsftpd' --raw | audit2allow -M my-vsftpd # semodule -X 300 -i my-vsftpd.pp

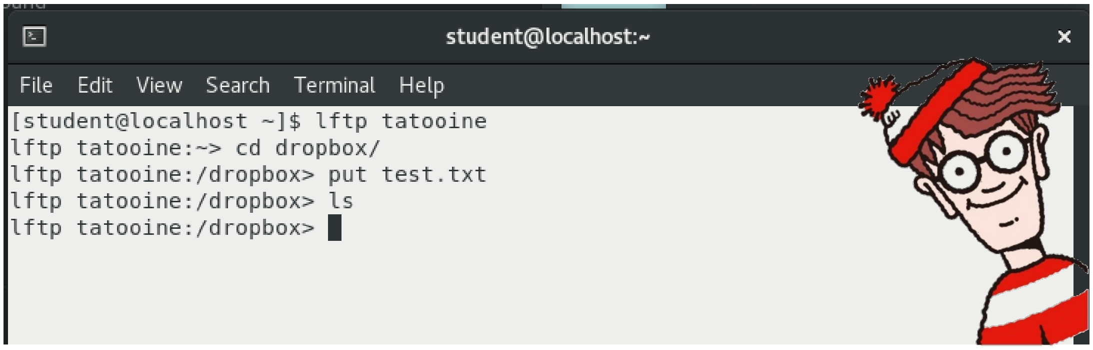
Note: the ↲ symbol denotes that the next line of text is a continuation of the current command; meaning the first two lines are entered on a single terminal line.

```
# semanage fcontext –a –t public_content_rw_t ↲
  “/var/ftp/dropbox(/.*)?”
# restorecon –R –v /var/ftp/dropbox
# setsebool -P ftpd_anon_write 1
```

e. Now that the “dropbox” directory has the correct SELinux context rules and Boolean value for anonymous write access, you can upload the file to the directory.

f. However! If you type the `ls` command from the client you will not see the file (Figure 20).

Figure 20 – Where’s ~~Waldo~~ the File!



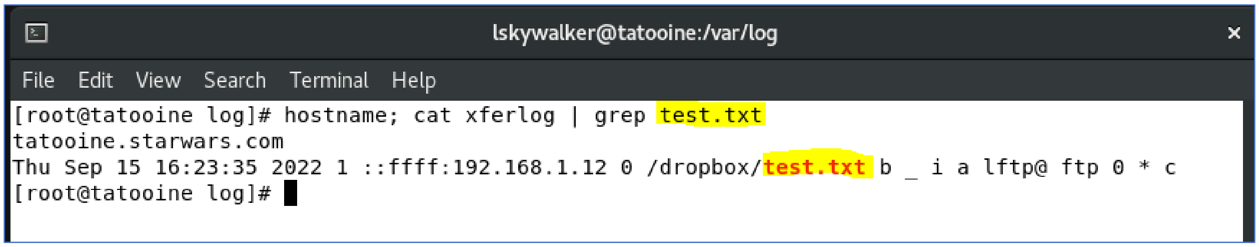
```
[student@localhost ~]$ lftp tatooine
lftp tatooine:~> cd dropbox/
lftp tatooine:/dropbox> put test.txt
lftp tatooine:/dropbox> ls
lftp tatooine:/dropbox> █
```

g. Check the server to see if the file is there. Or better yet, check the log located in /var/log/xferlog.

IMPORTANT NOTICE

For the report, include a single screenshot showing the output from the **hostname** command and the contents of the transfer log (/var/log/xferlog) by “grepping” for the file. Refer to Figure 21 for an example.

Figure 21 – Sample Output of the xferlog File



lskywalker@tatooine:/var/log
File Edit View Search Terminal Help

```
[root@tatooine log]# hostname; cat xferlog | grep test.txt
tatooine.starwars.com
Thu Sep 15 16:23:35 2022 1 ::ffff:192.168.1.12 0 /dropbox/test.txt b _ i a lftp@ ftp 0 * c
[root@tatooine log]# 
```

Activity 5 – Installing Samba and Creating a Samba Share

For this activity, you will install the required packages to set up a Samba server. Once the service is installed it will be configured to restrict user access based on group membership.

a. There are three packages required for Samba, check to see if they are installed on the server. Hint: Check the **dnf** cheat sheet posted to myCourses. To see what the required packages are for your Linux distribution check the Samba documentation.

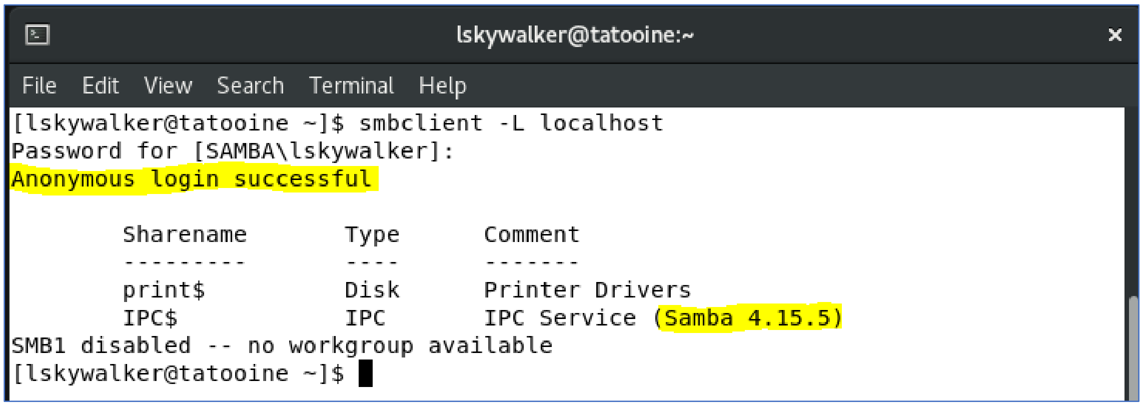
   https://wiki.samba.org/index.php/Main_Page

b. Next, configure the smb and nmb services to start and enable for system reboots. Remember the **systemctl** command?

c. Enter the following command to test local connectivity to the service. The output will be similar to Figure 22. Note the Samba version.

```
# smbclient -L localhost
```

Figure 22 – Samba Version Information

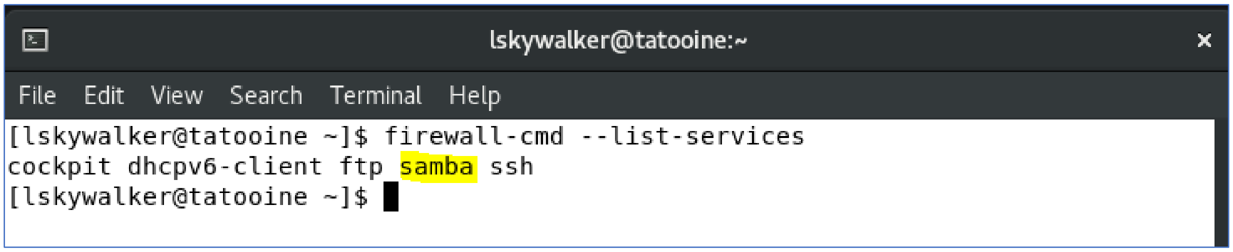


```
[lskywalker@tatooine ~]$ smbclient -L localhost
Password for [SAMBA\lskywalker]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.15.5)
SMB1 disabled -- no workgroup available
[lskywalker@tatooine ~]$ 
```

c. Before going further, we need to set the firewall rules on the server to allow clients access to the Samba shares. At this point you should be comfortable with the **firewall-cmd** command, so the specific command will not be provided. However, the ports that Samba uses are TCP port 445, and 139. Or you can add the samba service. To verify that the firewall is configured correctly enter the follow command and reload the firewall. Afterwards you will see the Samba service listed (Figure 23).

```
# firewall-cmd –list-services
```

Figure 23 – Firewall Services



```
[lskywalker@tatooine ~]$ firewall-cmd --list-services
cockpit dhcpv6-client ftp samba ssh
[lskywalker@tatooine ~]$ 
```

d. Next, we'll create Samba users on the server. Samba users are associated with local Linux user accounts, but they are given a Samba specific **password**. On the server I created the four users: joey, johnny, deedee, and marky, using the **useradd** command. I also want to prevent them from logging in remotely and using other services, like SSH, do to this I am setting their shell to /sbin/nologin.

```
# for i in joey johnny deedee marky; do useradd -s /sbin/nologin $i; done
```

e. Create a group whose member will have write access to the Samba share. The following command adds the "writers" group.

```
# groupadd writers
```

f. Use the **usermod** command to add joey, johnny, and deedee to the group. Unfortunately, the Ramones change drummers often, so we don't them involved in writing songs, so Marky is not added to the "writers" group.

```
# for i in joey johnny deedee; do usermod -aG writers $i; done
```

g. Verify that Joey, Johnny, and Deedee are members of the "writers" group, by viewing the /etc/group file. Or to find out if a specific user is a member of a group enter the following command, where "joey" is the example user.

```
# groups joey
```

h. Create Samba specific passwords for the users. Since this a lab and not the "real world," keep it simple, like "password." To do this, use the **smbpasswd -a** command, it is important to add the "a" argument, because even though the local user accounts have been created, these are "adding" passwords for the Samba user accounts.

```
# for i in joey johnny deedee marky; do smbpasswd -a $i; done
```

