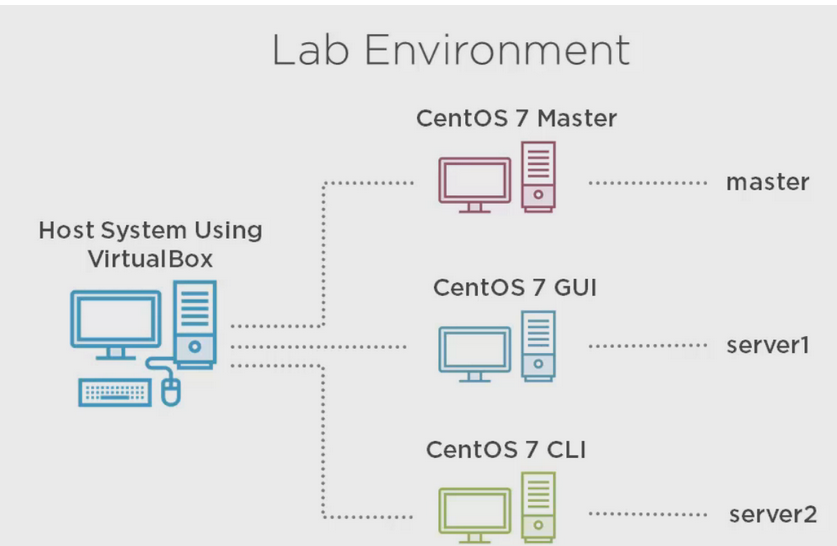
**Excercise Notes Zip**

[[https://www.evernote.com/images/file-generic.png](https://www.evernote.com/shard/s555/res/0ebff42c-fe98-4261-b602-4ea942e21135/lfcs-linux-user-group-management.zip)**lfcs-linux-user-group-managem...**2.6 MB](https://www.evernote.com/shard/s555/res/0ebff42c-fe98-4261-b602-4ea942e21135/lfcs-linux-user-group-management.zip)



PluralSight - LCFS Linux User and Group Management Course Notes

**Diagram of Lab Environment**

**Introduction to Linux User and Group Management and the Associated Certification Module**

Certification Track

1. LFCS Linux Essentials
2. LCFS Linux Operation Essentials
3. LCFS Linux User and Groups Management
4. LCFS Linux Storage management
5. LCFS Linux Networking
6. LCFS Linux Service Management
7. LCFS Linux Virtualization Management

Course Breakdown

Login Scripts

Managing Users

PAM

Managing Groups

OpenLDAP

Kerberos

Demo

Understanding the Command getent

**Listing Users and Services**

$ cat /etc/passwd

Contains the users and can contain the passwords

getent Defined

The  getent  command displays entries from databases supported by the Name Service Switch libraries, which are configured in /etc/nsswitch.conf. If one  or  more key arguments are provided, then only the entries that match the supplied keys will be displayed.  Otherwise, if no  key  is  provided, all  entries  will be displayed (unless the database does not support enumeration)

$ getent passwd

$ grep passwd /etc/nsswitch.conf

Verifying reference in the nsswitch configuration file from the password database

$ getent group

Listing local groups

$ getent networks

Listing networks

$getent services

Listing ports/services

**Managing Login Scripts Module Introduction**

Exam Objectives

Manage system-wide environment profiles

Manage template user environment

Demo

Login shells and non-login-shells

System login scripts

User login scripts and /etc/skel

**Login Scripts in CentOS7**

Login Shell

SSH or login console

Look for /etc/profile login script

/etc/profile

~/.bash\_profile

Call the ~/.bashrc file

/etc/bashrc

Non-login Shell

Executing bash by itself

~/.bashrc

Execute ~/.bashrc from home directory

/etc/bashrc

System bashrc file

**Investigating the Execution Order**

$ su -

Switching to the root account

short for "su -l"

If no user is specified, switch to root account

After successful password, directs to root directory

Similar to using "su" except routes to last users directory

Unless specified, always switches to root account

# echo $USER

# echo id

Both above commands will display root info

# exit

logs out of current user

Executes the .bash\_logout

# su

Will provide root privileges except maintains current home directory

# echo $USER

The user will display the **last user** instead of root

Anything using this variable will not function correctly as the user will not be the same

# cd

Ensure that you are in the root users home directory

# vi .bashrc

File is used for login and non-login shells

Enter in  a new line at the end of the file

echo "from bash"

# vi .bash\_profile

Enter in  a new line at the end of the file

echo "profile"

# exit

exit out of root as the dansaf user

# su

After entering the password, will be prompted "from bash"

# exit

# su -

After entering the password, will be prompted "from bash profile"

There is a .bash\_logout file that handles the logout process which may contain history information

**System Login Scripts**

# cd /etc

# ls profile\*

There is a profile file and a profile.d directory

profile.d Directory

Becomes an easy drop in directory for shell scripts that can be executed along with the /etc/profile file

Install of additional products may also add files in this directory to ensure ordered execution on startup

# ls /etc/bash\*

bashrc file is located in this directory

bashrc

This file will be executed multiple times if shells are executed. APpending to that path is best to be done in the bash\_profile

Aliases being set can be done in both files as this does not endanger the system

# echo $PS1

Display the order of PS1

PS1 Defined

PS1 (Prompt String 1) is one of the prompts available in Linux/Unix. When you try to login to any machine, you have to enter user name and password. Once you are done with this you are presented with some info like who logged in, on what machine he logged in, what is his present working directory and if the logged in user is a super user or a normal user. This is done by using PS1 prompt which is a inbuilt shell variable. The other prompts are PS2, PS3 and PS4

# su -

# vi /etc/bashrc

Navigate to the line with "PS1=" and replace the uppercase to lowercase for "W"

Ensure the change is made to the active line and not the commented

This will allow the display of the entire path navigated to instead of latest as set by default

# su -

# cd /var/tmp

After routing to the directory, the change should already be in effect

**Home Directory Templates**

# cd /etc/skel

Location of the skel files

This is the default directory to create default home directories for **new users only**

# ls -A

View the hidden files

# vi .bashrc

This command will open the .bashrc file from the skel directory. Any change on this file will reflect on new users only

