NSSA221 Systems Administration I
Lab 05 Report

Name: Edwin Chen   Grade ___/100

Requirements:

Use this document to answer questions related to Lab 04. Submit this template to the LR04 drop box by the due date. Late submissions are subject to a 20% penalty. You must use this template and submit the document in Microsoft Word any other format will result in a zero grade for the report.

When submitting your material, please use the following format; LRXX, followed by an underscore, and the first initial of your first name in uppercase, and your entire last name with the first letter in uppercase. Example: LR04_GArcoraci.doc.

Answer ALL questions in your OWN words. Your answer should give some indication that you understand the topic in question and can relate it to your experience in the lab, and not a general textbook answer. DO NOT copy and paste or plagiarize. You may use an outside resource and cite it accordingly to support your answer, but it should **NOT** be your entire answer.

Your answers must be complete sentences and use proper sentence structure. Phrases or one-word responses will receive a zero. Each response is worth a maximum of 20 points.

1. Describe the functionality of the `rsync` command in Linux systems for file synchronization and data transfer. How does `rsync` improve efficiency and accuracy compared to traditional file copying methods? Provide examples of `rsync` command options for different use cases, such as local file synchronization, remote file transfer, and incremental backup, explaining how each option enhances the command's functionality.

   The rsync command in Linux systems utilizes files and directories locally or between systems over network to transfer changed or new data while minimizing the bandwidth usage whilst improving speed. Comparing it to default Linux file methods like cp, rsync ensures the file is accurate by verifying the data and rsync supports backups to save resources and time. Options that can be utilized in rsyc include -a for archiving, -z for compression, and –delete for the cleaning tasks in local servers.

2. In the context of FTP file transfers, explain how you can verify on the server side that a file has been successfully transferred from the client, especially when direct verification on the client side is not possible. Describe the role of the xferlog file in this verification process and provide an example of how you would inspect this log to confirm a successful file transfer.

   To verify if an FTP file transfer was successful you can check the xferlog file which holds the details of all commands like file transfers, their status, etc. The log records information about the file name, the size, the time it took to transfer, and the status of completion. An entry with today's date and a "c" at the end of the entry would indicate that the file was transfer was complete.

3. Discuss the role and configuration of the smb.conf file in setting up and managing a Samba server in Linux. How does smb.conf allow for the customization of Samba services, such as file sharing and printer services in a network? Provide examples of key directives in the smb.conf file and explain how they influence the behavior of the Samba server. Additionally, discuss the importance of security considerations in configuring smb.conf.

The smb.conf file is a configuration file for setting up a Samba server in Linux. It enables the customization of services for files within a network. Configuring the [global] allows for serverwide settings, security for authentications within the network, read-only for control write access, and more. There is also security settings for valid users, encrypted passwords, and restrictions on sensitive data so if a config is poorly configured it could expose the server to unauthorized access.

4. Discuss the role of the `/etc/exports` file in configuring NFS services on a Linux system. What are the key options and parameters that can be specified in this file, and how do they impact the behavior and security of NFS shares? Provide examples of different configurations in the `/etc/exports` file for various network sharing scenarios, explaining how each configuration tailors the NFS service to specific needs.

The /etc/exports file is a file that configures the NFS services on a Linux system. It does this by defining which directories it is allowed to share, to whom to share, and what permissions the user has. Each entry specifies the directory that a user is allowed access to with options to change if it is only read-only (ro) or read-write (rw). For example if I specified 192.168.1.0/24(rw,sync) it would allow read-write to all users in that subnet. The options in the configuration file of the exports file provide a balance in usability and security.

5. Compare and contrast the approaches for configuring user access in Samba and NFS systems. How do user permissions and authentication differ between Samba shares and NFS mounts? Provide examples of configuration settings from both `/etc/samba/smb.conf` for Samba and `/etc/exports` for NFS that demonstrate how user access is managed in each system.