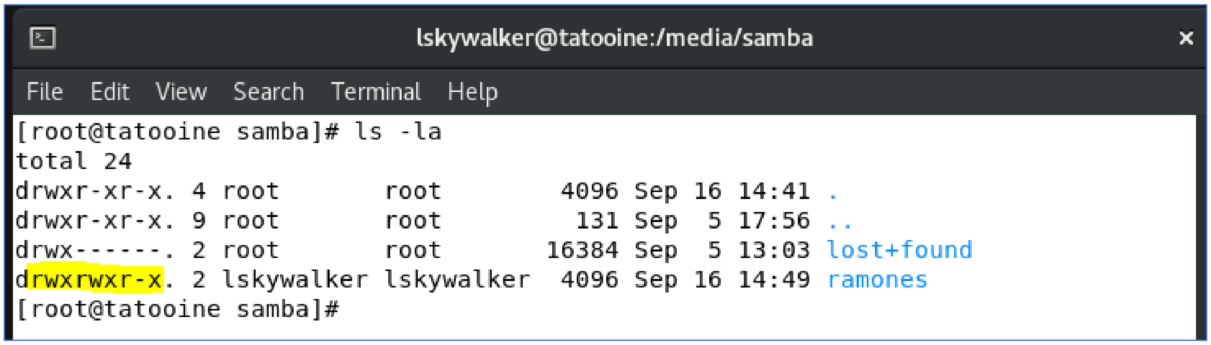
i. When prompted enter the password twice for each user.
j. Create the ramones directory on the */media/samba* partition created in Lab 4.
k. Set the group ownership of the directory to the writers group.

```
# chgrp writers /media/samba/ramones
```

l. Change the permissions so that the "*writers*" group has write access.

```
# chmod g+rwx /media/samba/ramones
```

Figure 24 – Example Permissions and Group membership for Samba Share


lskywalker@tatooine:/media/samba
File Edit View Search Terminal Help

```
[root@tatooine samba]# ls -la
total 24
drwxr-xr-x. 4 root       root        4096 Sep 16 14:41 .
drwxr-xr-x. 9 root       root         131 Sep  5 17:56 ..
drwx------. 2 root       root       16384 Sep  5 13:03 lost+found
drwxrwxr-x. 2 lskywalker lskywalker  4096 Sep 16 14:49 ramones
[root@tatooine samba]#
```

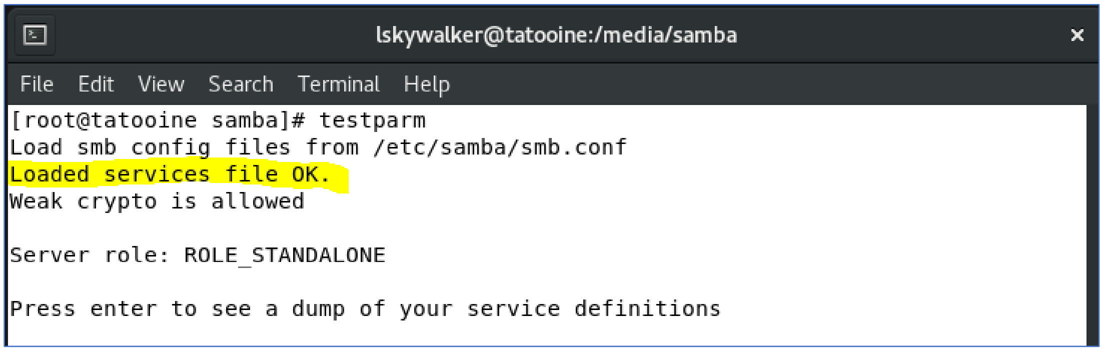
m. Add some files to the directory. I added two, the "*lyrics*" and "*chords*" text files for demonstration purposes.
n. Finally, edit the */etc/samba/smb.conf* file for the share directory by adding the following to the bottom of the file.

```
[ramones]
comment = Blitzkrieg Bop
path = /media/samba/ramones
read only = no
write list = @writers
```

o. Use the **testparm** command, to check for any syntactical errors in the *smb.conf* file. The output will show the share and indicate that the "*Loaded services file*," is "*OK*," see Figure 25.

```
$ testparm
```

Figure 25 – Sample Output from the testparm Command

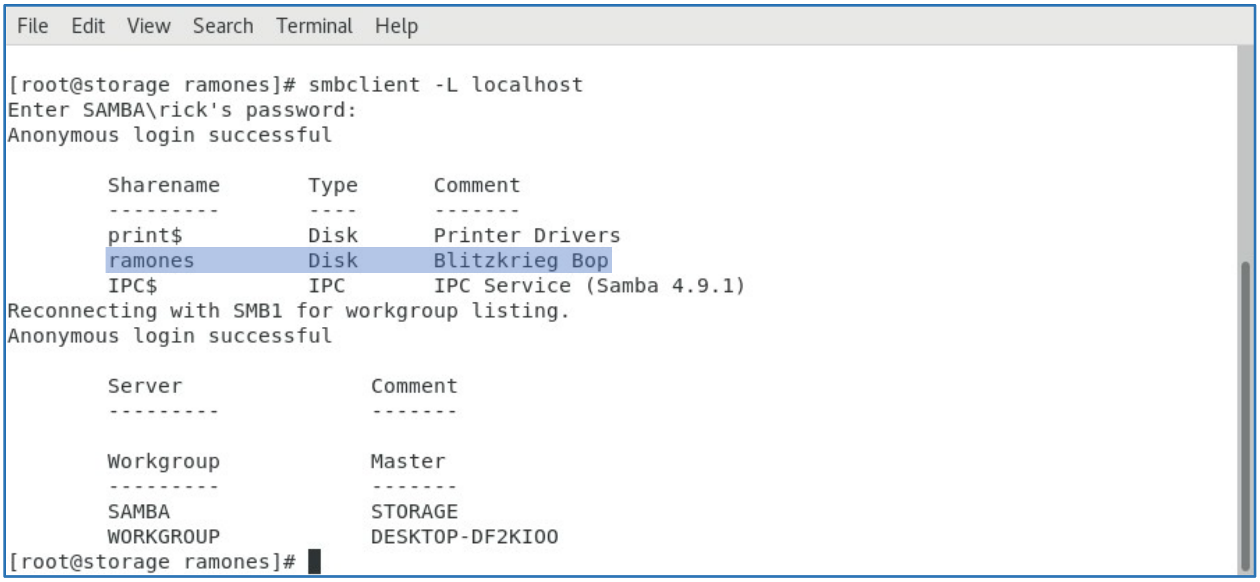

lskywalker@tatooine:/media/samba
×
File Edit View Search Terminal Help

```
[root@tatooine samba]# testparm
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed

Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions
```

p. Restart the nmb and smb.services.

q. Next, on the server use the `smbclient -L` command to verify the share exists (Figure 18).

Figure 25 – Sample Output from the `smbclient` Command



```
[root@storage ramones]# smbclient -L localhost
Enter SAMBA\rick's password:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        ramones         Disk      Blitzkrieg Bop
        IPC$            IPC       IPC Service (Samba 4.9.1)
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        SAMBA                STORAGE
        WORKGROUP            DESKTOP-DF2KIOO
[root@storage ramones]#
```

r. Check the status of SELinux, use the `setenforce 0` command to temporarily disable it, or edit the /etc/selinux/config file to disable on boot.

s. On the client virtual machine, install the samba-client and cifs-utils packages.

t. Use the smbclient command to confirm the share can be accessed remotely, substituting "localhost" with the hostname, of the device where the samba share is located. In the example command, the hostname of the server is "storage," the hostname of your server maybe different. The output will be similar to that of the server (Figure 18). If you run into problems, double check the firewall and SELinux settings.

Note: Depending on the user you are currently logged in as you may be prompted for a password, otherwise just hit enter.

```
$ smbclient –L //storage/
```

u. Log in as one of the users that is a member of the writers group. In the following example, I am logging in as the user Johnny to the Samba share, “ramones.”