Change "./etc/bashrc" to "source/etc/bashrc"

the "." and "source" are equivalent. This is for testing purposes, save and exit

# vi bash\_profile

Make the same change in this file and save, close

**Managing Login Scripts Module Recap**

Shells

Login

- /etc/profile, ~/.bash\_profile, ~./bashrc, /etc/bashrc

Non-Login

- ~/.bashrc, /etc/bashrc

What Goes Where

Profile

- PATH=$PATH:~/bin

Adding in the local bin directory

- export PATH

Bashrc

- PS1="[\u@\h\w] \$"

Set for each and every bash shell

/etc/skel

Template directory for NEW users

Sample personal login scripts

Source and . commands

**Creating and Managing Local Users in CentOS 7 Module Introduction**

Exam Objectives

Create, delete, and modify local user accounts

Demo

Linux user identification

Create local users

Manage user passwords

Working with user accounts

Modify and delete user accounts

**id Command**

$ id

Display current user information

$ id root

Display root user information

CentOS and Red Hat systems use private groups

First user and groups created will always default to 1000

Any user who is added as an administrator will automatically be added to wheel group

$ id -g

List primary group id

-G switch will display secondary groups

-Gn switch will also add names

**Creating User Accounts**

Local or domain based accounts

$ id -Gn

List group id, names including secondary

useradd Defined

When invoked without the -D option, the useradd command creates a new user account using the values specified on the command line plus the default values from the system. Depending on command line options, the useradd command will update system files and may also create the new user's home directory and copy initial files. By default, a group will also be created for the new user

$ sudo useradd -m user1

Add username with default settings

-m switch is not necessary as it is default in centOS

$ tail -n 1 /etc/passwd

This will list the last entry in the passwd file which confirms the last user created

user1:x:1001:1001::/home/user1:/bin/bash

There are 7 fields in the useradd process

Username

Password - "x" means that it is stored in a shadow file

User id

Group id

Comments

User home directory path

Default shell path

$ ls -a /home/user1

Cannot access, no permission

$ sudo ls -a /home/user1 will work

$ sudo useradd -N user2 -g users -G adm

Add another user without the -m switch

-g and -G switches will also add user to groups

$ tail -n 1 /etc/passwd

Will give nearly same results except the group permission is different

$ ls /home

Display the home directories including the created users without the -m switch

$ sudo useradd user3 -G adm -s /bin/sh

This command will add the user to the adm group and the path /bin/sh

ls -l /sbin/adduser

in CentOS and Red Hat, this location is only a symbolic link. In other systems, pearl script

**Managing User Passwords**

$ sudo passwd user1

This command will prompt the change password process. This process is similar to the UI and will respond similarly

$ sudo grep user1 /etc/shadow

Provide the password key specific to user1

$ sudo grep user. /etc/shadow

Will provide the passwords for any "user" single character

user2:!!:16979:0:99999:7:::

user3:!!:16979:0:99999:7:::

Each fiwld is marked with a preceeding ":"

Any password keys that have 1 or 2 exclamation marks in the second field are invalid passwords

These password keys are also referred to as shadow data

Third field is the number of days after January 1970

Fourth field is a optional setting for required amount of days to keep password

Fifth field is the amount of time before changing the password

$ echo 'user2:Password' | sudo chpasswd

Will send the STDOUT of the user2 user and it's password which is Password

$ sudo grep user. /etc/shadow

The output of the command will include a password added for the user2 user

$ echo Password | sudo passwd user3 --stdin

Is similar command as above except this is only compatible with Red Hat and CentOS systems

**Password Age Data**

chage Defined

The chage command changes the number of days between password changes and the date of the last password change. This information is used by the system to determine when a user must change his/her password

$ chage -l dansaf

List information regarding the password age and notification for the user dansaf

$ grep user1 /etc/passwd

This will list the information except the password field is marked with an "x" This means that the password is stored with the shadow data instead of in the passwd

pwunconv Defined

The pwunconv command creates passwd from passwd and shadow and then removes shadow

$ sudo pwunconv

All passwords that were stored in shadow file has been moved to passwd

$ grep user1 /etc/passwd

Now the data for user1 has changed for the passwd field. Instead of "x" has been replaced with the passwd field and password is stored

This key is an encryption derived and will require sudo privileges to access it

The shadow file is used to keep the access to just the user that way the real code is only visible to user and root

$ sudo pwconv

This will bring all users back to the shadow file

$ grep user1 /etc/passwd

This will verify that the listed information displays the "x" for passwd field

$ passwd --help

List the several switches that can be used

$ chage --help

List the several switches that can be used

$ sudo chage -M 40 user1

Modify the max days using the -M switch will change the maximum days for password to 40 for user1

Root privileges are required

$ sudo chage -l user1

This will verify that the maximum number of days has been changed to 40

$ sudo grep user1 /etc/shadow

Verify that the maximum number days result is 4 in the shadow file

$ sudo passwd -l user1

Lock the password to user1

$  sudo grep user1 /etc/shadow

The list will display an "!" for password and passwd indicating invalid

$ sudo passwd -u user1

This will unlock the password for user1

$ sudo grep user1 /etc/shadow

Display the password and passwd field and verify that the passwd is not "!"

**Account Defaults**

$ less /etc/login.defs

List the login defaults configuration file

PASS\_MAX\_DAYS is a field that stores the max number of days default value

UID\_MIN stores the minimum value of the user id

Several other related values are stored in this file

useradd Defined

When invoked without the -D option, the useradd command creates a new user account using the values specified on the command line plus the default values from the system. Depending on command line options, the useradd command will update system files and may also create the new user's home directory and copy initial files

$ sudo useradd -D

Requires sudo privileges  
List some of the defaults when adding a user

$ sudo useradd -Ds /bin/sh

Change the default shell path to the indicated, in this case /bin/sh

-s switch is for shell

$ sudo useradd -D

Verifies that the default has been changed to /bin/sh

$ sudo cat /etc/default/useradd

Display contents of the useradd defaults file

Possible to write directly to this file as long as the user has the ability to use sudo

$ sudo vi /etc/default/useradd

**Modify and Delete User Account**s

usermod Defined

The usermod command modifies the system account files to reflect the changes that are specified on the command line

