**Lesson 8**

**User Authentication & the Pluggable Authentication Modules** (**PAM)**

**Contents**

[1. Passwords and Salt 2](#_Toc93834411)

[2. The effect of the pam\_nologin module 3](#_Toc93834412)

[3. Using pam\_permit to allow access without passwords 4](#_Toc93834413)

[4. Using pam\_listfile to control access to vsftpd service 4](#_Toc93834414)

[5. Using pam\_faillock to track login attempts 6](#_Toc93834415)

[6. Using pam\_time to restrict access to services based on time 10](#_Toc93834416)

[7. Configure httpd basic authentication via PAM (Optional) 13](#_Toc93834417)

[8. authselect and the nsswitch.conf 16](#_Toc93834418)

[9. Allocating admin tasks to normal users with sudo 20](#_Toc93834419)

1. Passwords and Salt

The following exercises are to validate your understanding of how Linux stores the user passwords.

There are three system files you need to know:

* /etc/password
  + Contains the user account information.
* /etc/shadow
  + Contains the user passwords using the hash values.
* /etc/group
  + Contains the user group information.

On the server:

1. Select any two users and set their passwords to be the same.
   1. For example: set the passwords of paul and peter to 'P@ssw0rd'.
2. View /etc/shadow and check that the hashed passwords for these two users are different. (or looks different.)
3. Given that the following 1 liner python command :

python -c 'import crypt; print(crypt.crypt("apple", "$1$OLto9FYs"))'

can generate a MD5 password hash of "apple" with the salt value "OLto9FYs" which is equivalent to the following openssl command.

echo "apple" | openssl passwd -stdin -1 -salt OLto9FYs

Timeline

Description automatically generated with low confidence

The output of the two commands is the same:

$1$OLto9FYs$aTBdegtcwu0WPtdTcuA0R.

1. Based on the following screen shot of /etc/shadow file\*:

Text

Description automatically generated

Can you determine which two accounts are using '1qw2' as their password ?

\*This segment of the shadow file is posted at BB too.

1. The effect of the pam\_nologin module

The pam\_nologin.so module provides a shortcut to the system administrator to lockdown the system from any user accounts to login via GUI, Virtual terminal or ssh.

1. Create an empty file, /etc/nologin. Type in:

touch /etc/nologin

1. Switch to one of the virtual terminals, verify that only root can login to the system.

For instance, student cannot login now.

Text

Description automatically generated

As shown at the above, the first login attempt was failed due to a wrong password. The system message is Login incorrect. In the second attempt, the password was correct, but the system shown Authentication failure.

1. Login as root and remove the /etc/nologin file. Now you should be able to login to the system using the student account.
2. Examine the nologin.so configurtion:

Go to /etc/pam.d folder , type:

grep nologin \*

Timeline

Description automatically generated

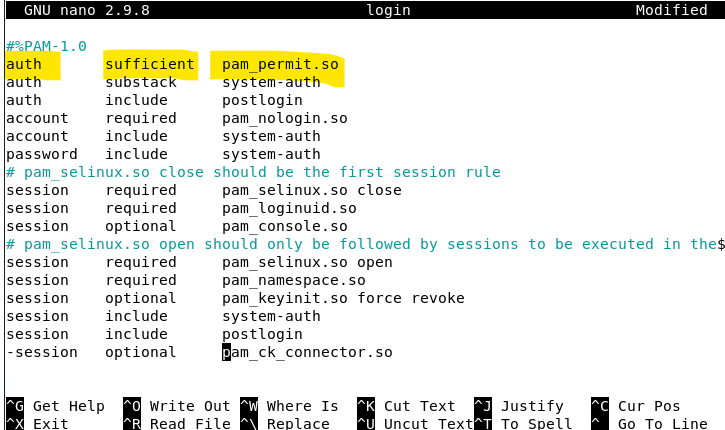
The pam\_nologin.so module is loaded for gdm-password, login, and sshd pam configuration files. Thus all these logins will be suspended when the /etc/nologin file exists.

1. Using pam\_permit to allow access without passwords

On Client:

1. View the man page for pam\_permit to understand what it does.
2. Add the following line to the TOP of /etc/pam.d/login.

auth sufficient pam\_permit.so



1. Switch to a virtual terminal (eg press Ctrl-Alt-F4) and try logging in as any user. Users can login without passwords now.
2. Return to the GUI (press Ctrl-Alt-F2).
3. Verify if you cannot login at the GUI without using the password. You still need to provide the correct password to access to the system.
4. Repeat Step 4 but apply the pam\_permit.so to the file :