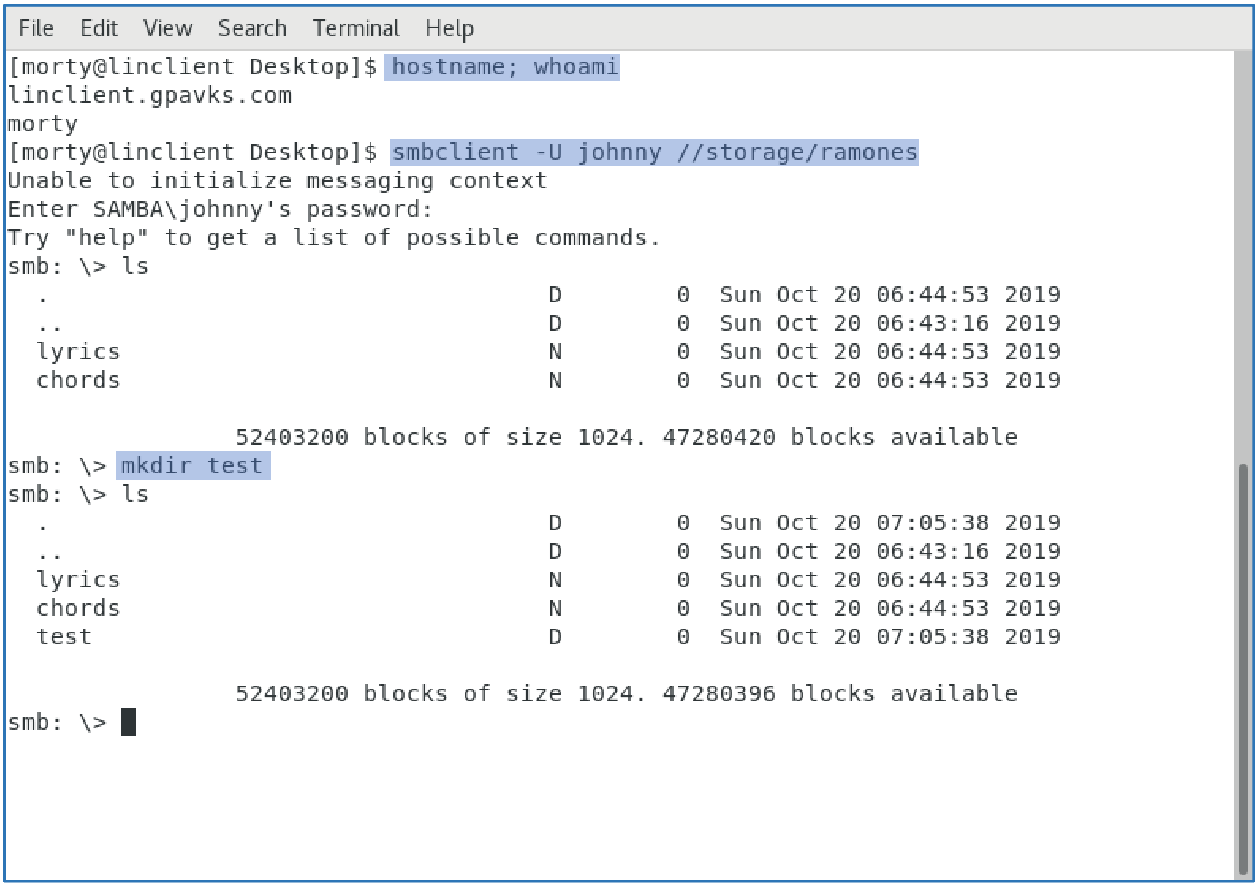
```
$ smbclient –U johnny //storage/ramones
```

v. When prompted for the user’s password make sure to enter the Samba password and not the local Linux user account password (they are different). Once, you have successfully logged in, use the ls command to view the contents of the directory. To demonstrate you have write access, use the mkdir command and create a “test” directory.

IMPORTANT NOTICE

For the report, include a single screenshot showing the output from the, `hostname` and `whoami` commands. You will also need to show the Samba user successfully logging into the shared directory. Once you are able to access the share use the `mkdir` command to create a “test” directory and then use the `ls` command to list it. Refer to Figure 19 for an example.

Figure 19 – Sample Screenshot for Remote Samba Access and Write Verification



```
[morty@linclient Desktop]$ hostname; whoami
linclient.gpavks.com
morty
[morty@linclient Desktop]$ smbclient -U johnny //storage/ramones
Unable to initialize messaging context
Enter SAMBA\johnny's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Oct 20 06:44:53 2019
  ..                                  D        0  Sun Oct 20 06:43:16 2019
  lyrics                              N        0  Sun Oct 20 06:44:53 2019
  chords                              N        0  Sun Oct 20 06:44:53 2019

                52403200 blocks of size 1024. 47280420 blocks available
smb: \> mkdir test
smb: \> ls
  .                                   D        0  Sun Oct 20 07:05:38 2019
  ..                                  D        0  Sun Oct 20 06:43:16 2019
  lyrics                              N        0  Sun Oct 20 06:44:53 2019
  chords                              N        0  Sun Oct 20 06:44:53 2019
  test                                D        0  Sun Oct 20 07:05:38 2019

                52403200 blocks of size 1024. 47280396 blocks available
smb: \> 
```

Activity 6 – Accessing the Samba Share from Windows

For this activity, you will access the share from Windows using the user that does not have write permissions. If you have been using the user that was created in the previous activity the user will be “marky.” Also, you’ll access the share using PowerShell CLI and graphical utilities.

a. On the windows client, open PowerShell as Administrator.

b. Establish an SMB connection with a user who is a member of the “writers” group by entering the following command. The example command uses the user “joey,” whose password is “password” your may be different.

```
> New-SmbMapping -LocalPath ‘Z:’ -RemotePath ‘\\storage\ramones’ -UserName ↵
‘joey’ -Password ‘password’
```

c. Use the cd command to navigate to the “ramones” directory.

```
> cd \\storage\ramones\
```

d. To test that the user can write to the directory use the `mkdir` or `touch` commands and create another directory or file.

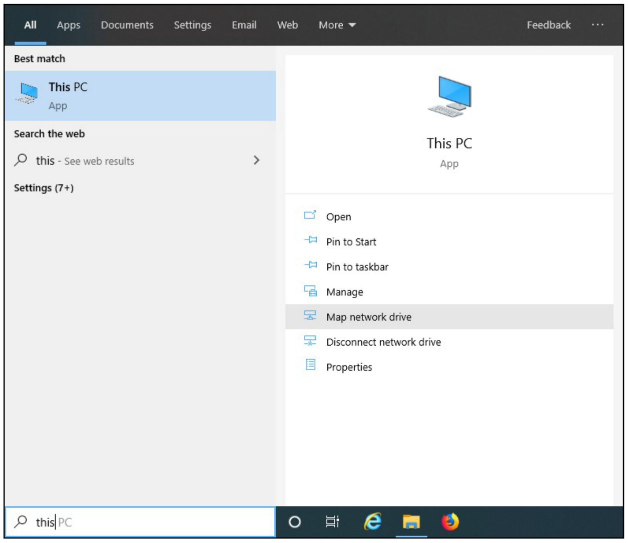
Figure 20 – PowerShell Write Test

e. To terminate the current connection type the following command. When prompted hit enter, the default is Yes.

```
> Remove-SmbMapping -RemotePath ‘\\storage\ramones’
```

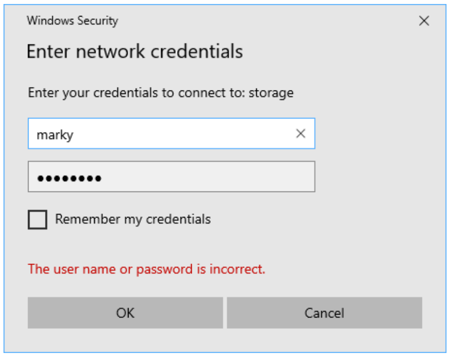
f. Next establish an SMB connection by mapping the drive (figure 21).

Figure 21 – Mapping to the SMB Share



g. Select a user that does not have “write” permission, continuing with the previous examples I’ll stick with “marky” and enter their credentials (Figure 22).

Figure 22 – Enter SMB User Credentials



Windows Security

Enter network credentials

Enter your credentials to connect to: storage

marky

••••••••

□ Remember my credentials

The user name or password is incorrect.

h. Once you have logged in you will see two directories, a home directory for the user and the “ramones” directory (Figure 23). Double-click the “ramones” directory and try adding a file or directory to it, as expected you will be told “You need permission to perform this action” (Figure 24).