Samba and NFS differ in the aspects of user security in authentication and permissions. Samba uses a password-based database and settings in the samba.conf for valid users and read-only for restricted access. NFS specifies a more in-depth system that would only allow access based on the IP address or hostnames.
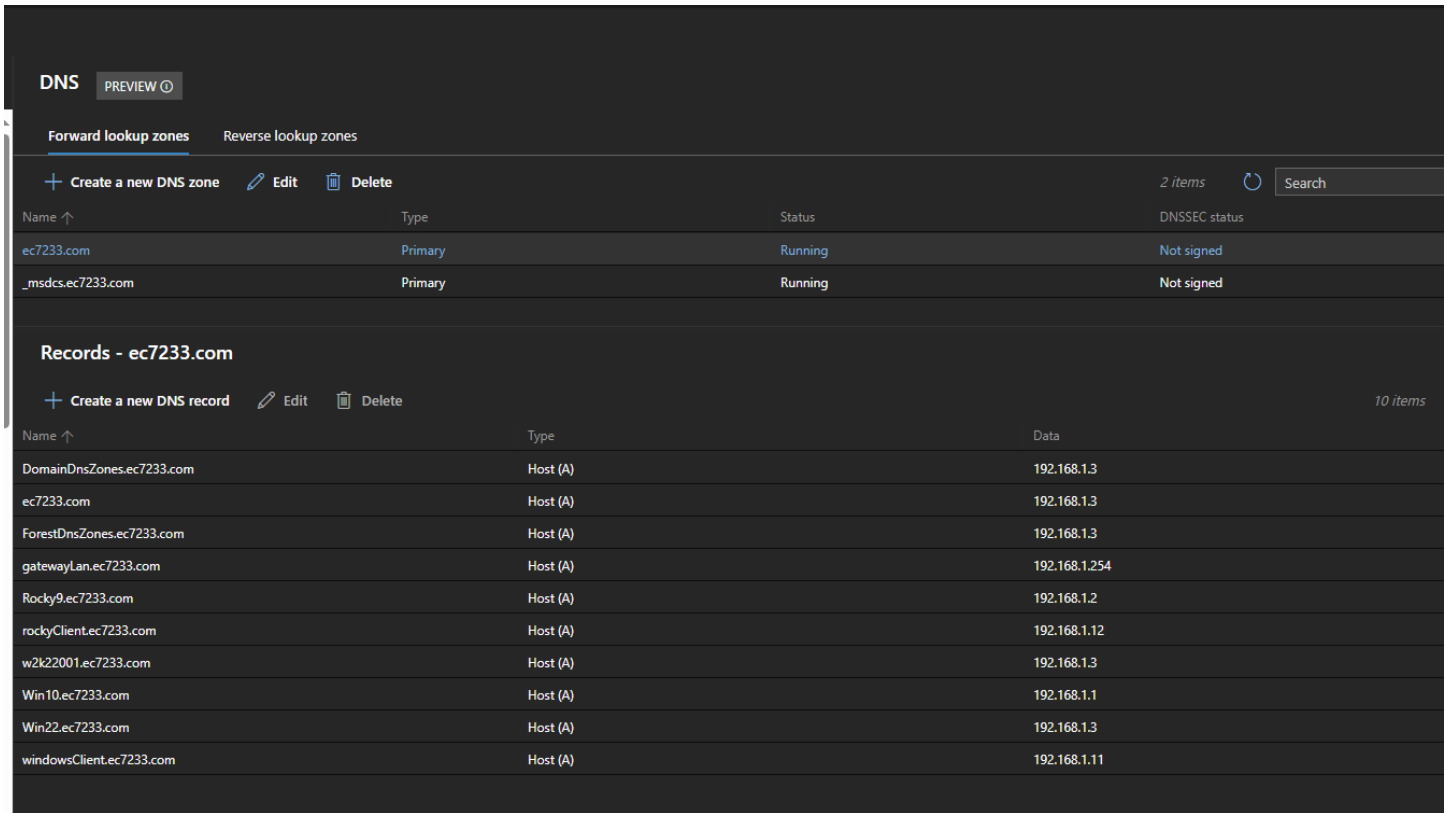
PLEASE READ CAREFULLY

All screenshots for the lab must be included in the report. If you are missing more than three screenshots your grade is a zero. 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 is academic dishonesty, and you will receive a zero grade for the report.