/etc/pam.d/gdm-password

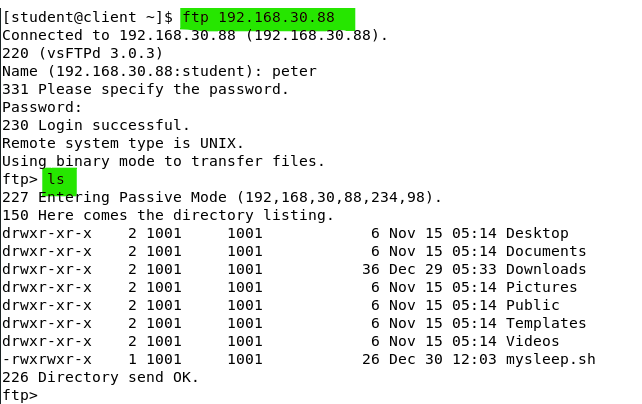
1. Verify if you can login at the GUI without using the password now.
2. After testing, remove the line you just added from both files.
3. Using pam\_listfile to control access to vsftpd service

The vsftpd service is using pam for authentication. We can configure its pam configuration to apply some access control measure for the vsftpd operations.

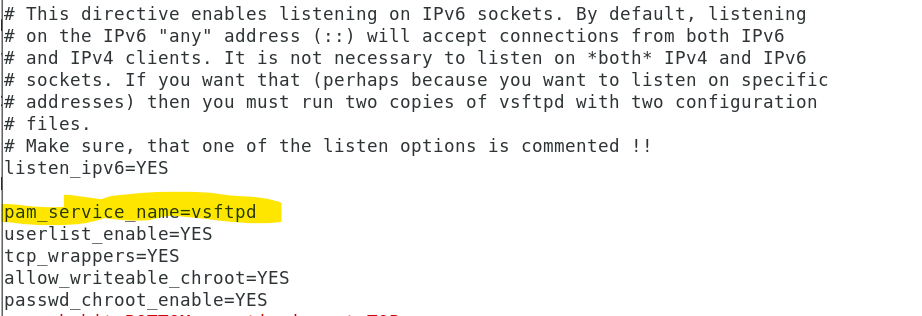
On server:

1. Ensure your vsftpd service is up and running:

It allows local users to access (chroot or not , does not matter) from the client.

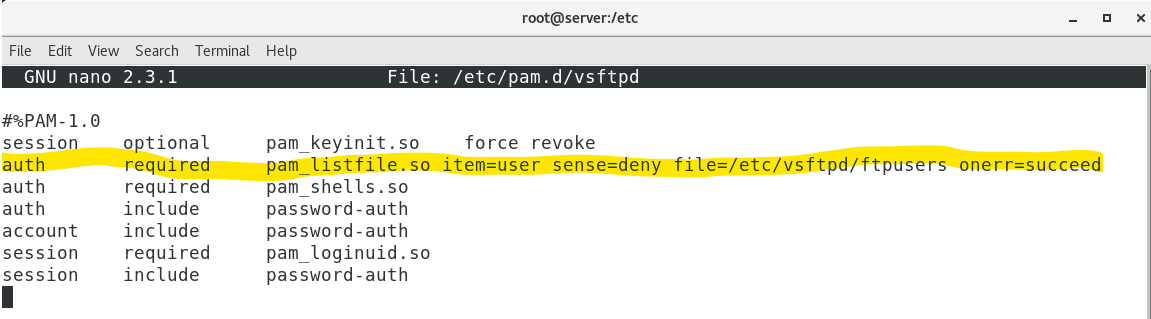


1. View the man page for pam\_listfile to understand how it works.
2. Verify that the /etc/vsftpd/vsftpd.conf has defined the 'pam\_service\_name' setting.

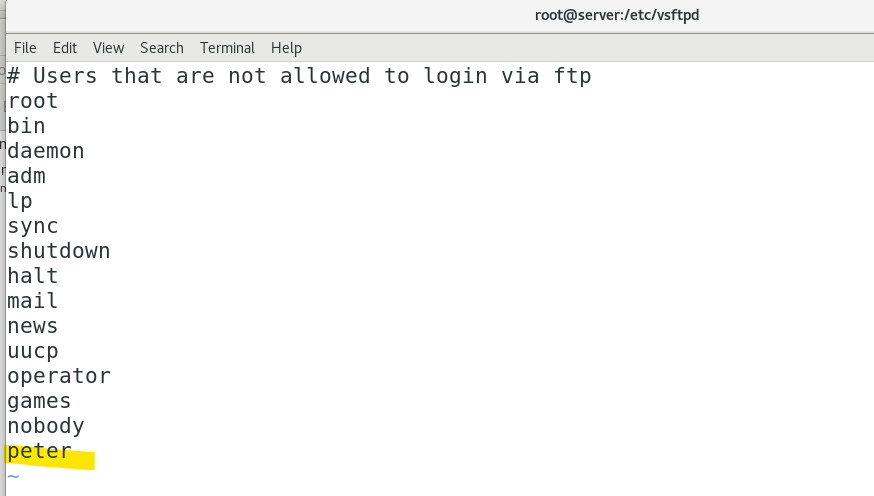


1. View the PAM config file for vsftpd : /etc/pam.d/vsftpd. Note the following line that is already in the file.

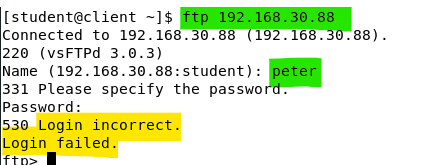
auth required pam\_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers onerr=succeed