Figure 23 – SMB Shares

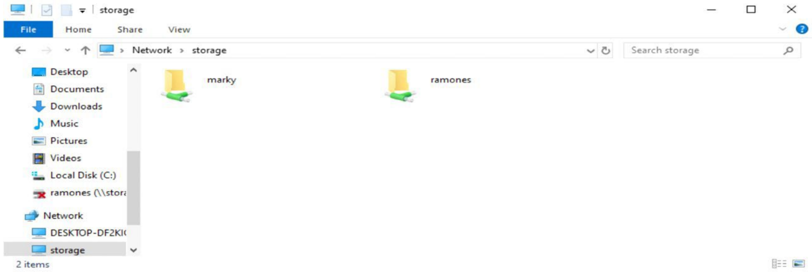
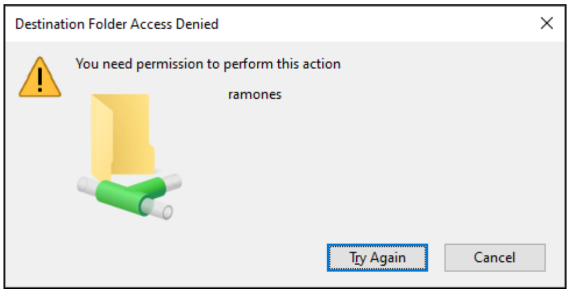


Figure 24 – Permission Denied



Destination Folder Access Denied

You need permission to perform this action

ramones

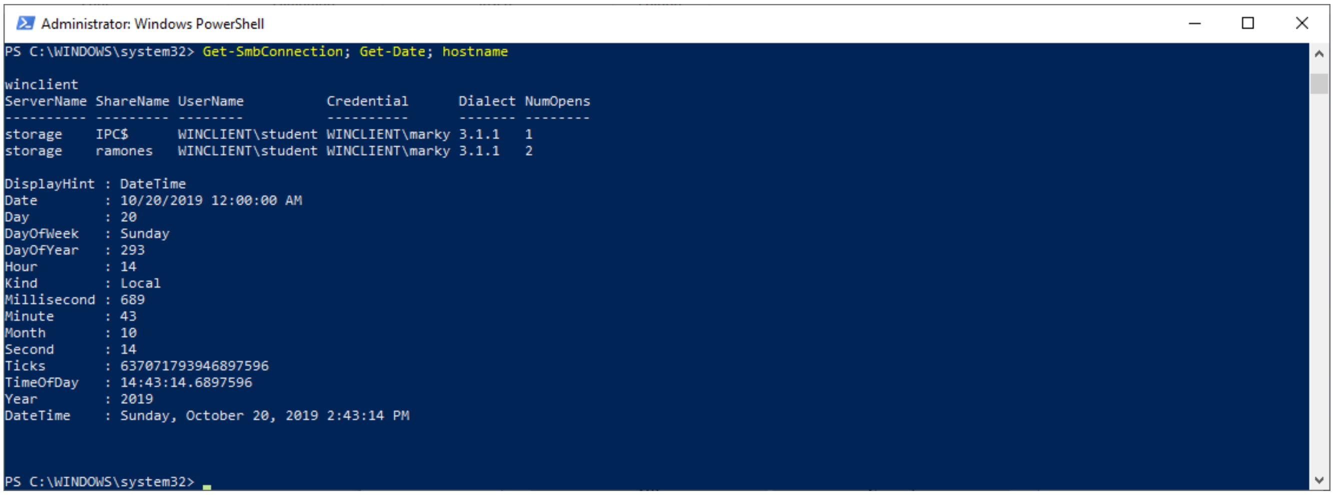
i. To terminate the SMB connection and log in as a different user enter the following command in PowerShell. But wait! Before you disconnect you need to get some screenshots.

```
> Remove-SmbMapping -RemotePath '\\storage\ramones'
```

IMPORTANT NOTICE

For the report, obtain a screenshot showing the output of the `Get-SmbConnection`, `Get-date`, and `hostname` cmdlets. Refer to Figure 25 for an example.

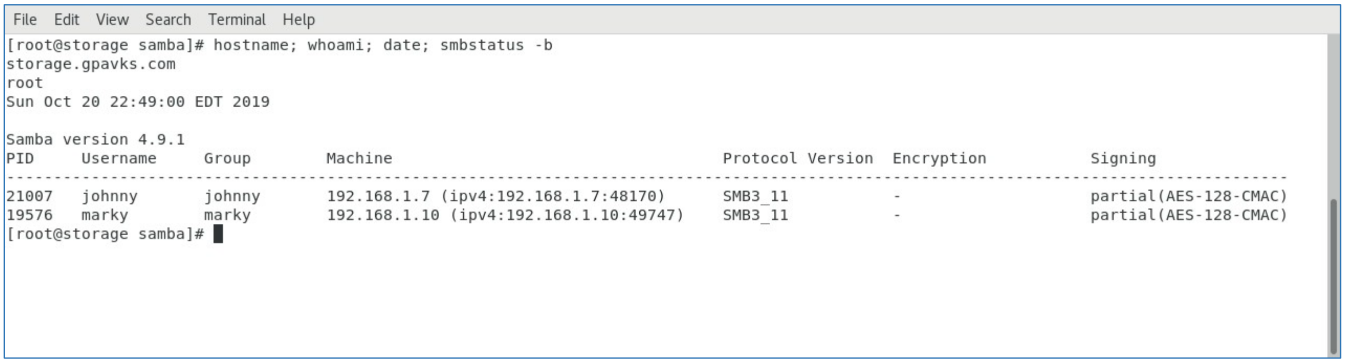
Figure 25 – Windows Client SMB Verification



```
PS C:\WINDOWS\system32> Get-SmbConnection; Get-Date; hostname

winclient
ServerName ShareName UserName           Credential        Dialect NumOpens
---------- --------- --------           ----------        ------- --------
storage    IPC$      WINCLIENT\student WINCLIENT\marky 3.1.1   1
storage    ramones   WINCLIENT\student WINCLIENT\marky 3.1.1   2

DisplayHint : DateTime
Date        : 10/20/2019 12:00:00 AM
Day         : 20
DayOfWeek   : Sunday
DayOfYear   : 293
Hour        : 14
Kind        : Local
Millisecond : 689
Minute      : 43
Month       : 10
Second      : 14
Ticks       : 637071793946897596
TimeOfDay   : 14:43:14.6897596
Year        : 2019
DateTime    : Sunday, October 20, 2019 2:43:14 PM


PS C:\WINDOWS\system32>
```

Include a second screenshot from the server showing the output from the `hostname`,`whoami`, and `date` commands. Include the output of the smbstatus –b command showing connections from the Windows and Linux clients and two different users. Refer to Figure 26 for an example.

Figure 26 – Sample Output from the `smbstatus –b` Command



```
[root@storage samba]# hostname; whoami; date; smbstatus -b
storage.gpavks.com
root
Sun Oct 20 22:49:00 EDT 2019

Samba version 4.9.1
PID     Username     Group        Machine                                   Protocol Version  Encryption           Signing
----------------------------------------------------------------------------------------------------------------------------------------
21007   johnny       johnny       192.168.1.7 (ipv4:192.168.1.7:48170)      SMB3_11           -                    partial(AES-128-CMAC)
19576   marky        marky        192.168.1.10 (ipv4:192.168.1.10:49747)    SMB3_11           -                    partial(AES-128-CMAC)
[root@storage samba]# █
```

Activity 7 – Creating an RSYNC Module

For this activity, you will configure the storage server to as an RSYNC server and use the client to transfer data from it. For this activity, use on of the existing mounts from the previous lab, or create a new directory in /media. The example in the instructions uses the "rsync" subdirectory.

a. Verify that the rsync packages are installed on the server.
b. Add the necessary firewall rules to allow the RSYNC server. RSYNC runs on TCP port 873.
c. RSYNC servers export modules as Samba servers export shares. Edit the rsyncd configuration file (/etc/rsyncd.conf) and append the following lines:

```
[ramones]
chroot = false
path = /media/rsync
comment = Ramones RSYNC Module
read only = yes
list = yes
uid = nobody
gid = nobody
```

Configuration Summary

[ramones] specifies the name of the module.

chroot = false prevents the rsync daemon from changing the root path before starting the file transfer with the client. If set to true you will need root privileges to access the path.

path = /media/rsync specifies the path of the directory to export as the module. In this case we are using the rsync1 logical volume we created in lab 3.

comment = Ramones RSYNC Module provides a user friendly comment to describe the module.

read only = yes configures the module to be read only. While RSYNC can be used to both upload and download files, it is usually used only to download files (similar to how FTP servers frequently allow only anonymous downloads).

list = yes allows clients to list the directory contents in the module.

d. Restart the rsyncd service.