All screenshots must be labeled using the following titles.
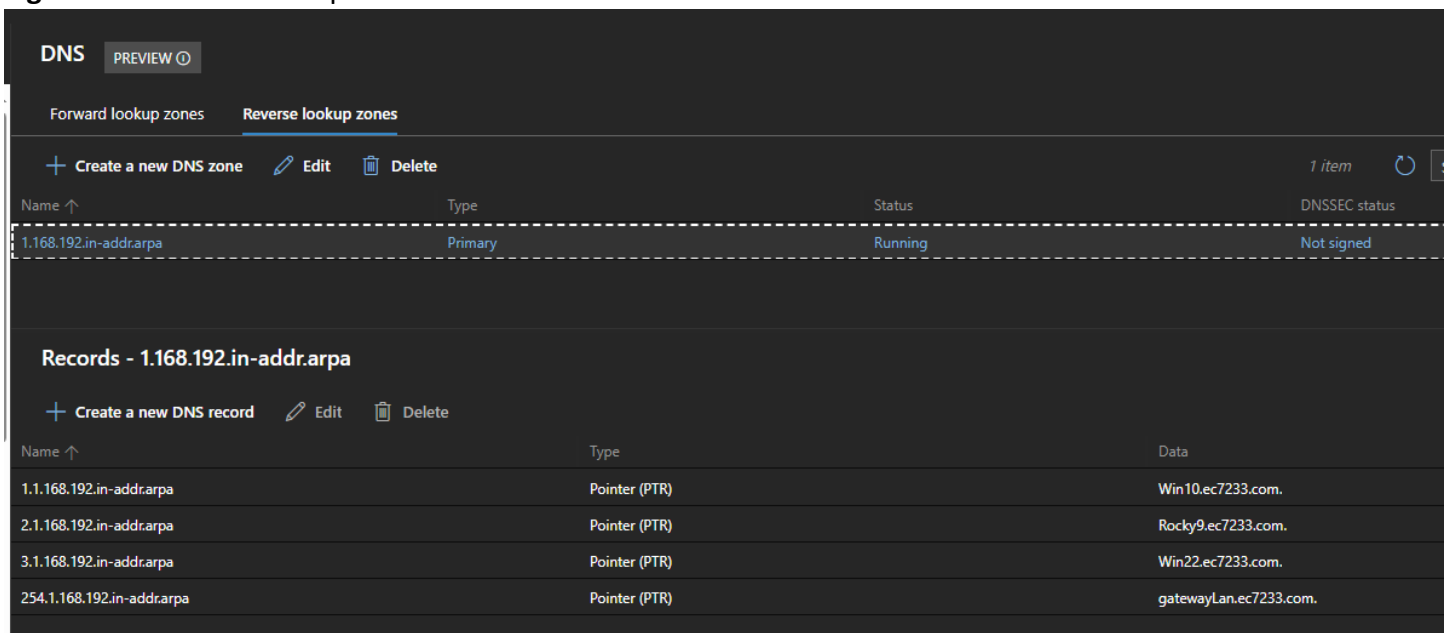
Figure 1 – Forward Lookup Zone

Figure 2 – Reverse Lookup Zone

Figure 3 – Rsync History

```
[student@localhost ~]$ hostname; date; ls original/ backup/
localhost
Mon Nov  4 12:45:14 PM EST 2024
backup/:
ATT00001.txt  ATT00005.txt  IMG_1005.jpg  IMG_1012.jpg  IMG_1031.jpg
ATT00002.txt  IMG_0998.jpg  IMG_1006.jpg  IMG_1019.jpg  IMG_1046.jpg
ATT00003.txt  IMG_0999.jpg  IMG_1007.jpg  IMG_1028.jpg  IMG_1051.jpg
ATT00004.txt  IMG_1004.jpg  IMG_1011.jpg  IMG_1030.jpg

original/:
ATT00001.txt  ATT00005.txt  IMG_1004.jpg  IMG_1011.jpg  IMG_1030.jpg
ATT00002.txt  ec7233.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@localhost ~]$
```

```
[student@localhost ~]$ hostname; date; history
rocky9.ec7233.com
Mon Nov  4 12:54:40 PM EST 2024
    1  rsync -av original/ backup --log-file=ec7233.log
    2  rsync -av --exclub '*.jpg' original/ backup
    3  rsync -av --exclude '*.jpg' original/ backup
    4  touch test{1..9}.txt
    5  rsync -av --delete original/ backup
    6  rsync -av --delete original backup
    7  rsync --dry-run --remove-source-files -av original/ destination/
    8  rsync --remove-source-files -av original/ destination/
    9  hostname; date; history
[student@localhost ~]$
```

Figure 4 – FTP User Login


```
[student@rocky9 ~]$ hostname; date; whoami
rocky9.ec7233.com
Thu Nov 14 12:41:19 PM EST 2024
student
[student@rocky9 ~]$ lftp -u r2d2 localhost
Password:
lftp r2d2@localhost:~> ls -a
drwx------    3 1006     1006           78 Nov 14 17:39 .
drwxr-xr-x   12 0        0             145 Nov 14 17:39 ..
-rw-r--r--    1 1006     1006           18 Apr 30  2024 .bash_logout
-rw-r--r--    1 1006     1006          141 Apr 30  2024 .bash_profile
-rw-r--r--    1 1006     1006          492 Apr 30  2024 .bashrc
lftp r2d2@localhost:~>
```