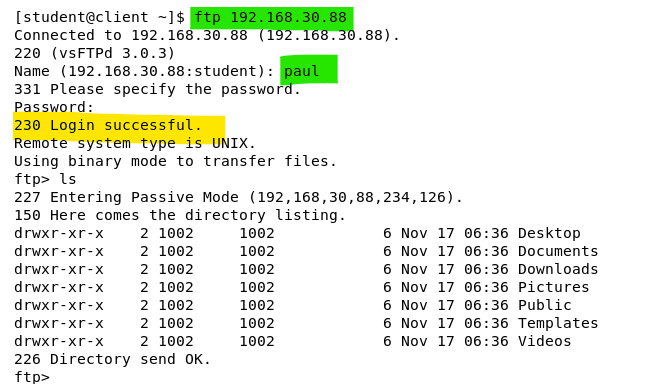
1. Add a user (e.g. peter) to /etc/vsftpd/ftpusers



1. Verify that this user, peter, can no longer ftp to your server.



1. Verify that you can access to the server ftp service using user who is not in the file (e.g. paul).



(Take note that , you may use pam\_listfile to work with any other PAM relevant services !)

1. Using pam\_faillock to track login attempts

This pam\_faillock module may seem to provide similar lockout feature of fail2ban we have learned in last term. However, take note, the pam-faillock is targeting at particular user account while fail2ban is targeting at particular remote client IP address.

On Server:

1. View the man page for pam\_faillock.
2. View the man page for faillock, the utility for reset lockout accounts and viewing the audit log entries related to pam\_faillock.
3. Take note that, pam\_faillock is usually applying two PAM configuration files: system-auth and password-auth. system-auth is usually used by the system login interface process (virtual terminal login, GUI login, sudo … etc), password-auth may be used by other application services (vsftpd, cockpit … ). To ensure the pam\_faillock is applying to all types of logins, we need to add it to both of these two configurations.
4. Review all the PAM configuration files to identify how many services are relating to system-auth and password-auth run the following commands:

grep system-auth /etc/pam.d/\* | awk -F: '{print $1}' | uniq

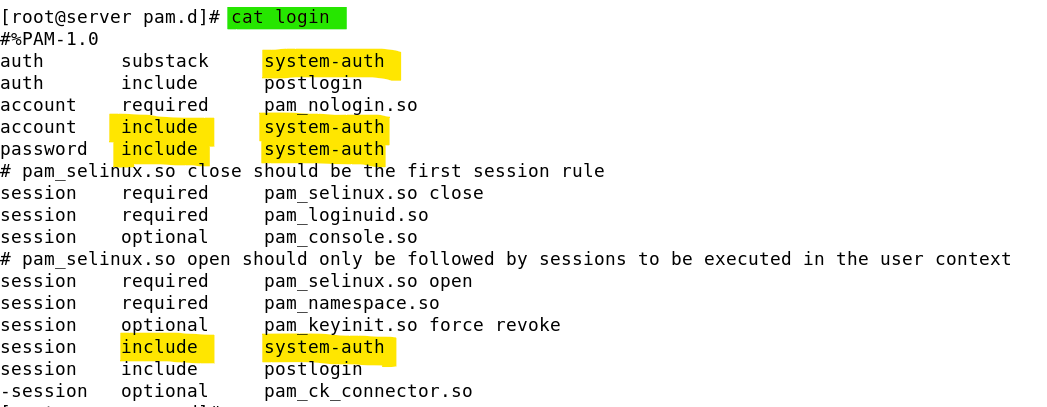
Graphical user interface, text, application