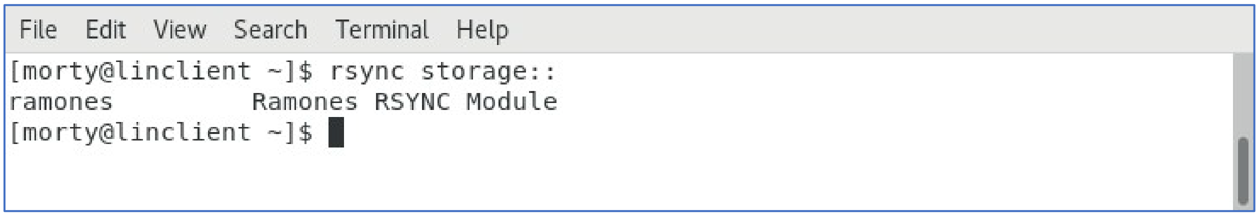
e. Create a file in /media/rsync and add content to it (i.e. a text file).

f. On the client, verify that the rsync package is installed.

g. Run the following command on the client to list the available RSYNC modules on the storage server. You should see output similar to Figure 27.

```
$ rsync storage::
```

Figure 27 – Available RSYNC Modules

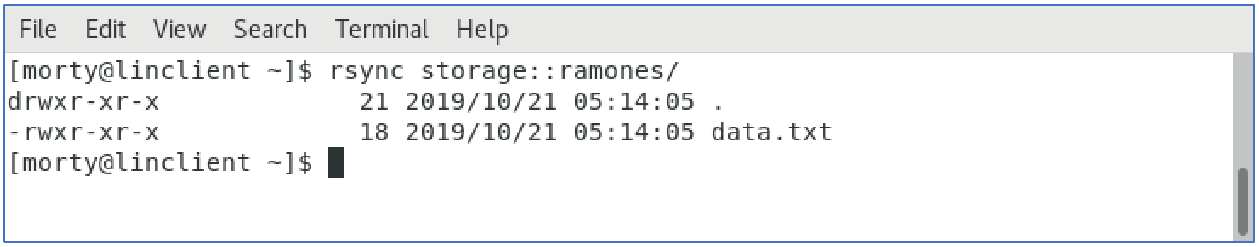


```
[morty@linclient ~]$ rsync storage::
ramones         Ramones RSYNC Module
[morty@linclient ~]$ 
```

h. From the client, list the contents of the of the ramones module's directory. You should see output similar to Figure 28.

```
$ rsync storage::ramones/
```

Figure 28 – RSYNC Directory Listing



```
[morty@linclient ~]$ rsync storage::ramones/
drwxr-xr-x             21 2019/10/21 05:14:05 .
-rwxr-xr-x             18 2019/10/21 05:14:05 data.txt
[morty@linclient ~]$ 
```

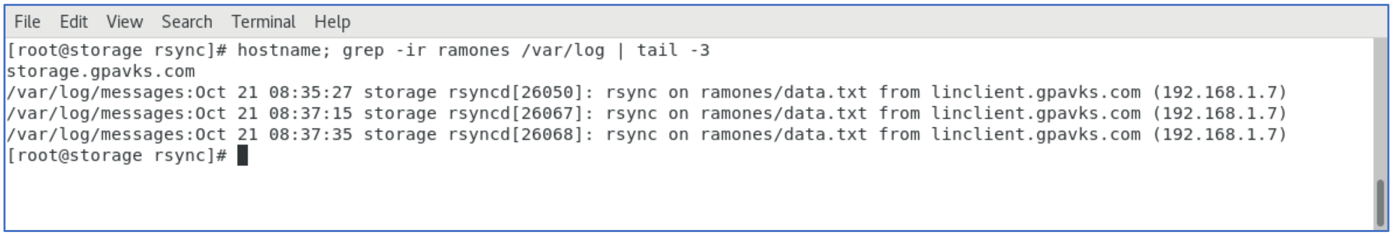
i. Transfer the test.txt file over to the client using the following command. Unless there is an error, you will notice that there is no output from the command. When using RSYNC, no news is good news.

```
$ rsync storage::ramones/test.txt ./
```

IMPORTANT NOTICE

For the report, provide a screenshot showing the output from the `hostname` command and rsync log messages by grepping /var/log for rsync log messages. The suggested command to do this is `grep -ir ramones /var/log | tail -3`. Figure 29 provides an example of the output.

Figure 29 - Server Log



```
[root@storage rsync]# hostname; grep -ir ramones /var/log | tail -3
storage.gpavks.com
/var/log/messages:Oct 21 08:35:27 storage rsyncd[26050]: rsync on ramones/data.txt from linclient.gpavks.com (192.168.1.7)
/var/log/messages:Oct 21 08:37:15 storage rsyncd[26067]: rsync on ramones/data.txt from linclient.gpavks.com (192.168.1.7)
/var/log/messages:Oct 21 08:37:35 storage rsyncd[26068]: rsync on ramones/data.txt from linclient.gpavks.com (192.168.1.7)
[root@storage rsync]#
```

Activity 8 – Creating an NFS Exports

For this activity, you will configure the client and server to use the Network File System or NFS. NFS was native to Unix/Linux environments and was developed by Sun Microsystems in 1984. As old as it is, it is still used and being developed today, mainly because it requires little overhead and it's easy to use. In industry, the two most common deployments are to provide access to home directories for LDAP users and access to shared file systems on other Linux servers. Additionally, NFS is supported by Windows Server 2016 and 2019. For this activity you will mount to NFS exports from Windows and Linux clients.

a. Install the nfs-utils packages on the storage server.

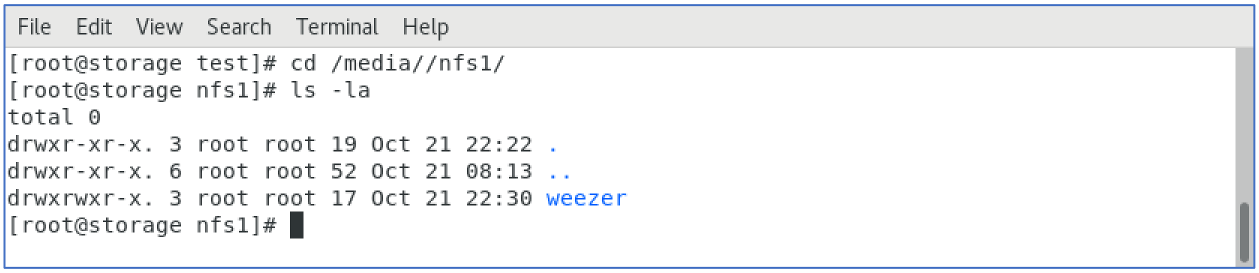
```
$ dnf install nfs-utils
```

b. Create the directory to be "exported," or shared on `/media/nfs1`. For example, I created the `/media/nfs1/weezer` directory (which is located on the partition md0p1 that was created in Lab 3).

c. Change the permission and ownership for the directory to match those in Figure 30.

```
$ chmod 775 /media/nfs1/weezer
```

Figure 30 – Directory Permission for the NFS Export



```
[root@storage test]# cd /media//nfs1/
[root@storage nfs1]# ls -la
total 0
drwxr-xr-x. 3 root root 19 Oct 21 22:22 .
drwxr-xr-x. 6 root root 52 Oct 21 08:13 ..
drwxrwxr-x. 3 root root 17 Oct 21 22:30 weezer
[root@storage nfs1]#
```

d. Edit the "exports" file located in the /etc directory. The file will be empty; add the following line to the file.

```
/media/nfs1/weezer *(rw,sync,no_root_squash)
```

Report Question – Explain the function of the three options in the entry. Hint: Chapter 20 of the Negus book and the nfs man page are good resources.

- rw
- sync
- no_root_squash

What would the entry look like if you want only to allow access to the NFS share from a host device whose IP address is 192.168.1.4?

e. Start/restart the nfs service and configure it to start on boot.
f. Create the firewall rules to allow access to the following three services, nfs, mountd, and rpc-bind. You may wish to make use of the `--add-service` flag of `firewall-cmd`. Alternatively, you can use the Firewall Configuration Window through *Applications* → *Sundry* → *Firewall* and check the services for ***nfs*** and ***rpc-bind*** which will open the required ports for clients to mount to the NFS export.
g. On the client, install the nfs-utils packages.
h. Create a directory to mount the NFS share. Usually, these are placed in /media. For example, I created the */media/nfsmount* directory.
i. To mount to the share enter the following command as root, where “storage” is the hostname of the device that the NFS share is located on. Your hostname may be different.