$ sudo usermod -c "User One"user1

-c switch adjusts the full name field

Space character is a delimiter and requires quotes

Requires root privileges

$ grep user1 /etc/passwd

Verifies the full name has been changed to "User One"

chsh Defined

chsh  is  used to change your login shell.  If a shell is not given on the command line, chsh prompts for one

$ chsh -l

List the available shells

$ chsh -s /bin/sh dansaf

Changes the shell for dansaf user to /bin/sh

$ grep dansaf /etc/passwd

Verifies the field has in fact been changed

$ sudo usermod -s /bin/bash dansaf

Command can also make the change for the shell except requires sudo

Reverted the change back to default for shell

Command is mostly used as opposed to teh above for system administrators

$ grep dansaf /etc/passwd

Displays the change has been made

userdel Defined

The userdel command modifies the system account files, deleting all entries that refer to the user name LOGIN. The named user must exist

$ sudo userdel -r user1

The -r switch removes user's home directory, mail spool, and other related files for user1

$ ls /home

Verifies the user1 user's home directory has been deleted

$ sudo userdel user2

Without switches only removes the user, other related information remains

$ ls /home

Verifies that the directory remains

$ ls -l /home

Displays the long listing and the user id instead of the username as it has been deleted for the user2

$ sudo find /home -uid 1002 -delete

Easier way to scrub the file system for the user2 user as the user id is listed with the -uid switch

Searches for files that were also owned by the user by uid

-delete switch will delete the user2's files and directory and relevant info

**Creating and Managing Local Users in CentOS 7 Module Recap**

User ID

$ id

Every user has a user id and a group id

Information can be displayed using this command

No switches will display current id

$ id dansaf

Display information for dansaf user

$ id -G

-G switch displays secondary group id numbers

$ id -Gn

-n switch will display name

-g switch will display primary group information

Creating Users

/etc/passwd

Database for local user accounts

User Name information stored here

$ sudo useradd bob

Create user account with defaults using login name bob

user name and password are case sensitive

$ tail -n 1 /etc/passwd

List last line and previous user added

$ sudo useradd -N -g users -G adm sally

-N switch indicates not to create the private group

-g add the user to the users group

-G switch adds the user to the adm secondary group

Managing Passwords

/etc/shadow

Generated passwords and aging information stored here

Aging information

Only root can rerad from the shadow file

pwconv / pwunconv

If required, converts shadow paswords and back again

passwd

Set passwords, lock and unlock user account

chpasswd

Change passwords

chage

Manage password aging specifically

User Defaults

/etc/login.defs

Where the login default file is stored

/etc/default/useradd

Other settings are stored here

$ sudo useradd -D

-D display the defaults

Other switches can set the defaults

Modify Accounts

$ chsh -l

Lists shells for user

$ chsh -s /bin/sh dansaf

Set shell for dansaf user

$ sudo usermod -s /bin/bash dansaf

This command will also do the same

-s switch

$ sudo usermod -c "User One" user1

-c switch changes the full name field of user

Delete Accounts

$ sudo userdel -r user1

Delete the user and mail, cron jobs and home directory

$ sudo find / -uid 1001 user1 -delete

Searches for user by id and deletes all relevant information

**Managing Local Groups in CentOS 7 Module Introduction**

Exam Objectives

Create, delete and modify local groups and group memberships

Configure set-GID directories for collaboration

Demo

Create local groups

Modify groups memberships

set GID permission on directory

Group Passwords

**Creating Groups**

$ grep dansaf /etc/group

Lists the dansaf user from the group file

Red Hat and CentOS systems use private groups by default

$ id

Lists all id's associated to current user including groups

newgrp Defined

The newgrp command is used to change the current group ID during a login session. If the optional - flag is given, the user's environment will be reinitialized as though the user had logged in, otherwise the current environment, including current working directory, remains unchanged

$ newgrp wheel

Changes the group ID from current to wheel

$ touch g1

Creating a test file

$ ls -l g1

Listing properties. Verifies that the group is wheel

$ exit

Exit out of the wheel group

$ touch g2

$ ls -l g2

Verifies that the group has been reverted back to dansaf

This process does not affect permissions, only labeling

groupadd Defined

The groupadd command creates a new group account using the values specified on the command line plus the default values from the system. The new group will be entered into the system files as needed

$ sudo groupadd sales

Creates a group named sales with defaults

$ grep sales /etc/group

List the sales group details

$ sudo grep sales /etc/gshadow

List the shadow information for the group sales

Password is listed with an "!" invalid password

Generally groups do not have passwords

groupmod and groupdel are also used to modify group information

**Manage Group Membership**

$ id -G

$ id -Gn

Checking secondary group name and id

$ id -gn

Checking primary group name

$ sudo usermod -G sales,wheel dansaf

Add dansaf user to newly created sales group with the wheel group as secondary

This is looking at group membership from user perspective

$ sudo usermod -G sales dansaf

This command will adjust the user to only be added to the sales as a secondary group

This is looking at group membership from user perspective

gpasswd Defined

The gpasswd command is used to administer /etc/group, and /etc/gshadow. Every group can have administrators, members and a password

$ sudo gpasswd -a dansaf sales

Add user dansaf to sales group

$ sudo gpasswd -M dansaf,root sales

-M switch can add multiple users with the colon as a delimiter

dansaf and root were added to sales group

$ gpasswd --help

There are several switches that can be used

$ grep dansaf /etc/group

List the groups that are associated with dansaf

$ id

This confirms the session hasn't changed yet and requires the user log out and back in for changes to take effect

$ su - dansaf

This command will create a whole new shell which is similar outcome as logging out and back in

$ id and id -Gn

The results should depict the adjustment of adding sales group

**Making Use of the SGID Permission**

$ sudo yum install httpd w3m

Install prerequisites before working with group permissions

w3m is a command line browser based

$ sudo systemctl status httpd

If service has installed successfully, should result in active (running)

$ sudo systemctl start httpd

Execute the command if the status displays an issue

$ ls /var/www/html/

Data location for httpd service

$ ls -ld /var/www/html

Lists permissions and results in both user and group set to root

Web server is reliant on group permissions given to others

$ sudo vi /var/www/html/index.html

Welcome

Save and exit

$ w3m localhost

This will execute w3m with the localhost to open the index.html file

$ grep apache /etc/group

This provides the permissions for Apache and will result in

chgrp Defined

Change  the  group  of  each  FILE to GROUP.  With --reference, change the group of each FILE to that of RFILE