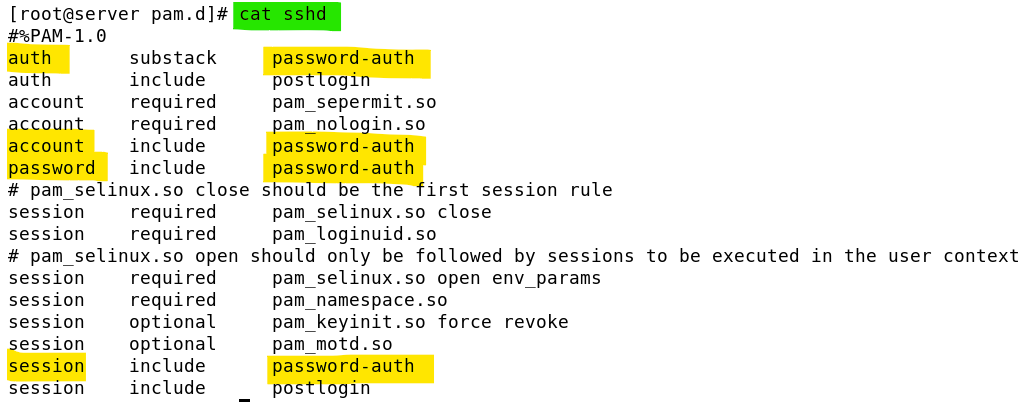
Description automatically generated

grep password-auth /etc/pam.d/\* | awk -F: '{print $1}' | uniq

Graphical user interface, application

Description automatically generated

1. View /etc/pam.d/login and notice that it applies the system-auth files in a few entries, each entry is called at different stages of the login process. (ie. auth, account, and session)
   1. (login links to text based terminal login process)
   2. 
2. View /etc/pam.d/sshd and notice that it applies password-auth file in a few entries though it does not apply any system-auth:



1. Edit /etc/pam.d/system-auth and /etc/pam.d/password-auth. Add the following 3 lines to both configuration files **at the particular order and places** which will activate the pam\_faillock module to all login interface and password authentication related services.

**auth required pam\_faillock.so preauth silent audit deny=3 unlock\_time=never**

**auth [default=die] pam\_faillock.so authfail audit deny=3 unlock\_time=never**

**account required pam\_faillock.so**

A picture containing text

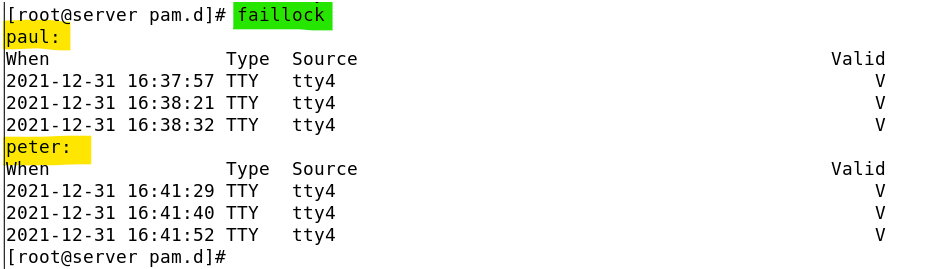
Description automatically generated

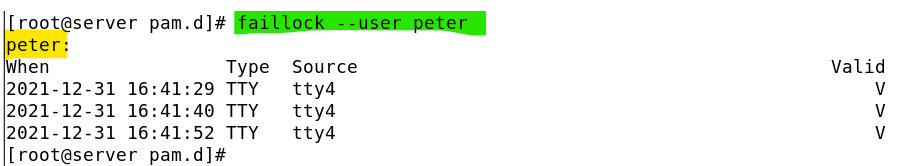
A picture containing graphical user interface

Description automatically generated

Based on the settings shown, when a user has failed to login for more than 3 attempts, the user account will be lockout indefinitely. (as unlock\_time=never is specified, if unlock\_time=600 is used the lockout will be automatically reset after 600 seconds = 10 minutes.).

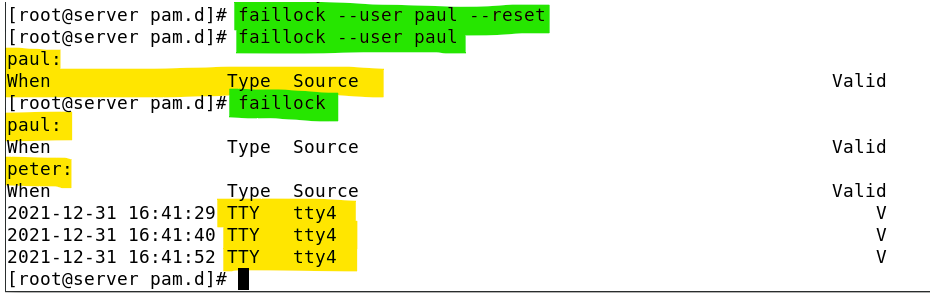
1. To test and verify the settings: switch to a virtual terminal (i.e. press CTRL-ALT-F4).
2. Login with two different user accounts, (e.g. peter and paul), and enter wrong password 3 times for each of them.
3. Try to login with the correct password. The login should still fail.
4. Return to the GUI.
5. Run faillock or faillock --user <username> to view the failed attempts.





1. To reset the user’s account immediately, run the following (replace *username* with your username) :

faillock --user <*username>* --reset

****

Now only peter is remaining locked out but the faillock still returns paul's name. It indicated that paul has been locked out previously.