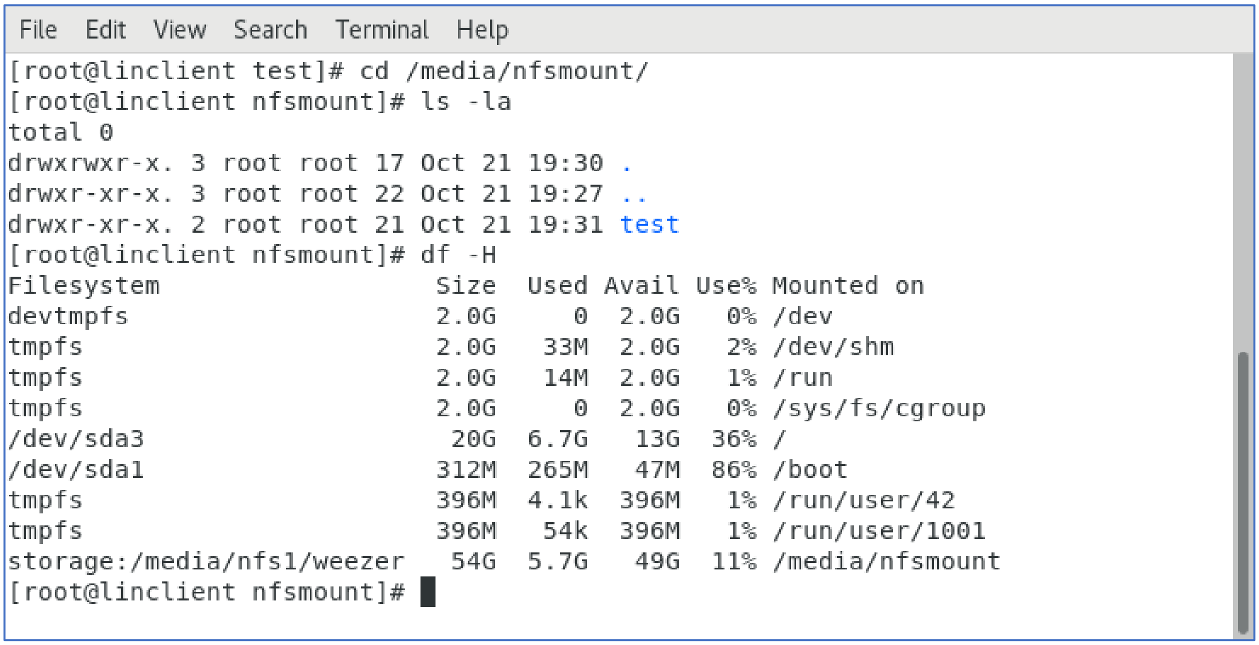
```
# mount storage:/media/nfs1/weezer /media/nfsmount
```

j. Perform a test to confirm that the share can be accessed by creating a file or directory in the remote NFS share.

IMPORTANT
NOTICE

For the report, provide a single screenshot showing that you can navigate to the NFS export and use the `ls -la` command to show the permissions and ownership of the file or directory on the shared resource. Also, include the output from the, hostname and whoami commands. Finally, use the `df -H` command to show that you are currently mounted to the resource. Refer to Figure 31 for an example.

Figure 31 – Sample Linux Client NFS Mount Verification



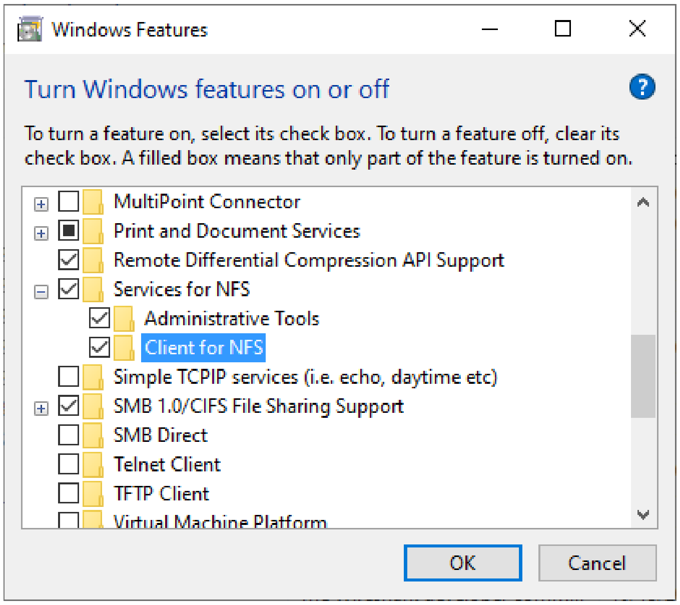
```
[root@linclient test]# cd /media/nfsmount/
[root@linclient nfsmount]# ls -la
total 0
drwxrwxr-x. 3 root root 17 Oct 21 19:30 .
drwxr-xr-x. 3 root root 22 Oct 21 19:27 ..
drwxr-xr-x. 2 root root 21 Oct 21 19:31 test
[root@linclient nfsmount]# df -H
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    2.0G     0  2.0G   0% /dev
tmpfs                       2.0G   33M  2.0G   2% /dev/shm
tmpfs                       2.0G   14M  2.0G   1% /run
tmpfs                       2.0G     0  2.0G   0% /sys/fs/cgroup
/dev/sda3                    20G  6.7G   13G  36% /
/dev/sda1                   312M  265M   47M  86% /boot
tmpfs                       396M  4.1k  396M   1% /run/user/42
tmpfs                       396M   54k  396M   1% /run/user/1001
storage:/media/nfs1/weezer   54G  5.7G   49G  11% /media/nfsmount
[root@linclient nfsmount]# 
```

k. Edit the /etc/fstab file so that the NFS export mounts when the system boots up. The entry will look similar to the example provided below. It is important to identify the mount as an "nfs" export and to use "_netdev" as the option (in bold for emphasis).

```
# storage:/media/nfs1/weezer /media/nfsmount nfs _netdev 0 0
```

j. On the windows 10 client, turn on the Windows feature "Services for NFS" (Figure 32).

Figure 32 – Services for NFS



Turn Windows features on or off

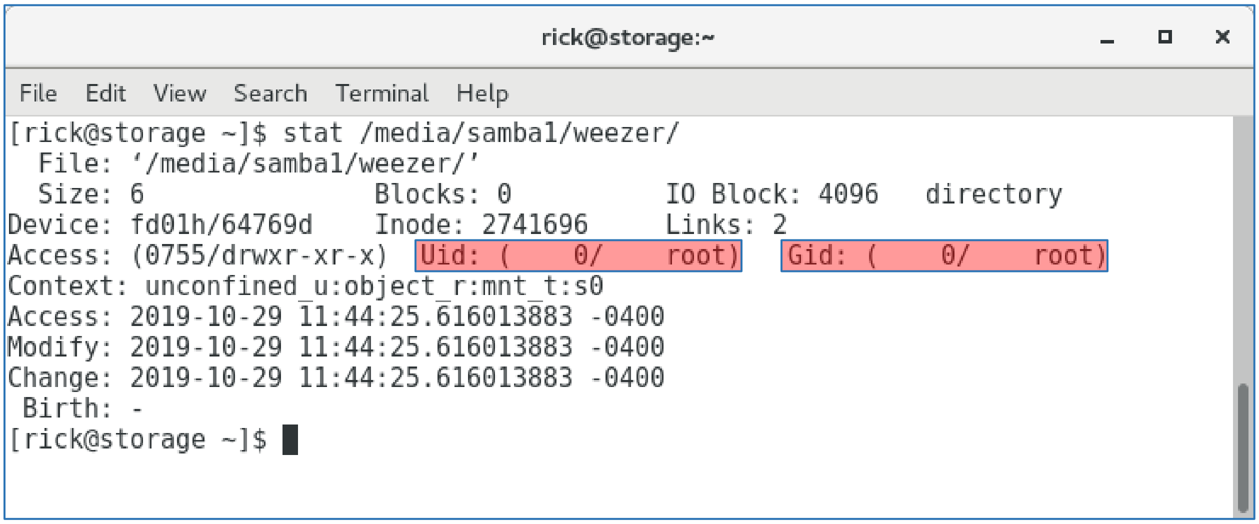
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ⊞ ☐ MultiPoint Connector
- ⊞ ■ Print and Document Services
- ☑ Remote Differential Compression API Support
- ⊟ ☑ Services for NFS
  - ☑ Administrative Tools
  - ☑ Client for NFS
- ☐ Simple TCPIP services (i.e. echo, daytime etc)
- ⊞ ☑ SMB 1.0/CIFS File Sharing Support
- ☐ SMB Direct
- ☐ Telnet Client
- ☐ TFTP Client
- ☐ Virtual Machine Platform

k. Next, enable write permissions for the anonymous user by editing the registry. Open the Registry Editor (regedit) and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsfot\ClientForNFS\CurrentVersion\Default**.

l. On the storage server, use the stat command to find the UID and GID of the NFS export (Figure 33).