$ sudo chgrp -R apache /var/www

Changing the group membership recursively in www directory

$ ls -ld /var/www

This verifies that the secondary group membership has been set to apache

$ sudo chmod -R o= /var/www

Take away privileges that are being granted to others using "" for others recursively

$ ls -ld /var/www

Verify that the permissions have been taken away

$ sudo ls -l /var/www/html

Verifies Apache has read rights to the index.html page

$ w3m localhost

This verifies apache has access to the

$ sudo -i

$ cd /var/www/html

$ ls -ld

Results in the ownership remains for apache

$ chmod g+s

Set group and the special permissions using the g and s switches on directory

$ ls -ld

Results have special permissions applied and indicates group id set

$ umask 027

Keeps permissions away from the others group

$ vi test.html

**Group Passwords**

$ id -Gn

-G secondary group switch

-n Displays names instead of ID numbers

$ id -gn

-g primary group switch

-n name switch

$ newgrp adm

Create a new group with the name adm

Prompted password for group

Always result in invalid as the password hasn't been created yet

$ sudo gpasswd adm

Prompted to enter user credentials

Will prompt to enter new password

$ newgrp adm

Enter Password for group that was previously created

$ id -gn

The group will now display as adm

$ id -Gn

Group will also be included in secondary groups

**Managing Local Groups in CentOS 7 Module Recap**

Groups

/etc/group

Location of group configurations

groupadd

groupdel

groupmod

Less oftern used

Group Membership

$ sudo gpasswd -a bob sales

Add bob to sales

$ sudo gpasswd -M bob,joe,tux sales

Replace complete membership

$ sudo usermod -G sales,wheel bob

Secondary group memberships

Set-GID Permission

Controls ownership of new files created in the directory

Set group ID

$ sudo chgrp -R apache /var/www

Set ownership to apache

$ sudo chmod -R o= /var/www

Recursively remove

$ sudo g+s /var/www/html

Ensure all new files are created and group owned by directory

Group Passwords

/etc/gshadow

Lessens security if group has password

**Using PAM to Control User Access Module Introduction**

Exam Objectives

Configure PAM

Configure user resource limits

Manage user processes

Demo

Creating home directories during login

Configure password policies

Restrict consecutive logins and resource access

Control login times

**Configure PAM**

PAM defined

Pluggable Authentication Modules provide dynamic authentication support for applications and services in a Linux or GNU/kFreeBSD system. Linux PAM is evolved from the Unix Pluggable Authentication Modules architecture

User Login

SSH

Console

GUI

PAM

Authentication

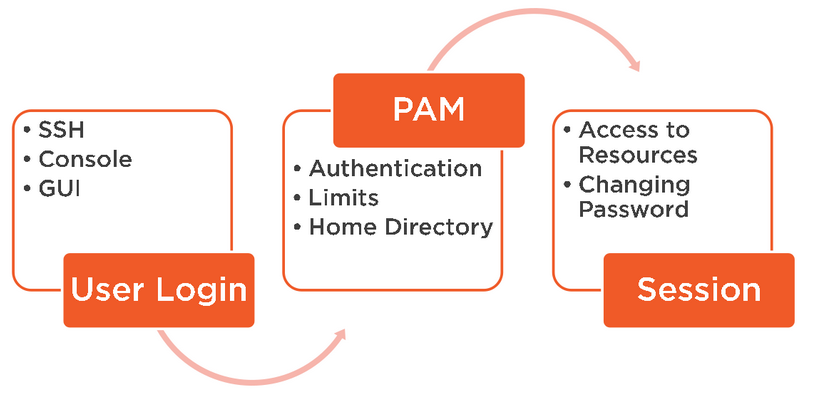
Limits

Home Directory

Session

Access to Resources

Changing Password



**Create Home Directories at Login**

$ ls /etc/pam.d/

Combined listing of configurations for all apps that can use authentication

List will make reference to PAM modules

$ ls /lib64/security/

Consists of several shared libraries that PAM will use

$ ls /etc/security

Configure some of the PAM modules

limits.conf

Configure limits of the amount of configurations

pwquality.conf

Configure password policies

All three directories work in conjunction with each other to provide sure access

$ sudo vi /etc/login.defs

Location of the default login configurations

CREATE\_HOME no

Change the configuration of the line to "no" save and exit

This will not create a home directory when the useradd is executed

Particularly usefule if useradd is done in batch and would like to thin out the resource on one shot

$ sudo useradd bob

Add bob user

$ sudo password bob

Displays the change password prompt for bob

$ rpm -qa | grep oddjob

Ensuring the oddjob module has been installed

Should display the package manager and the PAM module location

oddjob Defined

oddjob is a D-BUS service which performs particular tasks for clients which connect to it and issue requests using the system-wide message bus

$ systemctl enable oddjobd

Enabling service at boot-up

$ systemctl start oddjobd

Start the oddjob service right away

Not required to use sudo

$ systemctl status oddjobd

Checks whether the service is up and running

$ sudo authconfig --enablemkhomedir --update

$ sudo -i

# cd /etc/pam.d

# grep mkhomedir \*

See all mkhomedir files in the pam.d directory

References in several authentication and all have reference to the oddjob modules

# su - bob

This will automatically create a home directory when logged in as bob

This demonstrates created staged home directories through pluggable modules

**Implementing Password Policies**

$ cat /etc/pam.d/system-auth

Lists several configurations for user password processes as well as options for the module

All stored in the /lib64/security

$ less /etc/security/pwquality.conf

Look at the policy

Security directory, lists authentication for the modules

difok = 5

This configuration determines how many characters need to be different when the password is changed

$ pwscore

This command will check the input to determine whether the password meets the standards set

If passed the dictionary checks, the resulting number determines the quality of password. Higher the number, the better the security, 100 is strong

**Restricting or Limiting Access to Resources**

ulimit defined

ulimit builtin sets or outputs the resource usage limits of the shell and any processes spawned by it. If a new limit value is omitted, the current value of the limit of the resource is printed; otherwise, the specified limit is set to the new value.