1. Review the log entries in /var/log/secure:

cat /var/log/secure | grep 'FAILED LOGIN'

A screenshot of a computer

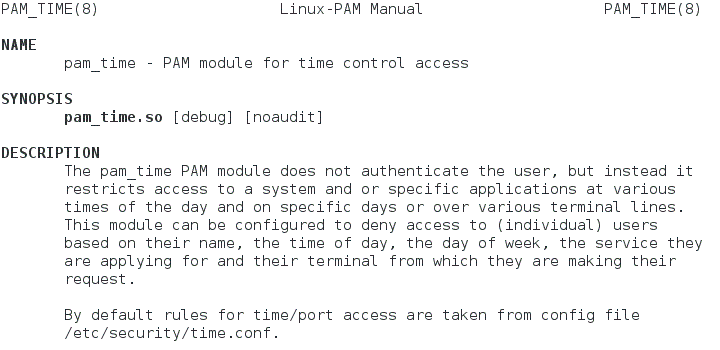
Description automatically generated with medium confidence

1. Using pam\_time to restrict access to services based on time

On server:

1. Read through the man pages for pam\_time. What is the name of the config file for pam\_time?

man pam\_time



Config file for pam\_time

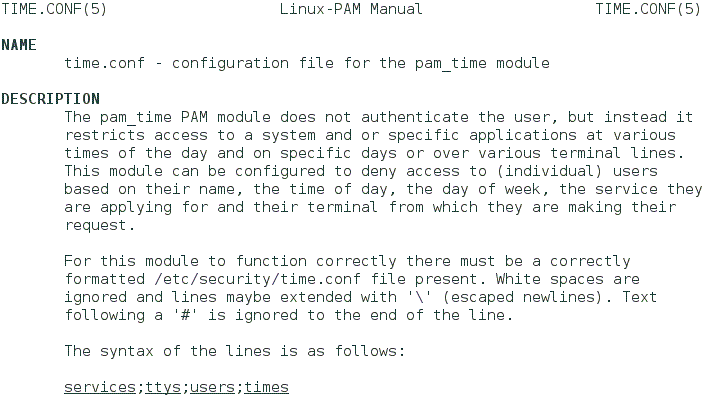
Man page for pam\_time

1. Scroll down to the end of the man page for pam\_time. Under the Section “See Also", it is listed that there is a man page for “time.conf” too.

pam_time2

1. Look at the man page for time.conf and note the syntax of the lines to specify which users are restricted access to various services at different times.

man time.conf



Man page for time.conf

1. Check that the vsftpd service is running.

systemctl status vsftpd

On client:

1. Verify you can ftp to your server as a normal user.

ftp *serverIP*

1. Use one of the user accounts on your server to login (eg "student", "paul", etc).
2. Check that you can login successfully.
3. Type “bye” or “quit” to exit the FTP client.

On server:

1. Configure the vsftpd service to use the pam\_time module to check when users can access the service. Edit the file /etc/pam.d/vsftpd and add the following line (indicated in bold) :

auth required pam\_shells.so

auth include password-auth

Add this line

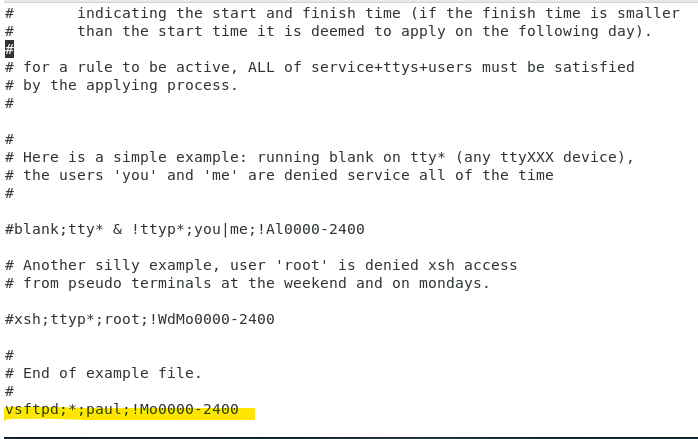
**account required pam\_time.so**

account include password-auth

You will now use the pam\_time module to prevent a user from connecting to the vsftpd service on one particular day of the week.

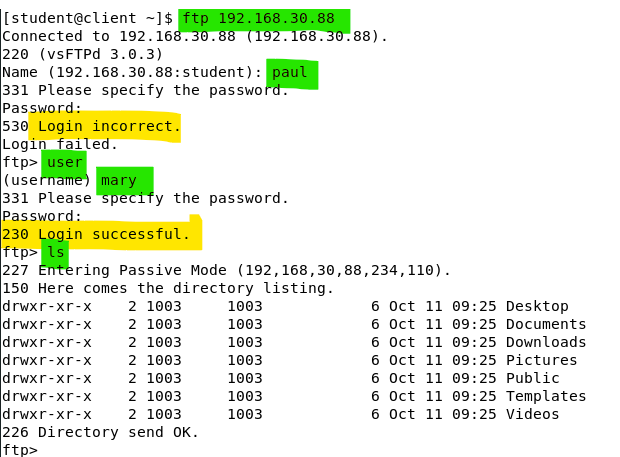
1. Edit the file /etc/security/time.conf and add the following line to the end of the file. This line will prevent user paul (you can use any other username) from connecting to the vsftpd service on Mondays whole day. (If today is Tues, then change “Mo” to “Tu”, if today is Wed, then use “We”, etc)

vsftpd;\*;paul;!Mo0000-2400

****

On client:

1. Check you cannot connect to the vsftpd service on your server as the specified user (e.g. paul) anymore.
2. Check that other users can still connect to the vsftpd service.