Figure 33 – Stat Command on NFS Export

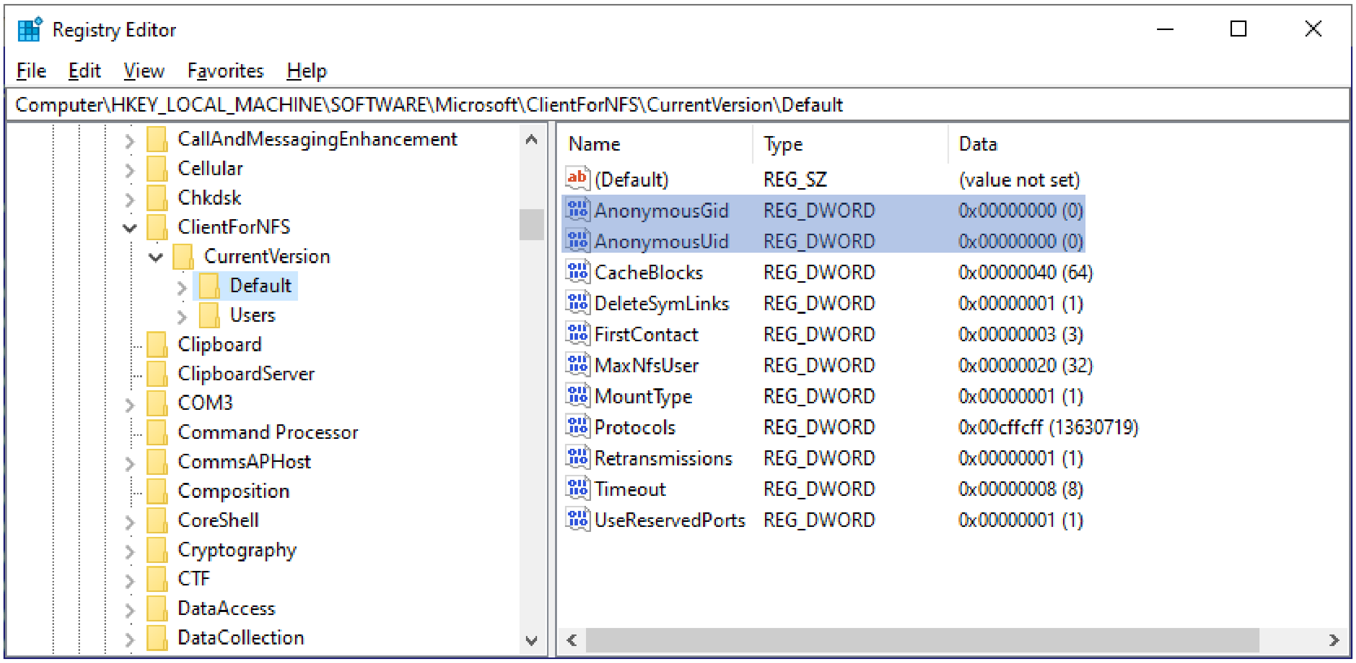


```
[rick@storage ~]$ stat /media/samba1/weezer/
  File: '/media/samba1/weezer/'
  Size: 6               Blocks: 0          IO Block: 4096   directory
Device: fd01h/64769d    Inode: 2741696     Links: 2
Access: (0755/drwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:mnt_t:s0
Access: 2019-10-29 11:44:25.616013883 -0400
Modify: 2019-10-29 11:44:25.616013883 -0400
Change: 2019-10-29 11:44:25.616013883 -0400
 Birth: -
[rick@storage ~]$ 
```

m. Create a new **New DWORD (32-bit) Value** named **AnonymousUid** and assign the UID found on the NFS Export.

n. Create a new **New DWORD (32-bit) Value** named **AnonymousGid** and assign the GID found on the NFS Export.

Figure 34 – Registry Entries



Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ClientForNFS\CurrentVersion\Default

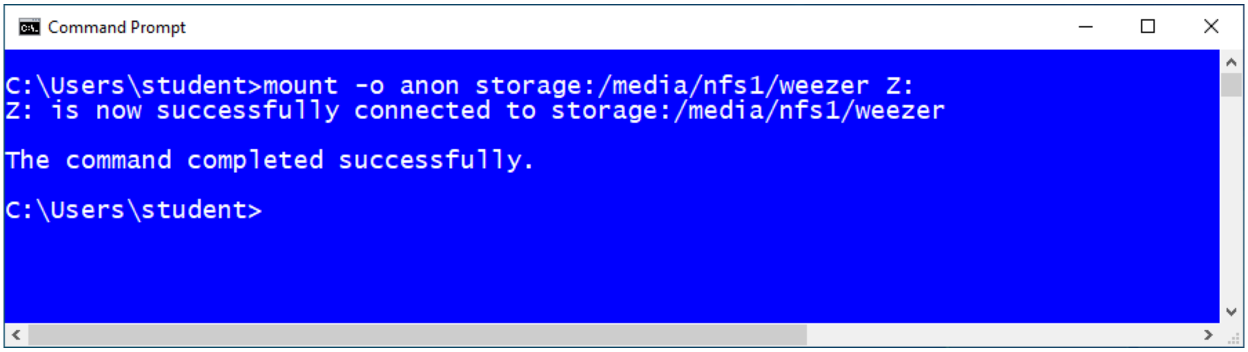
Name	Type	Data
(Default)	REG_SZ	(value not set)
AnonymousGid	REG_DWORD	0x00000000 (0)
AnonymousUid	REG_DWORD	0x00000000 (0)
CacheBlocks	REG_DWORD	0x00000040 (64)
DeleteSymLinks	REG_DWORD	0x00000001 (1)
FirstContact	REG_DWORD	0x00000003 (3)
MaxNfsUser	REG_DWORD	0x00000020 (32)
MountType	REG_DWORD	0x00000001 (1)
Protocols	REG_DWORD	0x00cffcff (13630719)
Retransmissions	REG_DWORD	0x00000001 (1)
Timeout	REG_DWORD	0x00000008 (8)
UseReservedPorts	REG_DWORD	0x00000001 (1)

o. Restart Windows

p. To mount to the NFS export, open the Windows Command Prompt and type the following command. This command will mount the NFS export anonymously to drive letter "Z:". The output will look similar to Figure 35.

```
> mount -o anon storage:/media/nfs1/weezer Z:
```

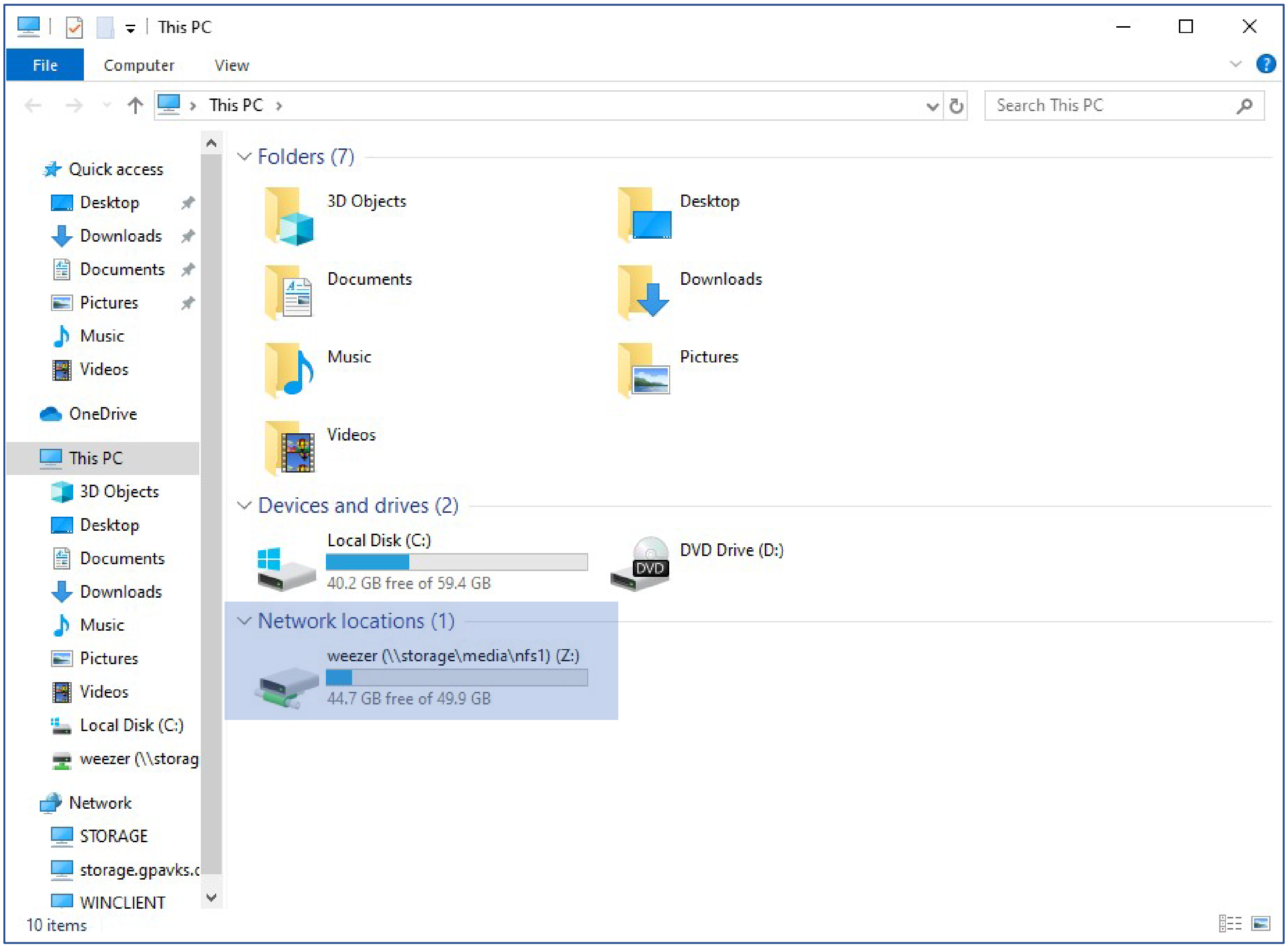
Figure 35 – Mount Success



```
C:\Users\student>mount -o anon storage:/media/nfs1/weezer Z:
Z: is now successfully connected to storage:/media/nfs1/weezer

The command completed successfully.

C:\Users\student>
```