$ ulimit -a

List all of the restrictions in place

$ ulimit -u

Lists the limit for amount user processes

$ ulimit -u 10

Adjust the setting to 10

$ ulimit -u

Result in 10

$ vi test.sh

#! /bin/bash

echo "test"

$0

$ chmod +x test.sh

Allow execution on file

$ ./test.sh

Execute script and at some point will fail as only max 10 processes can be at one time

$ rm ./test.sh

Remove test file

$ ps

Verify whether or not the process has stopped

$ sudo vi /etc/security/limits.conf

Resource file for configuring limits for users logged in

"\*" is for all users

"@" for groups

Replace the #@student with a \* to indicate the max logins for everyone is 4

\* - maxlogins 4

Add hard and soft limits for the max number of processes

@users soft nproc 50

Sets a soft limit for the number of logins to 50 which is the max logins

@users hard nproc 75

Sets a hard limit for the number of max logins that can be configured to 75

Exit and SSH back to test

$ ulimit -u

Output will result in the adjustment of 50

$ ulimit -u 70

$ ulimit -u

Output will result in 70 instead of 50 with this quick command.

Hard setting is 75 and can be adjusted until that setting

$ ulimit -u 80

Result in error cannot modify limit

**Adding Login Time Restrictions**

$ cd /etc/pam.d/

Manage programs that use authentication pluggable modules

Restrict access to SSH would require editing the sshd file

If the file needs to be accessed on multiple systems, should be in a shared location

$ sudo vi sshd

Insert a new line above the first account entry

count required pam\_time.so

Account restriction is passed and checked by the library file pam\_time.so

By itself, will not process correctly

$ cd /etc/security/

$ sudo vi time.conf

In new line add the command

\*;\*;dansaf|bob;Wk0800-1800

Interpreting this for all shells, shell types, user dansaf and bob access during the week mon-fri from 8am-6pm

Duplicating a ssh session should be OK with this setting. But, if the command had included "!" to indicate not to allow during the times indicated like below and duplicating will result in restriction

\*;\*;dansaf|bob;!Wk0800-1800

**Using PAM to Control User Access Module Recap**

Home Directories

/etc/login.defs

Modifying the default setting in the file

# yum install oddjob-mkhomedir

Installation path for oddjob daemon

# systemctl start oddjobd

Start the service

# systemctl enable oddjobd

Set to enable during boot

# authconfig --enablemkhomedir --update

Write to PAM configuration, easy way to configure once installed

Password Policies

$ pwscore

Command to check score for password

$ /etc/security/pwquality.conf

FIle that has configurations

ulimit

$ ulimit -a

List all effective limitations

$ /etc/security/limits.conf

File that has configuration

Hard and soft limits

Restrict and Limit Access Times

/etc/pam.d/sshd

Location of the sshd configuration that can use PAM

account required pam\_time.so

Account restrictiolives in lib64 in security

/etc/security/time.conf

implement the restriction in this file

sshd;\*;dansaf|bob;Wk0800-1800

This is command for users dansaf and bob for the mon-fri 8am-6pm

**Implement OpenLDAP Directories on CentOS 7 Module Introduction**

Exam Objectives

LDAP client configurations (Setting up the LDAP Server is not an objective but will need the LDAP server to set up LDAP client configurations)

Demo

Install OpenLDAP  
Configure OpenLDAP server

Create directory structure

Create groups and users

**Install OpenLDAP and Firewall COnfiguration**

Installing the LDAP server on server1

server1 is fried on my vm, had to use master

$ hostname

Verify the server information

$ su -

Login as root

# ip a s

Verifying IP information and that the enps08 is up and the interface for the connection to server2 machine

# echo "192.168.56.102 server1.example.com" >> /etc/hosts

Appending the necessary information for LDAP in the /etc/hosts. This includes the ip address and hostname

# ping server1.example.com

This verifies that the server1 hostname is set to the ip address

# netstat -ltn

Check listening ports

OpenLDAP will listen on port 389 or 636 for ssl

By default, firewall is enabled

firewall-cmd defined

firewall-cmd is the command line client of the firewalld daemon. It provides interface to manage runtime and permanent configuration

# firewall-cmd --permanent --add-service=ldap

Add the ldap service

# firewall-cmd --reload

Reload the firewall service daemon for the changes to take effect

#firewall-cmd --list-all

List the firewall services

# yum install -y openldap openldap-clients openldap-servers migrationtools.noarch

Install all requires packages and dependencies including migration tools to migrate password files to LDAP directory

**Configuing OpenLDAP**

# cp /usr/share/openldap-servers/DB\_CONFIG.example /var/lib/ldap/DB\_CONFIG

Copy backup of standard configuration and rename copy to DB\_CONFIG

# ls -l /var/lib/ldap

Verify the file

# slaptest

This will present errors at the moment. This will create the default files

# ls -l /var/lib/ldap

There will be several others files present including the database files

# chown ldap.ldap /var/lib/ldap/\*

check to ensure all files listed in directory are owned by ldap for permissions

# systemctl start slapd

Start slapd service

# systemctl enable slapd

Enable service at boot

# systemctl status slapd

Double check for status

# netstat -ltn & netstat -lt

Display that the ldap is configured for both the tcp ports

# cd /etc/openldap/schema/

Configuring the schema files which controls what can be created in the service

ldapp defined

ldapmodify, ldapadd   is  a  shell-accessible  interface  to  the  ldap\_add\_ext(3), ldap\_modify\_ext(3), ldap\_delete\_ext(3) and ldap\_rename(3).  library calls. ldapadd  is  implemented  as  a  hard  link  to the ldapmodify tool.  When invoked as ldapadd the -a (add new entry) flag is turned on automatically

# ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f cosine.ldif

Configuration to add an external ldap process with specified parameters for cosine.ldif

# ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f nis.ldif

Configuration to add an external ldap process with specified parameters for nis.ldif

# slappasswd -s Password1 -n > rootpwd

Create an encrypted password for administrators directory

# cat rootpwd

Should display the hash of Password1

# sudo vi config.ldif

Create a config.ldif in users home directory

Use the exercise file config.ldif

When the excercise file details has been copied, navigate to olcRootPW and read from the rootpwd

:r rootpwd file

# ldapmodify -Y EXTERNAL -H ldapi::/// -f config.ldif

Modify using the quick command ldapmodify and read in the file that was just created

Any errors that occur are due to the copy and paste process, be careful!

