LECTURE 8

Configure and Manage Group Policy

- Configure and Manage Group Policy
  - Configure Group Policy in Windows
  - Managing Group Policy in Windows
  - Linux Users and Groups

- Group Policy is a tool that is available to administrators that are running a Windows 2000 or later Active Directory Domain.
- It allows for centralized management of settings on client computers and servers joined to the domain as well as providing a rudimentary way to distribute software.
- Group Policy provides centralized management and configuration of operating systems, applications, and users' settings in an Active Directory environment.
- A set of Group Policy configurations is called a Group Policy Object (GPO).

- **Active Directory Domain Services (ADDS) is a server role within Microsoft Windows that is used to store and structure objects.**
- **Objects managed within ADDS can be computers, users or groups.**
- **To add additional configuration and management to object types within ADDS, group policy is used.**
- **Group Policy is a feature within Windows used to control configuration and behavior settings.**
- **A collection or group of settings are called group policy objects.**

ef: https://study.com/academy/lesson/group-policy-objects-in-windows-server-2012-r2-overview-types.html#:~:text
There%20are%20three%20types%20of%20GPOs%20%2D%20local%2C%20nonlocal%20and%20starter,create%20new
20GPOs%20within%20ADDS.

- **Introduction to Group Policy**
- https://www.youtube.com/watch?v=cKbgHaQG6BI&list=PL1l78n6W8zyor7Fd46FQpFL8qo4cb0xMT
- **Installing group policy**
- https://www.youtube.com/watch?v=7LxGCExHDgU&list=PL1l78n6W8zyor7Fd46FQpFL8qo4cb0xMT&index=2

Windows group policy types

- **The three Group Policy types are Local, Non-local, and Starter.**
  - **Local GPOs (GPO stands for Group Policy Object) apply to the local computer only on Windows client.**
  - **Often referred to as LGPO which stands for Local Group Policy Object.**
  - **Non-local- apply settings to one or multiple Windows clients by linking them to sites, domains or organizational units (OUs) within ADDS.**
  - **Starter - are templates used to create new GPOs within ADDS.**
- **Group Policy Types and components**
- **https://youtu.be/0sei0cE2aUw**

*** Active Directory Domain Services (ADDS)**

- **Benefits of Group Policy Objects**

- **Ease of administration -- system administrators can deploy software, patches and other updates via GPO.**
- **Better password policy enforcement -- GPOs determine password length, reuse rules and establish other requirements for passwords to keep a company's network safe.**

- **Group Policy Processing Order**
- https://www.youtube.com/watch?v=UmEyq49rYyk

- **Group Policy Preferences**
- https://www.youtube.com/watch?v=vPCiFU_0I5E

- **Components of Group Policy Settings Preferences**
- https://www.youtube.com/watch?v=M_JORkMO62U&list=PL1I78n6W8zyor7Fd46FQpFL8qo4cb0xMT&index=4

- **Creating and linking GPOs**
- https://www.youtube.com/watch?v=Igru_UrBVZI&list=PL1I78n6W8zyor7Fd46FQpFL8qo4cb0xMT&index=5

- **Filtering Group Policy**
- https://www.youtube.com/watch?v=hkBWKEw9I6I&list=PL1I78n6W8zyor7Fd46FQpFL8qo4cb0xMT&index=8

- **Configure a Central Store**
  - https://www.youtube.com/watch?v=0-0luRhTES4&list=PL1I78n6W8zyor7Fd46FQpFL8qo4cb0xMT&index=9

Linux Users and Groups

- Linux was designed to allow more than one user to have access to the system at the same time.
- In order for this multiuser design to work properly, there needs to be a method to protect users from each other.
- This is where permissions come in to play.