Figure 5 – FTP Anonymous Login


```
[student@rocky9 ~]$ hostname; date; whoami
rocky9.ec7233.com
Thu Nov 14 12:42:20 PM EST 2024
student
[student@rocky9 ~]$ lftp localhost
lftp localhost:~> ls -a
drwxr-xr-x    3 0        0             17 Nov 04 17:58 .
drwxr-xr-x    3 0        0             17 Nov 04 17:58 ..
drwxr-xr-x    2 0        0              6 Oct 30  2023 pub
lftp localhost:/>
```

Figure 6 – FTP xferlog FileS

```
[student@rocky9 log]$ hostname; sudo cat xferlog | grep test1.txt
rocky9.ec7233.com
Thu Nov 14 12:58:38 2024 1 ::ffff:127.0.0.1 0 /dropbox/test1.txt b _ i a lftp@ ftp 0 * c
[student@rocky9 log]$
```

Figure 7 – Remote Samba Access and Write Verification

```
[student@localhost ~]$ hostname; whoami
rocky9.ec7233.com
student
[student@localhost ~]$ smbclient -U johnny //storage/ramones
Password for [SAMBA\johnny]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Nov 15 15:11:08 2024
  ..                                  D        0  Fri Nov 15 15:11:08 2024
  lyrics.txt                          N        3  Fri Nov 15 15:11:03 2024
  chords.txt                          N        3  Fri Nov 15 15:11:08 2024

              4128768 blocks of size 1024. 4066912 blocks available
smb: \> mkdir test
smb: \> ls
  .                                   D        0  Fri Nov 15 15:32:10 2024
  ..                                  D        0  Fri Nov 15 15:32:10 2024
  lyrics.txt                          N        3  Fri Nov 15 15:11:03 2024
  chords.txt                          N        3  Fri Nov 15 15:11:08 2024
  test                                D        0  Fri Nov 15 15:32:10 2024

              4128768 blocks of size 1024. 4066912 blocks available
smb: \> █
```

Figure 8 – Windows Client Samba Verification

Figure 9 – Server Samba Status

Figure 10 – RSYNC File Transfer

Figure 11 – Linux Client NFS Verification

```
student@rocky9:/media/nfsmount

[student@rocky9 nfsmount]$ ls -la
total 0
drwxrwxr-x. 2 root root 22 Nov 15 16:25 .
drwxr-xr-x. 3 root root 22 Nov 15 16:22 ..
-rw-r--r--. 1 root root  0 Nov 15 16:25 test.txt
[student@rocky9 nfsmount]$ df -H
Filesystem                 Size  Used Avail Use% Mounted on
devtmpfs                   4.2M     0  4.2M   0% /dev
tmpfs                      2.0G     0  2.0G   0% /dev/shm
tmpfs                      768M  9.8M  758M   2% /run
/dev/mapper/rl-root         48G  6.2G   42G  13% /
/dev/sda2                  1.1G  422M  586M  42% /boot
/dev/sda1                  628M  7.4M  621M   2% /boot/efi
tmpfs                      384M   99k  384M   1% /run/user/1000
storage:/media/nfs1/weezer 1.1G   41M  967M   5% /media/nfsmount
[student@rocky9 nfsmount]$ S
```

Figure 12 – Windows NFS Write Verification

```
[root@storage weezer]# whoami; hostname; ls -la
root
storage.ec7233.com
total 4
drwxrwxr-x. 2 root root 45 Nov 15 16:35 .
drwxr-xr-x. 3 root root 20 Nov 15 16:14 ..
-rw-r--r--. 1 root root  0 Nov 15 16:25 test.txt
-rwxr-xr-x. 1 root root 20 Nov 15 16:36 winTest.txt.txt
[root@storage weezer]# stat winTest.txt.txt
  File: winTest.txt.txt
  Size: 20            Blocks: 8          IO Block: 4096   regular file
Device: 10309h/66313d   Inode: 133         Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Context: system_u:object_r:unlabeled_t:s0
Access: 2024-11-15 16:36:09.246770635 -0500
Modify: 2024-11-15 16:36:09.263770723 -0500
Change: 2024-11-15 16:36:09.263770723 -0500
 Birth: 2024-11-15 16:35:43.320635061 -0500
[root@storage weezer]# 
```

Figure 13 – NFS Network Connection Verification

```
[root@storage weezer]# hostname; whoami; date; netstat -an | grep 192.168.1.2:2049
storage.ec7233.com
root
Fri Nov 15 04:37:33 PM EST 2024
tcp        0      0 192.168.1.2:2049        192.168.1.16:983        ESTABLISHED
tcp        0      0 192.168.1.2:2049        192.168.1.11:853        ESTABLISHED
[root@storage weezer]#
```