****

Peter may not be able to login too, due to the early exercise on pam\_listfile.

On server:

1. Edit /etc/security/time.conf and change the day to another day of the week (eg “Fr”).

On client:

1. Check you can connect to the vsftpd service on your server as the specified user.

On server:

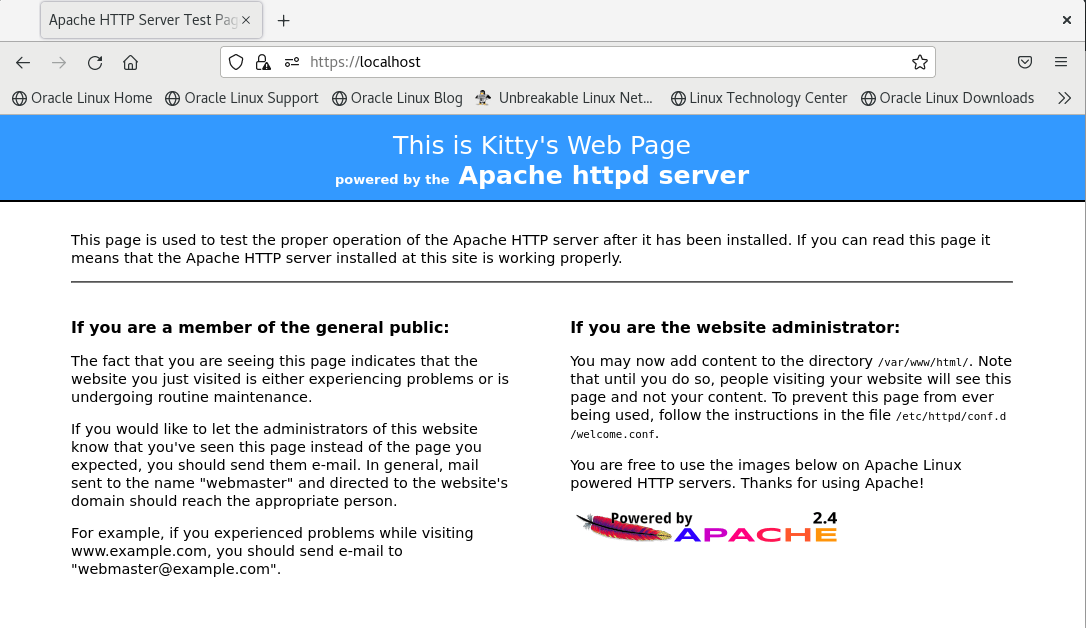
1. Edit /etc/security/time.conf and comment out the line you added so that normal access to the vsftpd service is allowed again.
2. Configure httpd basic authentication via PAM (Optional)

Pam is designed to be a pluggable authentication module to support other system services or applications. Apache web server can also tap on to PAM to use local user account for assess control to its web content.

If your httpd is still running and you have extra time you may on this optional exercise.

On server:

1. Ensure and verify that your httpd service is running fine. You can access to https://localhost page too.



Your default page may not look the same as the above.

1. Installation of the mod\_authnz\_pam package. Type:

dnf -y install mod\_authnz\_pam

1. Edit the file /etc/httpd/conf.modules/55-authnz\_pam.conf. Uncomment the line:

LoadModule authnz\_pam\_module modules/mod\_authnz\_pam.so

Text

Description automatically generated

1. Edit the file /etc/httpd/conf.d/authnz\_pam.conf. Append the following Directory container definition to the end of the file:

# add to the end

<Directory /var/www/html/books>

SSLRequireSSL

AuthType Basic

AuthName "PAM Authentication"

AuthBasicProvider PAM

AuthPAMService httpd-auth

Require valid-user

</Directory>

Text

Description automatically generated with medium confidence

1. Create a new pam configuration file, /etc/pam.d/httpd-auth.

Type in the following lines to the file:

#auth sufficient pam\_permit.so

auth required pam\_listfile.so item=user sense=deny file=/etc/httpd/conf.d/denyusers onerr=succeed

auth include system-auth

account include system-auth

(Take note that the first line is commented out. It is for debugging if needed.)

Text

Description automatically generated

1. Create a new user list file, /etc/httpd/conf.d/denyusers. Write users you'd like to prohibit authentication by the pam\_listfile module:

Text

Description automatically generated

At this sample, we do not allow root to authenticate to access the books directory.