**Creating the Directory Tree**

Copy Exercise file in the home directory as structure.ldif and save and close

the details are also configured in the configure.ldif

# ldapadd -x -W -D "cn=manager,dc=example,dc=com" -f structure.ldif

Enter the password that was configured in the configure.ldif (Password1) No resulting error leads to success

#ldapsearch -x -W -D "cn=manager,dc=example,dc=com" -b "dc=example,dc=com" -s sub "(objectclass=organizationalunit)"

Enter the same password and will display all the search results for the base of the example and com and for organizationalunit

**Create Groups and Users**

Copy Excercise file in the home directory as group.ldif, save and close

# ldapadd -x -W -D "cn=manager,dc=example,dc=com" -f group.ldif

This command will create and configure the group ldapusers

Add users using the migrationtools package that was installed

**Implement OpenLDAP Directories on CentOS 7 Module Recap**

Install

# yum install openldap-clients openldap-servers migrationtools

# firewall-cmd --permanent --add-service=ldap

Added LDAP service to firewall settings

# firewall-cmd --reload

Once added, restart is required

Configure

# cp /usr/share/openldap-servers/DB\_CONFIG.example /var/lib/ldap/DB\_CONFIG

Replacing file

# slaptest

Generate database files using this command

# chown ldap.ldap /var/lib/ldap/\*

Change ownerships of the files used

# systemctl start slapd

# systemctl enable slapd

for autostart

# ldapadd -Y EXTERNAL -H ldapi:/// -D "cn=config" -f /etc/openldap/schema/cosine.ldif

Creating schema processes which define what can be added to structure

# ldapmodify -Y EXTERNAL -H ldapi:/// -f /root/config.ldif

Need to be root

Structure

# ldapadd -x -W -D "cn=Manager,dc=example,dc=com" -f /root/structure.ldif

Define upper levels of authenticating tree

Users and Groups

# vi /usr/share/migrationtools/migrate\_common.ph

Create a template user, then update the file to ensure the domain and set are correct

# migrate\_passwd.pl passwd users.ldif

**Implementing OpenLDAP Authentication in CentOS 7 Module Introduction**

Exam Objectives

Configure a client to use LDAP for user and group information

Demo

Install and configure OpenLDAP client

List users and groups

Search users and groups

**Installing the OpenLDAP Client**

This module requires two virtual machines. My Master is the server1 and I have a server2

# echo "192.168.56.101 master.example.com" >> /etc/hosts

As root, execute this command and use the IP of the host that has the OpenLDAP installed

# ping master.example.com

This verifies that there is a connection and the IP address matches

# yum install oddjob oddjob-mkhomedir.x86\_64

Install the necessary oddjob modules for client

# systemctl start oddjobd

start oddjob daemon

# systemctl enable oddjob

# systemctl status oddjob

# yum install openldap-clients.x86\_64 nss-pam-lpad

# authconfig-tui

LDAP authentication and path through to the LDAP server using menu system

Use tab Key to navigate to "Use LDAP Authentication" and check

Tab to Next, then in LDAP Settings window, tab until server and change to "master.example.com"

# authconfig --enableldap --ldapserver=master.example.com --ldapbasedn="dc=example,dc=com" --enablemkhomedir --update

This command essentially does the same thing as the command above except there is no UI and may be necessary if scripting is created

# getent passwd

# grep passwd /etc/nsswitch.conf

# su - fred

This is the user that is available on the LDAP directory. Once the user has been added to the LDAP server, should be able to extract it to this server

**List Users and Groups**

On server2 of my Virtual Machine

# getent passwd

List the current users including fred recently created

# getent group

List all the current groups including ldapusers group recently created and set up through name service switch file

# grep ldap /etc/nsswitch.conf

Verifies that the ldap has been enabled for password, shadow, group, netgroup, automount

Each of these options can be disabled if not in use

# vi /etc/nsswitch.conf

This is the file that needs to be changed to remove the options

Remove ldap from netgroup, automount, then save and close

# grep ldap /etc/nsswitch

This will result in only the passwd, shadow and group

# ssh [dansaf@192.168.56.102](mailto:dansaf@192.168.56.102)

Accept the key and enter password to verify that ssh is enabled for dansaf user and access to server1

# getent passwd

This will display the users where the newly created user is not listed because of ssh into server1

# getent group

Verify that there is no ldapusers group for server1

**Searching LDAP users and groups**

# ldapsearch -x -H ldap://master.example.com -b dc=example,dc=com

Initiate an LDAP search for query on the LDAP server

Metadata, search response and number of responses

# ldapsearch -x -LLL -H ldap://master.example.com -b dc=example,dc=com

Reduce the metadata using -LLL switch, only data that was resulted from the query

# ldapsearch -x -LLL -H ldap://master.example.com -b dc=example,dc=com "(objectclass=account)"

Results with the posixAccount or shadowAccount specific

# ldapsearch -x -LLL -H ldap://master.example.com -b dc=example,dc=com "(&(objectclass=account) (uid=fred))"

Another common search command for multiple variables to search for

# ldapsearch -x -LLL -H ldap://master.example.com -b dc=example,dc=com "(objectclass=account)" uidNumber uid

This will limit to only the uid and uidNumber variables

# ldapsearch -x -LLL -H ldap://master.example.com -b dc=example,dc=com "(&(objectclass=account) (uid=fred))" > newuser.ldif