q. You can also open “This PC” and you will see the NFS export listed under “Network Locations” (Figure 36).

Figure 36 – Network Locations

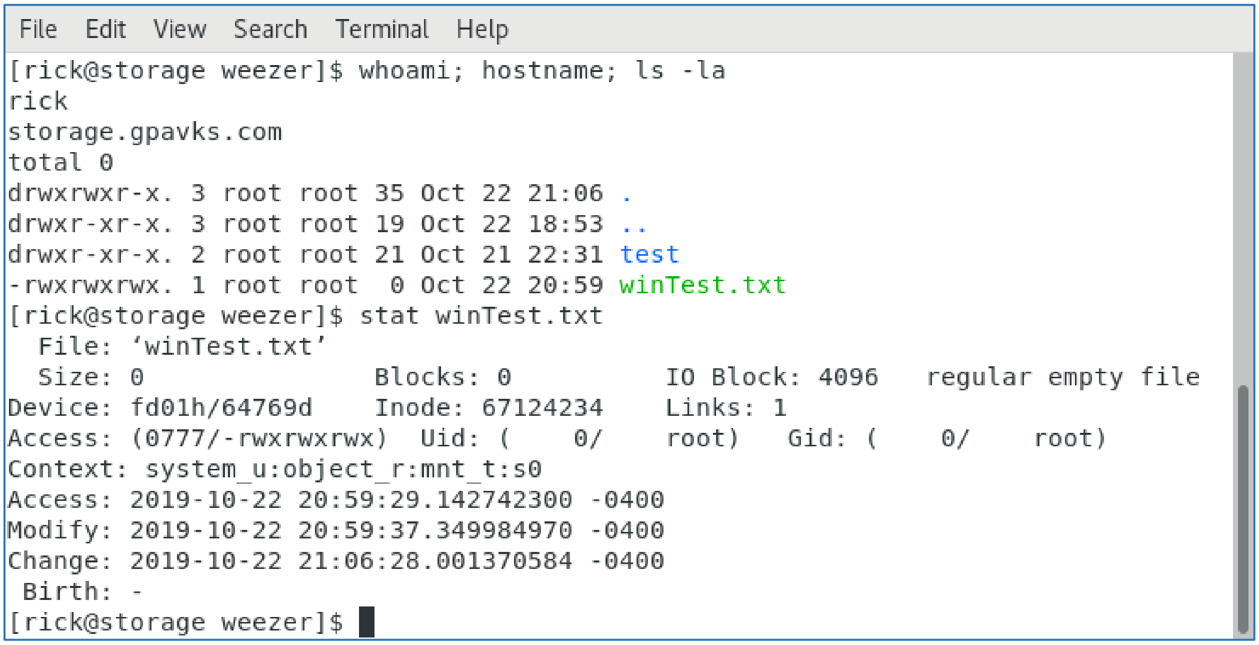


r. Double-click the export and create a file or directory.

IMPORTANT NOTICE

For the report, you will need to provide two screenshots. The first will show the file/directory created from the Windows client. On the server open a terminal, navigate to the NFS export directory and show the output from the **`whoami`**, **`hostname`**, **`ls -la`**, and **`stat`** command (you are using the stat command on the file/directory created from the Windows 10 client). Figure 37 provides an example of the output.

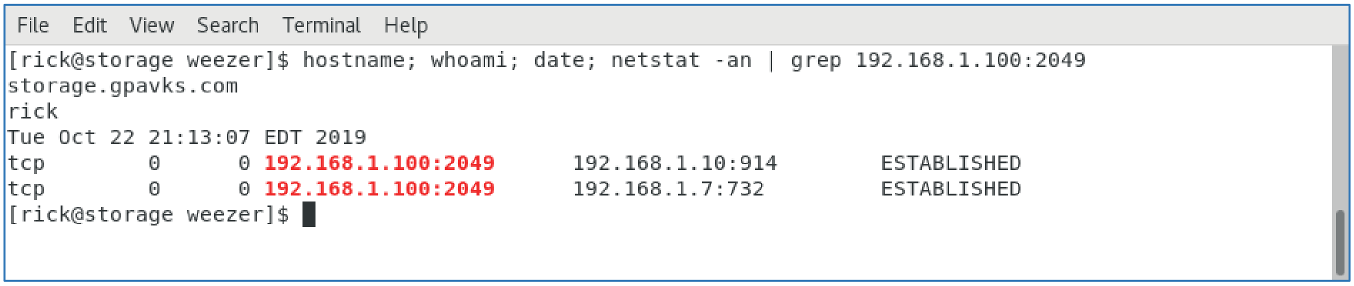
Figure 37 – Sample Windows NFS Write Verification



```
[rick@storage weezer]$ whoami; hostname; ls -la
rick
storage.gpavks.com
total 0
drwxrwxr-x. 3 root root 35 Oct 22 21:06 .
drwxr-xr-x. 3 root root 19 Oct 22 18:53 ..
drwxr-xr-x. 2 root root 21 Oct 21 22:31 test
-rwxrwxrwx. 1 root root  0 Oct 22 20:59 winTest.txt
[rick@storage weezer]$ stat winTest.txt
  File: ‘winTest.txt’
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: fd01h/64769d    Inode: 67124234    Links: 1
Access: (0777/-rwxrwxrwx)  Uid: (    0/    root)   Gid: (    0/    root)
Context: system_u:object_r:mnt_t:s0
Access: 2019-10-22 20:59:29.142742300 -0400
Modify: 2019-10-22 20:59:37.349984970 -0400
Change: 2019-10-22 21:06:28.001370584 -0400
 Birth: -
[rick@storage weezer]$ 
```

The second screenshot will show the connections from the Windows and Linux clients. From the terminal enter the **`whoami`**, **`hostname`**, and **`date`** commands. To show the NFS connections from both clients use the **`netstat -an`** command, pipe, and grep for the servers IP address followed by the NFS well-known port 2049. For an example, see Figure 38.

Figure 38 – Sample Network Connection Verification



```
[rick@storage weezer]$ hostname; whoami; date; netstat -an | grep 192.168.1.100:2049
storage.gpavks.com
rick
Tue Oct 22 21:13:07 EDT 2019
tcp        0      0 192.168.1.100:2049      192.168.1.10:914        ESTABLISHED
tcp        0      0 192.168.1.100:2049      192.168.1.7:732         ESTABLISHED
[rick@storage weezer]$ 
```

SCREENSHOT SUMMARY

READ PLEASE

All screenshots for Lab 5 must be included in the lab report. For each missing screenshot you will receive a 5% penalty to the lab report grade. If your screenshots do not include the required information, are illegible, blurry, or otherwise unreadable, you will not receive credit. Any attempt to alter the information in the screenshots in any way is academic dishonesty, and you will fail the course.

All screenshots **must** be labeled, using the following titles.

Figure 1 – Forward Lookup Zone
Figure 2 – Reverse Lookup Zone
Figure 3 – Rsync History
Figure 4 – FTP User Login
Figure 5 – FTP Anonymous Login
Figure 6 – FTP xferlog File
Figure 7 – Remote Samba Access and Write Verification
Figure 8 – Windows Client Samba Verification
Figure 9 – Server Samba Status
Figure 10 – RSYNC File Transfer
Figure 11 – Linux Client NFS Verification
Figure 12 – Windows NFS Write Verification
Figure 13 – NFS Network Connection Verification