1. Change the group id and the access permissions of the denyusers, to allow httpd service (which is run by the user id apache) to read from the denyusers file.

chgrp apache /etc/httpd/conf.d/denyusers

chmod 640 /etc/httpd/conf.d/denyusers

1. Change the permission to let httpd to read shadow

chgrp apache /etc/shadow

chmod g+r /etc/shadow

1. Enable SELinux boolean: httpd\_mod\_auth\_pam (if the system is selinux enforced)

setsebool -P httpd\_mod\_auth\_pam on

1. Restart httpd

systemctl restart httpd

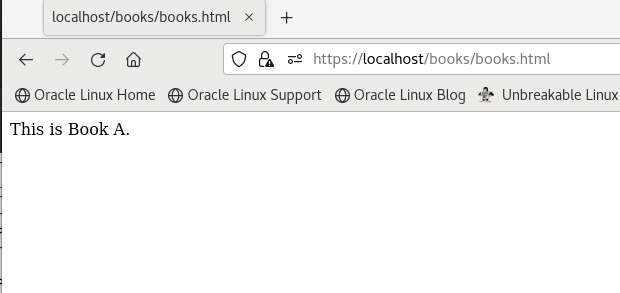
1. Test if the authentication scheme is working for https://localhost/books/books.html

Local user accounts should be able to get through the authentication process.

(Note: you should disable the browser cache of the firefox by referring to this : https://techwiser.com/disable-cache-google-chrome-firefox/. )

Graphical user interface, text, application

Description automatically generated



Yes, mary can authenticate successfully and access to books.html

1. authselect and the nsswitch.conf

authselect is a system command to configure the identity and authentication sources.

In the old days, the sources of the identify and authentication (ie. user accounts, shadow passwords and user grouping formation ) are the local files only (ie. /etc/passwd, /etc/shadow and the /etc/group).

For the modern Linux system, it can tap onto network (Enterprise grade) basis sources. The authselect command allows the administrators to change the choices and it will generate a configuration file, nsswitch.conf to specific the sources of the identity and authentication sources.

On client:

1. As root, run the authselect command to list out all the available choices. Type:

authselect list

Text

Description automatically generated

As shown, there are four profiles (choices).

For our case, we can only use minimal or sssd (Which includes the support of local users scheme.)

1. Run the authselect command to find out the current selected choices. Type:

authselect current

Text

Description automatically generated with medium confidence

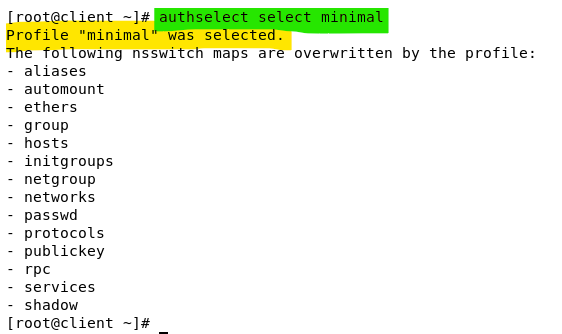
The default choice is sssd.

sssd stands for "System Security Services Daemon". With proper configuration, sssd enables the Linux system to access to a Windows Domain user database to authenticate user access. (ie. the system can let windows domain users to login to the system.)

In our case, we do not configure our system in this ways, so it falls back to support local user only.

1. You may try to select minimum profile. Type:

authselect select minimal



1. Please change it back to sssd profile to avoid any unpredictable issues as the system default has been set to sssd profile at the system installation time. Type:

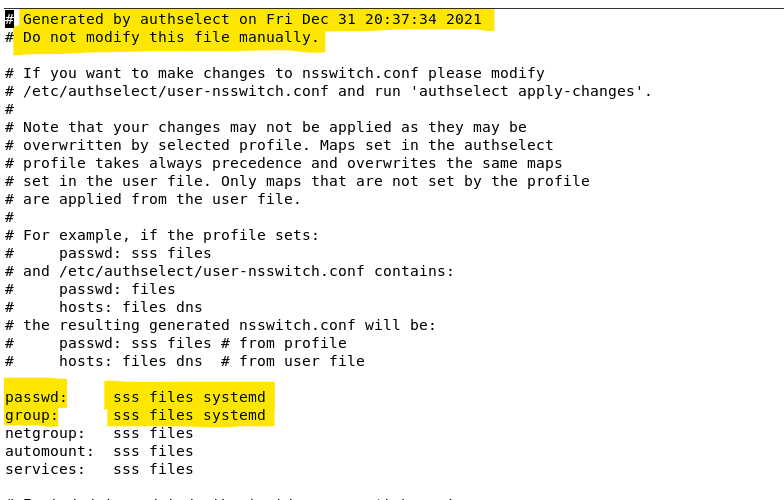
authselect select sssd

A screenshot of a computer

Description automatically generated with medium confidence

Each when we change the authentication sources selection, authselect will re-generate the nsswitch.conf file.