Send information to newuser.ldif

# vi newuser.ldif

Substitute the user from Fred to Sally using :%s/fred/sally/g

Change the uid number from 4000 to 4001

#ldapadd -x -W -D cn=manager,dc=example,dc=com -f newuser.ldif

Password =Password1

# getent passwd

Listing will include Sally the newly created user

# su sally

Will change the prompt to sally newly created user

**Implementing OpenLDAP Authentication in CentOS 7 Module Recap**

Install

# yum install openldap-clients nss-pam-ldapd

# yum install oddjob oddjob-mkhomedir

Creating home directory

# systemctl enable oddjobd

# systemctl start oddjobd

# authconfig-tui

Simple UI mechanism to setup authentication

List Users

# getent passwd

# getent group

$ sudo vi /etc/nsswitch.conf

Remove and add entries manually

LDAP Search

$ ldapsearch -x -LLL

-H ldap://master.example.com

-b dc=example,dc=com

$ ldapsearch -X -LLL

-H ldap://master.example.com

-b dc=example,dc=com

+(objectClass=account)"

**Implementing Kerberos Authentication Module Introduction**

Exam Objectives

Configure a system to authenticate using Kerberos

Demo

Configure NTP

Install and configure KDC

Enable SSH Kerberos authentication

Add additional Kerberos client

**Configure NTP**

NTP defined

NTP stands for Network Time Protocol, and it is an Internet protocol used to synchronize the clocks of computers to some time reference

# yum install -y ntp

Before Kerberos authentication can even be used, we must install the ntp service and daemon

# vi /etc/ntp.conf

Adjust the configuration in this file

Drift file will write to configuration file that will maintain the drift of other servers to keep in synch

Standard restrictions, default restrictions prevent specific modifications and network

Setup restrictions for the local network by un-commenting and adjusting the restrict line

This will prevent any signals and modifications of the server. Only setup of peer and synch services are allowed

restrict 192.168.56.0 mask 255.255.255.0 nomodify notrap

# systemctl enable ntpd

Set to start service on boot

# systemctl start ntpd

Start service

# systemcl status ntpd

Verify service

ntpq defined

The ntpq utility program is used to monitor NTP daemon ntpd operations and determine performance. It uses the standard NTP mode 6 control message formats defined in Appendix B of the NTPv3 specification RFC1305. The same formats are used in NTPv4, although some of the variable names have changed and new ones added. The description on this page is for the NTPv4 variables.  
The program can be run either in interactive mode or controlled using com- mand line arguments. Requests to read and write arbitrary variables can be assembled, with raw and pretty-printed output options being available. The ntpq can also obtain and print a list of peers in a common format by send- ing multiple queries to the server.  
If one or more request options is included on the command line when ntpq is executed, each of the requests will be sent to the NTP servers running on each of the hosts given as command line arguments, or on localhost by default. If no request options are given, ntpq will attempt to read com- mands from the standard input and execute these on the NTP server running on the first host given on the command line, again defaulting to localhost when no other host is specified. ntpq will prompt for commands if the standard input is a terminal device.  
ntpq uses NTP mode 6 packets to communicate with the NTP server, and hence can be used to query any compatible server on the network which permits it. Note that since NTP is a UDP protocol this communication will be some- what unreliable, especially over large distances in terms of network topology. ntpq makes one attempt to retransmit requests, and will time requests out if the remote host is not heard from within a suitable time- out time.  
Note that in contexts where a host name is expected, a -4 qualifier pre- ceding the host name forces DNS resolution to the IPv4 namespace, while a -6 qualifier forces DNS resolution to the IPv6 namespace

# ntpq -p

Lists peers known to the ntp server

# firewall-cmd --add-service=ntp --permanent

This command will add the ntp service to list of allowed services through the firewall

# firewall-cmd --reload

For the change to take effect, the firewall must be reloaded

# vi /etc/hosts

Allow the other servers/vm's access by including their IP in the list with the current

192.168.56.104 server2 server2.example.com

# ssh dansaf@server2

Make the same inclusions in the servers associated

# yum install -y ntp

# vi /etc/ntp.conf

Remove delete the listed pools except below

server 3.centos.pool.ntp.org iburst

Add pool

server master.example.com iburst prefer

iburst allows for quick synchronization

prefer being the preferred server

No necessary changes to firewall setting for this machine as the connections are being made on the other

# systemctl enable ntpd

# systemctl start ntpd

# systemcl status ntpd

**Configuring the Kerberos Services**

# yum install -y rng-tools.x86\_64

Install random number generator for generating keys

# systemctl start rngd

# systemctl enable rngd

# systemctl status rngd

Start and verify service. Most likely fails as there is configuration need to be done

# vi /usr/lib/systemd/system/rngd.service

Set device that will be initialized, adjust the line to include additional

ExecStart=/sbin/rngd/ -f -r /dev/urandom

# systemctl start rngd or # systemctl restart rngd

# systemctl daemon-reload

May need to reload all daemons

# systemctl status rngd

Should now include urandom in command

# yum install -y krb5-server krb5-workstation pam\_krb5

Install Kerberos server, utilities and PAM authentication

# cd /var/kerberos/krb5kdc/

List out the two config files associated

# cat kadm5.acl

This will display the admin rights associated to the [admin@EXAMPLE.COM](mailto:admin@EXAMPLE.COM) role which is the same as the realm

# cat kdc.conf

List the listening ports and the realm EXAMPLE.COM

default\_realm = EXAMPLE.COM

un-comment the default realm

[realms]

EXAMPLE.COM = {  
kdc = master.example.com

admin\_server = master.example.com

}

Define the realm by un-commenting and changing the realm block

admin server is the database server

kdc server is the key distribution servers

[domain\_realm

.example.com = EXAMPLE.COM

example.com = EXAMPLE.COM

# kdb5\_util create -s -r EXAMPLE.COM

Configure and setup the realm EXAMPLE.COM

Prompted to enter KDC database master key

Password1 was entered for password in this instance

# ls

Will display more databases in the krb5kdc directory that was created from executing the above command

# systemctl start krb5kdc kadmin

# systemctl enable krb5kdc kadmin

start and enable the required database and key servers

# systemctl status krb5kdc kadmin

Verify they are running

**Adding Kerberos Principles**

# netstat -lt

Continuing configuration of Kerberos server and double check firewall settings and verify the addition of kerberos-adm kpasswd and kerberos