Ref: https://www.linode.com/docs/guides/linux-users-and-groups/

- **User groups play an important role on Linux systems.**
- **They provide an easy way for a selected groups of users to share files with each other.**
- **They also allow sysadmins to more effectively manage user privileges, since they can assign privileges to groups rather than individual users.**

Ref: https://www.networkworld.com/article/3409781/mastering-user-groups-on-linux.html

Read, Write & Execute Permissions

- Permissions are the “rights” to act on a file or directory.
- The basic rights are read, write, and execute.
  - Read - a readable permission allows the contents of the file to be viewed.
    - A read permission on a directory allows you to list the contents of a directory.
  - Write - a write permission on a file allows you to modify the contents of that file. For a directory, the write permission allows you to edit the contents of a directory (e.g. add/delete files).
  - Execute - for a file, the executable permission allows you to run the file and execute a program or script. For a directory, the execute permission allows you to change to a different directory and make it your current working directory.
- Users usually have a default group, but they may belong to several additional groups.

Viewing File Permissions

- To view the permissions on a file or directory, issue the command ls -l <directory/file>.
- Remember to replace the information in the <> with the actual file or directory name.
- Below is sample output for the ls command:

  -rw-r--r-- 1 root root 1031 Nov 18 09:22 /etc/passwd
- The first ten characters show the access permissions.

Viewing File Permissions

- The first dash (-) indicates the type of file (d for directory, s for special file, and - for a regular file).
- The next three characters (rw-) define the owner's permission to the file.
- In this example, the file owner has read and write permissions only.
- The next three characters (r–) are the permissions for the members of the same group as the file owner (which in this example is read only).
- The last three characters (r–) show the permissions for all other users and in this example, it is read only.

Creating and Deleting User Accounts

- To create a new standard user, use the useradd command.
- The syntax is as follows:
  - useradd <name>

Add user

- **Most user accounts on Linux systems are set up with the user and group names the same.**
- **The user "jdoe" will be set up with a group named "jdoe" and will be the only member of that newly created group.**
- **The user's login name, user id, and group id will be added to the /etc/passwd and/etc/group files when the account is added.**

Ref: https://www.networkworld.com/article/3409781/mastering-user-groups-on-linux.html

The useradd command utilizes a variety of variables, some of which are shown in the table below:

Option	Description	Example
`-d <home_dir>`	home_dir will be used as the value for the user's login directory	`useradd <name> -d /home/<user's home>`
`-e <date>`	the date when the account will expire	`useradd <name>** -e <YYYY-MM-DD>`
`-f <inactive>`	the number of days before the account expires	`useradd <name> -f <0 or -1>`
`-s <shell>`	sets the default shell type	`useradd <name> -s /bin/<shell>`

```
passwd <username>
```

The user will be able to change their password at any time using the `passwd` command with the syntax. Below is an example:

```
$ passwd
Changing password for lmartin.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

To remove the user

- To remove the user, their home folder, and their files, use this command:
  - userdel -r <name>

Creating and Removing Directories

To make a directory use the command:

```
mkdir <directory name>
```

To make a directory and set the permissions at the same time, use the following option and syntax:

```
mkdir -m a=rwx <directory name>
```

The **-m** option is short for mode, and **a=rwx** means that all users have read, write, and execute permissions on the directory. To see a complete list of all options for the mkdir command enter `man mkdir` at a command prompt.

To remove a file, use the following:

```
rm <file>
```

To remove a directory:

```
rm -r <directory name>
```

It is important to note that if you remove a directory all the files inside will be deleted as well.

Changing Directory and File Permissions

To view file permissions and ownership on files and directories, use the `ls -al` command. The `a` option is to show hidden files or all files, and the `l` option is for the long listing. The output will be similar to the following:

```
drwxr-xr-x 2 user user 4096 Jan  9 10:11 documents
-rw-r--r-- 1 user user  675 Jan  7 12:05 .profile
drwxr-xr-x 4 user user 4096 Jan  7 14:55 public
```

The first column with the ten letters and dashes shows the permissions of the file or directory. The second column (with the single number) indicates the number of files or directories contained in the directory. The next column indicates the owner, followed by the group name, the size, date, and time of last access, and finally the name of the file .

Explanation of each lines

`drwxr-xr-x` are the permissions
`2` is the number of files or directories
`user` is the owner

`user` is the group
`4096` is the size
`Jan  9 10:11` is the date/time of last access
`documents` is the directory

chmod Command

- The command chmod is short for change mode.
- chmod is used to change permissions on files and directories.
- The command chmod maybe used with either letters or numbers (also known as octal) to set the permissions.

Ref: https://www.linode.com/docs/guides/linux-users-and-groups

Letter	Permission
r	Read
w	Write
x	Execute
X	Execute (only if file is a directory)
s	Set user or group ID on execution
t	Save program text on swap device
u	Current permissions the file has for owner
g	Current permissions the file has for users in the same group
o	Current permissions the file has for others not in the group

- **It is important to remember that the first character of the first column of a file listing denotes whether it is a directory or a file.**
- **The other nine characters are the permissions for the file/directory.**
- **The first three characters are for the user, the next three are for the group, and the last three are for others.**
- **The example drwxrw-r– is broken down as follows:**

"**d** is a directory

rwx the user has read, write, and execute permissions

rw- the group has read and write permissions

r– all others have read only permissions"

chmod Command

Conversely, the plus sign (+) is equivalent to granting permissions: `chmod u+r,g+x <filename>`

The example above translates as follows:

```
u is for user
r is for read
g is for group
x is for execute
```

In other words, the user was given read permission and the group was given execute permission for the file. Note, when setting multiple permissions for a set, a comma is required between sets.

Chmod Octal Format

To use the octal format, you have to calculate the permissions for each portion of the file or directory. The first ten characters mentioned above will correspond to a four digit numbers in octal. The execute permission is equal to the number one (1), the write permission is equal to the number two (2), and the read permission is equal to the number four (4). Therefore, when you use the octal format, you will need to calculate a number between 0 and 7 for each portion of the permission. A table has been provided below for clarification.

Octal Value	Read	Write	Execute
7	r	w	x
6	r	w	-
5	r	-	x
4	r	-	-
3	-	w	x
2	-	w	-
1	-	-	x
0	-	-	-

Chmod Octal Format

To use the octal format, you have to calculate the permissions for each portion of the file or directory. The first ten characters mentioned above will correspond to a four digit numbers in octal. The execute permission is equal to the number one (1), the write permission is equal to the number two (2), and the read permission is equal to the number four (4). Therefore, when you use the octal format, you will need to calculate a number between 0 and 7 for each portion of the permission. A table has been provided below for clarification.

Octal Value	Read	Write	Execute
7	r (4)	w (2)	x (1)
6	r (4)	w (2)	-
5	r (4)	-	x (1)
4	r	-	-
3	-	w	x
2	-	w	-
1	-	-	x
0	-	-	-

> “Sample syntax: `chmod <octal or letters> <file/directory name>`
>
> Letter format: `chmod go-rwx Work` (Deny rwx permission for the group and others)”

The output of ls -al after the chmod command above would looks as follows:

```
dr-------- 2 user user 4096 Dec 17 14:38 Work
```

Octal format: `chmod 444 Work`

The output of ls -al after the chmod command above would look as follows:

```
dr--r--r-- 2 user user 4096 Dec 17 14:38 Work
```

An octal table showing the numeric equivalent for permissions is provided below.

Permission string	Octal code	Meaning
`rwxrwxrwx`	777	Read, write, and execute permissions for all users.
`rwxr-xr-x`	755	Read and execute permission for all users. The file's owner also has write permission.
`rwxr-x---`	750	Read and execute permission for the owner and group. The file's owner also has write permission. Users who aren't the file's owner or members of the group have no access to the file.
`rwx------`	700	Read, write, and execute permissions for the file's owner only; all others have no access.
`rw-rw-rw-`	666	Read and write permissions for all users. No execute permissions for anybody.

rw-rw-r--	664	Read and write permissions for the owner and group. Read-only permission for all others.
rw-rw----	660	Read and write permissions for the owner and group. No world permissions.
rw-r--r--	644	Read and write permissions for the owner. Read-only permission for all others.
rw-r-----	640	Read and write permissions for the owner, and read-only permission for the group. No permission for others.
rw-------	600	Read and write permissions for the owner. No permission for anybody else.
r--------	400	Read permission for the owner. No permission for anybody else.

By default, all files are “owned” by the user who creates them and by that user’s default group. To change the ownership of a file, use the `chown` command in the `chown user:group /path/to/file` format. In the following example, the ownership of the “list.html” file will be changed to the “cjones” user in the “marketing” group:

```
chown cjones:marketing list.html
```

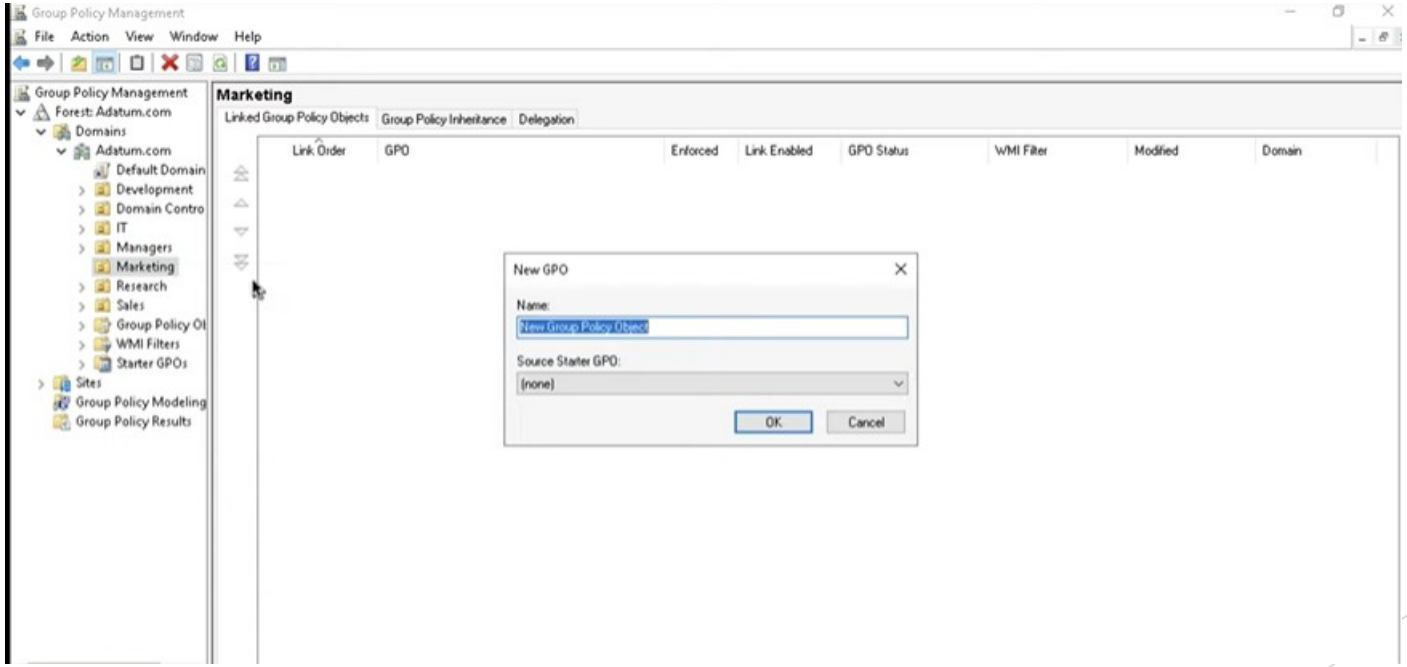
To change the ownership of a directory and all the files contained inside, use the recursive option with the `-R` flag. In the following example, change the ownership of `/srv/smb/leadership/` to the “cjones” user in the “marketing” group:

```
chown -R cjones:marketing /srv/smb/leadership/
```

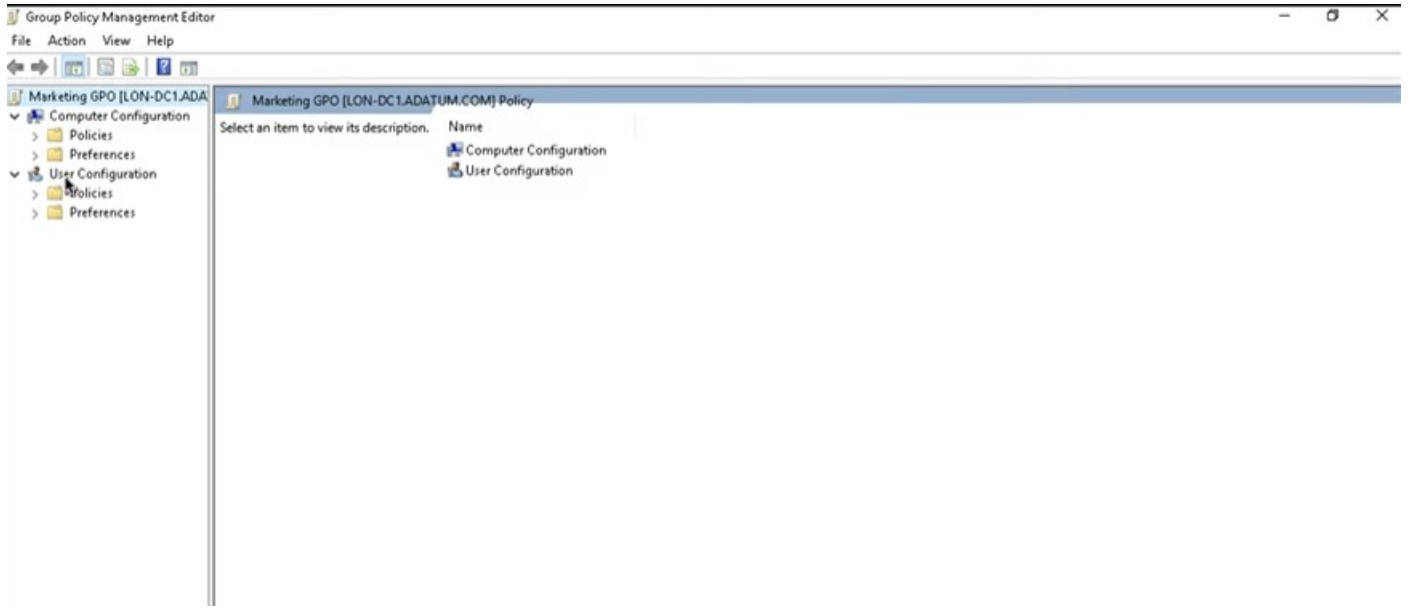
Leveraging Users and Groups

- In many cases, user permissions are used to provide your system with greater security without any direct interaction. Many operating systems create specific system user accounts for different packages during the installation process.
- The best practice is to give each user their own login to your system. This protects each user's files from all other users.
- Furthermore, using specific accounts for users allows more accurate system logging, particularly when combined with tools like sudo. We recommend avoiding situations where more than one individual knows the password for a user account for maximum security.
- In contrast, groups are useful for allowing multiple independent user accounts to collaborate and share files.
- If you create groups on a machine for common tasks on a per-task basis (e.g. web editors, contributors, content submitters, support) and add relevant users to the relevant groups, these users can all edit and run the same set of files without sharing these files with the world.
- Use of the chown command with file permissions of 770 and 740 would help accomplish this goal.

- **Ref: https://www.linode.com/docs/guides/linux-users-and-groups**



- **Ref:** **https://www.linode.com/docs/guides/linux-users-and-groups**
- **Error:**How to fix the specified server cannot perform the **requested** operation. - YouTube
- The specified server cannot perform the requested operation | ManageEngine Endpoint Central
- **Error:** **FIX: The Sign-in method you're trying to use isn't allowed**
- The error "The Sign-in method you're trying to use isn't allowed. For more info, contact your network administrator", commonly appears when you try to log on using the "Guest" account to a Windows 10 PC, or to a Domain Controller with any other user than then Domain Administrator. The error appears, because by default you cannot sign in locally with any user that hasn't administrator permissions on a Domain Controller or to a Windows 10 PC . SOULUTION: Fix “The sign in method you’re trying to use isn’t allowed” - YouTube



The End

TASBIH KIFARAH

Ucapan doa pada akhir majlis:

سُبْحَانَكَ اللَّهُمَّ وَبِحَمْدِكَ

اَشْهَدُ اَنْ لاَ اِلَهَ اِلاَّ اَنْتَ

اَسْتَغْفِرُكَ وَاَتُوْبُ اِلَيْكَ

Maha Suci Engkau, ya Allah, dan dengan memuji Mu,
aku bersaksi bahawa tiada Tuhan yang berhak disembah melainkan Engkau,
aku meminta ampun dan bertaubat kepada Mu