1. View /etc/nsswitch.conf and look at the following highlighted lines .

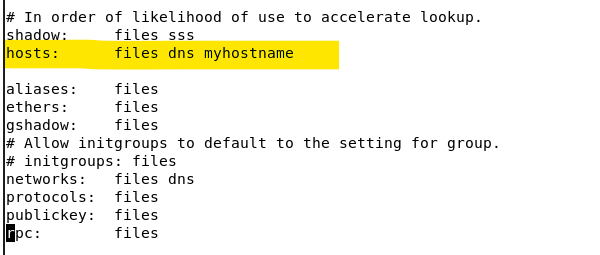


At the top of the file, it states the file generation date/time.

For the passwd, and group configuration, it will try the system security service approach first, then fall back to files (local users) basis. The 'systemd' parameters are appended to ensure the user root and the user nobody are always available even they are missing from any the other sources.

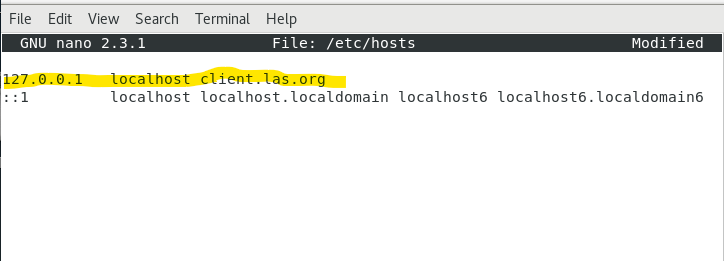
1. Hostnames are resolved by either looking at the local file (/etc/hosts) or by checking the DNS Server. The priorities of these two approach are defined in the nsswitch.conf file.

Observed the highlighted entry found in the nsswitch.conf file:



The default setting for the sssd profile is : /etc/hosts first, dns second, and the last is to check for the myhostname (set by hostnamectl command and stored in /etc/hostname.)

1. Edit the /etc/hosts file to ensure you have set client.las.org as your domain name:



1. Run the ping command to verify the setting is taking effect:

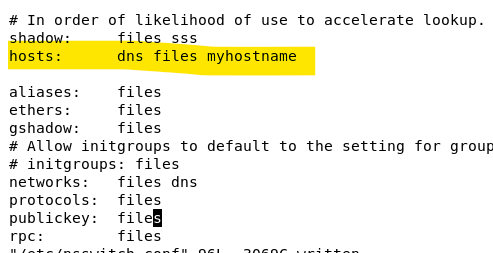
ping -c 3 client.las.org

Text

Description automatically generated

As the /etc/hosts has the highest priority to resolve the hostname, it will return 127.0.0.1 , thus the ping will go to 127.0.0.1

1. Edit the /etc/nsswitch.conf file, to swap the files and dns options for the host entry:



1. Repeat the ping command in step 4 to see the difference.

ping -c 3 client.las.org

Text

Description automatically generated with low confidence

The above shown that, the las.org domain should be existing, and the DNS has reported that there is no host with the name client.las.org found in the las.org domain.

1. Allocating admin tasks to normal users with sudo

We have learned the basic configuration of sudo at the beginning of this module.

Let's revisit the sudo configuration to grant specific user some but not all the administrative right.

In this exercise, you are assigning the user mary to be the administrator for the Apache Server. However, you do not want to give him the root password nor other root access right.

On server:

1. As user root, type “visudo”. (or you can use nano to edit the file /etc/sudoers)

Add the following line to the end of the file. You are allowing simon (or use any other user you like) to run the /usr/bin/systemctl script to start and stop the httpd service.

mary ALL=/usr/bin/systemctl \* httpd

1. Save the file and quit the editor.
2. As user mary, type “systemctl status httpd” to view the status of the Apache Server first.
3. As user mary, type “sudo systemctl stop httpd” or “sudo systemctl start httpd” to stop or start the Apache Server.

As the Apache Server administrator, mary would also need to be able to edit the Apache configuration files.

1. As user root, type “visudo”. (or you can use nano to edit the file /etc/sudoers)
2. Edit the following line to allow mary to use nano editor to modify any of the httpd config files.

mary ALL=/usr/bin/systemctl \* httpd, /usr/bin/nano /etc/httpd/\*

1. Save the file and quit the editor.
2. As user mary, check that you are able to use sudo to modify the files in /etc/httpd/ by tying the following command : “sudo nano /etc/httpd/conf/httpd.conf”. Make a change (eg add a blank line) and test that mary can save the modified file.
3. As user root, check the file /var/log/secure to see that mary’s sudo tasks have been logged down.

Text

Description automatically generated

With the /var/log/secure, we can always keep tracking what tasks the specific users have been doing use her sudo sessions.

*End of Practical*