-lt switches listing the listening ports and which processes

# netstat -ltn

-n switch verifies the addresses without ports

# firewall-cmd --add-service=kpasswd --permanent

Add kpasswd service to firewall allow list

# firewall-cmd --add-service=kerberos --permanent

Add Kerberos service to firewall allow list

# firewall-cmd --add-port=749/tcp --permanent

Add port 749 to firewall allow list

# firewall-cmd --reload

Reload the firewall to have updated configurations included

# kadmin.local

Authenticate as root user in Kerberos

kadmin.local: listprincs

List of principles available

kadmin.local: addprinc root/admin

Add principal root user to list

Password was set to Password1

kadmin.local: addprinc dansaf

User to be able to add and pick up Kerberos tokens

Password set to Password1

kadmin.local: addprinc -randkey host/master.example.com

Add principle for local host

kadmin.local: listprincs

List principles including newly created

kadmin.local: ktadd host/master.example.com

Add in a key tab file and copy it across

kadmin.local: quit

Quit session to return to standard root shell prompt

Added necessary users to list including principles which concludes necessary configurations for server

**Enabling Kerberos Authentication**

# vi /etc/ssh/ssh\_config

Setup the ssh service with Kerberos authentication

GSSAPIAuthentication yes

GSSAPIDelegateCredentials yes

Un-comment and adjust the lines to yes, save and close file. This will allow to authenticate to server

ssh refers to client while sshd is the server

# systemctl reload sshd

Reload the daemon

# authconfig --enablekrb5 --update

Ensure the PAM modules can be used

# exit

Log out of sudo and root to user

$ klist

Currently no tokens are displayed

$ kinit

Route to default server and authenticate using principle password, in my case Password1

$ klist

The key will be displayed and can be used for other processes using Kerberos authentication for 24 hours

$ ssh master.example.com

After the primary key fingerprint has been established. This command will automatically route to master server

$ exit

$ klist

Key will still be available in the key ring

$ kdestroy

This removes the key from the klist

$ cat /etc/hosts

We must have a record of our host in the host file for this work

**Adding Additional Hosts**

# vi /etc/hosts

Add in the necessary servers including the server2

192.168.56.104 server2 server2.example.com

# yum install -y krb5-workstation pam

Install Kerberos utilites

# scp [dansaf@master.example.com:/etc/krb5.conf](mailto:dansaf@master.example.com:/etc/krb5.conf) /etc

Copy krb5.conf file from master server, this will require authentication

Agree to all prompts

# cat /etc/krb5.conf

Will result in the copy from the file in master server

# kadmin

Prmpted for password in my case Password1

kadmin: listprincs

List all the principles remotely

kadmin: addprinc -randkey host/server2.example.com

need to adjust for server2 included in list

kadmin: listprinc

List will include the server2 from list

kadmin: ktadd host/server2.example.com

Add key tab file for server2 to local file

kadmin: quit

Quit from session

# vi /etc/ssh/ssh\_config

Change ssh client configuration

GSSAPIAuthentication yes

GSSAPIDelegateCredentials yes

Un-comment and adjust the lines to yes, save and close file. This will allow to authenticate to server

# authconfig-tui

Enable authentication back to this system through UI

Tab through to Kerberos setting and enable with space bar then next until OK

authconfig can be used through command line or UI

# systemctl reload sshd

Verify ssh daemon is using Kerberos

# klist

Should not display anything at the moment

# exit

Exit out of root back to user dansaf

$ kinit

Initialize key for dansaf on server2 as Password1

$ klist

Key displayed and is ready for use

$ ssh [dansaf@master.example.com](mailto:dansaf@master.example.com)

$ exit

$ ssh [dansaf@server2.example.com](mailto:dansaf@server2.example.com)

Ability to log into both servers without prompting for password

Will route to the user without password prompt or ssh key authentication. If password is prompted than the process is broken somewhere

**Implementing Kerberos Authentication Module Recap**

Random Number Entropy

When generating encrypted data we will need entropy to create the key pairs

This was done on Master server in my demo but normally done on the server1

# yum install -y rng-tools

# systemctl enable rngd

This may not work right away but after executing teh below command and adjuting the config

# vi /usr/lib/systemd/system/rngd.service

ExecStart=/sbin/rngd -f -r /dev/urandom

# systemctl daemon-reload && systemctl start rngd  
Host Entries

We are using local hosts entries but DNS would work well in a larger environment. On both server1 and server2 we need the following entries in the /etc/hosts files

# vi /etc/hosts

192.168.56.105 server1.example.com

192.168.56.104 server2.example.com

NTP

Time should be synchronized on both servers

Server 1:

# yum install -y ntp

# vi /etc/ntp.conf

restrict 192.168.56.0 mask 255.255.255.0 nomodify notrap

# systemctl enable ntpd

# systemctl start ntpd

Server 2:

# yum install -y ntp

# vi /etc/ntp.conf

server server1.example.com iburst prefer

# systemctl enable ntpd

# systemctl start ntpd

Open Firewall on Server 1

# firewall-cmd --add-service={kpasswd,kerberos,ntp} \ --permanent

# firewall-cmd --add-port=749/tcp --permanent

# firewall-cmd --reload

Install Kerberos on Server 1

# yum install -y krb5-server krb5-workstation pam\_krb5

# vi /etc/krb5.conf

Uncomment default\_realm, [realms], [domain\_realms]

# kdb5\_util create -s -r EXAMPLE.COM

# systemctl start krb5kdc kadmin

# systemctl enable krb5kdc kadmin

Add Initial Principles on Server 1

# kadmin.local

addprinc root/admin

addprinc tux

addprinc -randkey host/server1.example.com

ktadd host/server1.example.com

quit

Kerberize SSH on Server 1

# vi /etc/ssh/ssh\_config

GSSAPIAuthentication yes

GSSAPIDelegateCredentials yes

# authconfig --enablekrb5 --update

# systemctl sshd reload

Authenticate as Dansaf

$ kinit

$ klist

$ ssh server1.example.com

Install Kerberos Client on Server 2

# yum install -y krb5-workstation pam\_krb5

# scp tux@server1.example.com:/etc/krb5.conf /etc

# kdamin

addprinc -randkey host/server2.example.com

ktadd host/server2.example.com

quit

Kerberize SSH on Server 2

# vi /etc/ssh/ssh\_config

GSSAPIAuthentication yes

GSSAPIDelegateCredentials yes

# authconfig --enablekrb5 --update

# systemctl sshd reload