Using PAM to control users login

Configuring the pam_access.so module
PAM - Pluggable Authentication Modules for Linux is a system of libraries that handle the authentication tasks of applications (services) on the system. We can use the **pam_access.so**, a PAM module manages access to restrict access to any given user dynamically. Its default configuration file is the **/etc/security/access.conf**. Each line of the **access.conf** has three fields separated by a ":" character (colon):

permission:users/groups:origins

- The first field, the permission field, can be either a "+" character (plus) for access granted or a "-" character (minus) for access denied.
- The second field, the users/group field, should be a list of one or more login names, group names, or ALL (which always matches). To differentiate user entries from group entries, group entries should be written with brackets, e.g. (group).
- The third field, the origins field, should be a list of one or more tty names (for non-networked logins), host names, domain names (begin with "."), host addresses, internet network numbers (end with "."), internet network addresses with network mask (where network mask can be a decimal number or an internet address also), ALL (which always matches) or LOCAL. For more details, please read man page: ACCESS.CONF(5).
- The EXCEPT operator makes it possible to write very compact rules.

When someone logs in, the file **access.conf** is scanned for the first entry that matches the (user/group, host) or (user/group, network/netmask) combination, or, in case of non-networked logins, the first entry that matches the (user/group, tty) combination. The permissions field of that table entry determines whether the login will be accepted or refused.

EXAMPLES
These are some example lines which might be specified in the **/etc/security/access.conf**.

1. User root should be able to get access from tty1 and tty2.
   +: root : tty1 tty2
2. Disallow console logins to all but the shutdown, sync and all other accounts, which are a member of the wheel group.
   - :ALL EXCEPT (wheel) shutdown sync:LOCAL
3. Disable root login from all sources except tty4
   - : root : ALL EXCEPT tty4

Enforcing the pam_access.so module
Directives in the **/etc/security/access.conf** don't take effect unless they are implemented in particular applications (services). Each PAM-aware application or service has a configuration file in the **/etc/pam.d/** directory. For example, when accessing a system via ssh through sshd, the **/etc.pam.d/sshd** is consulted. Each configuration file contains multiple lines of directives formatted as follows:

<type> <control flag> <module> [<module arguments>]

Generally, when an application needs to authenticate a user, these directives in the list run from top to bottom. In each line, PAM calls a **module**. The **module** returns a success or failure result and the **control flag** tells PAM what to do with the result. The **type** represents a stage of the authentication process. In each stage(**type**), PAM may call multiple modules in different lines. These results stack and generate an overall pass or fail result for each **type**. If any one **type** fails, the application fails at that stage. For more details, please check Introduction to PAM.

We have learned how to grant or deny users permission using **access.conf**. Assuming we want to deny certain users login via ssh, we need to make sure that PAM calls the **pam_access.so** in the **/etc.pam.d/sshd** when the sshd service runs. And if the **pam_access.so** returns a fail result, PAM should treat it as a fatal failure of the authentication process. Therefore, we insert the following line at the beginning of **/etc.pam.d/sshd**:

account required pam_access.so

If you have read the Introduction to PAM, you know why the order of directives in **/etc.pam.d/sshd** matters.

Putting our line on top of others, so it won't be skipped. And the **required** flag means the result of the **pam_access.so** must be successful for the authentication to continue. Once the **pam_access.so** returns a failure, the overall result for the **account** type(stage) will be fail and the sshd service will fail in the stage of verifying a user account.

To fully control users login, these two files should also be configured to honor the pam_access.so module :
The **/etc/pam.d/login** configuration file controls console login.
The **/etc/pam.d/ gdm-password** configuration file controls GUI login.

Caution: some PAM-aware applications (services) may be set to honor the **pam_access.so** module as default. Run the following cmd to find them:

grep -l pam_access.so /etc/pam.d/*

Any change in the **/etc/security/access.conf** may affect users when they try to run these applications (services). Make some exceptions in the **access.conf** or comment out those lines in files under **/etc.pam.d/**. So you don't accidentally block users from running PAM-aware applications (services).

For more details, please check man page: PAM.CONF(5), PAM(8), ACCESS.CONF(5).

Exercise - Controlling root logins
This exercise requires access to the GUI console and it should be done on your virtual machine. Make sure you can login as root and student directly via ssh, virtual consoles and GUI console before you start. Remember to backup your files before you modify them.

1. From a terminal window on the host machine, ssh into your virtual machine's root account and connect.
2. Configure the **/etc/security/access.conf** only to deny user **student** to login from all sources.
3. Open another terminal window on the host machine, try if you can ssh to your VM as **student**. How about GUI console and virtual consoles.
4. From the root terminal window, Configure the **/etc.pam.d/sshd** to honor the pam_access.so module. Repeat step 3, see if there are any changes.
5. From the root terminal window, Configure the **/etc.pam.d/gdm-password** to honor the pam_access.so module. Repeat step 3, see if there are any changes.
6. From the root terminal window, Configure the **/etc.pam.d/login** to honor the pam_access.so module. Repeat step 3, all your attempts should be denied.
7. From the root terminal window, grep "pam_access" in the **/var/log/secure** to see what happened.
8. Comment out the line you added to the **/etc/security/access.conf**. Now you should able to login as **student** as before.
9. Configure the **/etc/security/access.conf** so **root** cannot login from all source expect tty4.
10. find out what other PAM-aware applications (services) honor the **pam_access.so** module. Fix them so that **root** can run those applications (services).

Answers and Hints

2. Add the following line to access.conf:

-:student:ALL

3. You should able to login your VM without any problem, since the pam_access.so module isn't active.
4. Add the following line at the beginning of the **/etc.pam.d/sshd**:

account required pam_access.so

Your existing terminal should work fine, but you won't be able to make any new ssh connection to your VM as **student**.

5. Add the following line at the beginning of the **/etc.pam.d/gdm-password**:

account required pam_access.so

You should not be able to login to GUI as student.

6. Add the following line at the beginning of the **/etc.pam.d/login**:

account required pam_access.so

You should not be able to login to any virtual console as student.

7. The **/var/log/secure** should show records that access denied for **student** from different sources.
9. Add the following line to access.conf:

- : root : ALL EXCEPT tty4

10. Both **/etc/pam.d/atd** and **/etc/pam.d/crond** call the **pam_access.so** module to verify user accounts. Run this cmd:

crontab -l

You should see a warning like this:

You (root) are not allowed to access to (crontab) because of pam configuration.

Comment out those lines or add **atd** and **cron** to the exception sources list in the **/etc/security/access.conf** like this:

- : root : ALL EXCEPT tty4 atd cron

Now run **crontab -l** again, crontab should run without any problem. And if you never create any cron job, it will show you **no crontab for root**.