1 How to configure User restriction with (PAM) (a kind of parental control)

kernel pam

I would like to use kernel module Pluggable Authentication Module (PAM) to restrict the access or deny the computer to some user in a specific set of hours. it also can be user limit access for children (a kind of parental control).

asked **Nov 17 '11**
hhlp
4066 ●26 ●46 ●79

specialy using the command-line.....

add a comment

1 Answer

Sort by » oldest newest most voted

2 When we talk about forcing a user to log off, what we're really talking about is implementing time restrictions on the account for system access or services. The easiest way I've found to implement time restrictions is using a plug-in module called **Linux-PAM**.

answered **Nov 17 '11**
hhlp
4066 ●26 ●46 ●79

updated **Nov 17 '11**

Pluggable Authentication Module (PAM) is a mechanism for authenticating users. Specifically, we're going to use the `pam_time` module to control timed access for users to services.

Using the `pam_time` module, we can set access restrictions to a system and/or specific applications at various times of the day as well as on specific days or over various terminal lines. Depending on the configuration, you can use this module to deny access to individual users based on their name, the time of day, the day of week, the service they're applying for, and their terminal from which they're making the request.

When using `pam_time`, you must terminate the syntax of each line (or rule) in the `/etc/security/time.conf` file with a newline. You can comment each line with the pound sign [#], and the system will ignore that text until the newline.

Here's the syntax for a rule:

> services;ttys;users;times

```
The first field – services – is a logic list of PAM service names.
The second field – tty – is a logic list of terminal names.
The third field – users – is a logic list of users or a netgroup of users.
The fourth field – times – indicates the applicable times.
```

Here's an example of a typical set of rules:

```
login ; * ; !user ; MoTuWeThFr0800-2000
login ; * ; !user ; !Al0000-2400
http ; * ; !user ; MoTuWeThFr0800-2000
http ; * ; !user; !Al0000-2400
```

These rules restrict user 'user' from logging on between the hours of 0800 and 2000, and they also restrict Internet access during these hours. Root would be able to logon at any time and browse the Internet during all times as well.

Note: The system logs errors with these rules as syslog(3).

Use your votes!

- Use the **30** daily voting points that you get!
- **Up-vote well framed questions** that provide enough information to enable people provide answers.
- **Thank your helpers** by up-voting their comments and answers to your questions.
- **Down-voting might cost you karma**, but you should consider doing so for incorrect or clearly detrimental questions and answers.

Question tools

Follow

2 followers

subscribe to rss feed

Stats

Asked:	**Nov 17 '11**
Seen:	**4,105 times**
Last updated:	**Nov 17 '11**

Related questions

- GRUB2 Colemak Keymap Default
- Why do I not have sound in my laptop speakers with the new kernel?
- Fedora 20 Install failing - no mirrors - kernel-PAE-devel-3.11.10-301
- Black screen after upgrade to kernel 3.10?
- F20: Unable to make prepare on kernel source
- Unable to load TPM kernel module: "required key not available"
- how to uninstall kernel in fedora17
- Intel HD 3000 Fedora 18: Cursor Issue

AES variant used by kernel crypto

kernel panic fedora 20 (64) vmware workstation 7.1

With Fedora, it is possible to assign to your computer time restrictions, to prevent the connection of one or more users to your system. With the time restrictions, you can, for example, limit access to the computer for your children *(a kind of parental control, in short)*, or even protect the connection to your server during certain hours.

Manual Configuration

Understand what you will do

Throughout this tutorial, we will use PAM (Pluggable Authentication Modules. It allows you to control user authentication when they connect. Then, we will use the security configuration files to define logon hours allowed. These manipulations can be performed on any version of Fedora, and require only a simple text editor (vim, emacs, nano, gedit, kate).

Enable Restrictions hours via the PAM Module

If we want to block the connection to the computer, we will have to change the gdm service. Edit the file so gdm and add this line of code (at the end of file):

```
account required pam_time.so
```

GDM is the login screen distributions for Fedora Gnome. For Fedora KDE spin, which uses KDE, kdm service is called, it will be the file it will open And you're done for configuring the PAM! This will enable the control of hours on this service.

Configure Access Hours

Now that the PAM service has been activated, we only have to configure access times. Open the `/etc/security`. Several configuration files are available:

```
ls /etc/security/

access.conf       console.perms    limits.d        opasswd
chroot.conf       console.perms.d  namespace.conf  pam_env.conf
console.apps      group.conf       namespace.d     sepermit.conf
console.handlers  limits.conf      namespace.init  time.conf
```

Edit the file `time.conf`. Some explanations and examples introducing the. To set access schedules, copy and paste the following line of code (at the end of the file, as always):

```
* ; * ; user ; scheduler
```

Instead of the user field, enter the login account you want to block.

If you want to block multiple users, enter their login in a row, separated by the '|' operator. For example, if I want to freeze the accounts of user1, user2 and user3:

```
* ; * ; user1|user2|user3 ; scheduler
```

By cons, if you want to block access to the system for all users but one in particular, use the '!' before the person concerned.

For example, if I want access to the computer is denied to all users, except user4 and user5:

```
 * ; * ; !user4|user5 ; scheduler
```

Turning now to the field zones. In this field that the selection of days and hours will be allowed connection possible. You must first specify the day of the week, using the following abbreviations:

```
Mo : Monday     Fr : Friday     Wd : Sa/Su
Tu : Tuesday    Sa : Saturday   wk : Mo/Tu/We/Th/Fr
We : Wenesday   Su : Sunday
Th : Thursday   Al : All Days
```

Be careful not to confuse the abbreviations Wk and Wd are misleading! particularly poorly identified on the Internet: you can easily find conflicting information!

Then, we specify the deadlines. These should be formatted 24H, consisting of 4 digits. For example, to restrict 3:17 p.m. to 6:34 p.m., we write: 1517-1834. To allow user6 to connect only on Tuesday, from 3:17 p.m. to 6:34 p.m., we obtain the result:

```
* ; * ; user6 ; Tu1517-1834
```

Connections outside of these hours will be banned. As for users, it is possible to use the operators '|' and '!' to indicate several times (the '!' then indicate that all logon hours are allowed, except those to be shown).

The two stars (wildcards) at the beginning of the line of code are, respectively, and tty services fields. Since you want to block all access to the system, it is unnecessary to specify what service or what tty you want to block. However, if you want to prevent the use of a particular service, simply specify it as the following example:

```
login ; tty1|tty4|tty5 ; user6 ; !Wd0000-2400
```

Thus, the user 'user6' can not connect to a TTY, 4 and 5 during the weekend.

Some Examples of Restrictions Schedule

user7 is allowed to connect every day from 1:20 p.m. to 3:20 p.m. and from 4:00 p.m. to 8:30 p.m.:

```
* ; * ; user7 ; Al1320-1520|Al1600-2030
```

user8, user9 and user10 are allowed to connect to 2:00 p.m. to 6:45 p.m. during the weekdays, and 2:00 p.m. to 10:15 p.m. for the weekend:

```
* ; * ; user8|user9|user10 ; Wk1400-1845|Wd1400-2215
```

user11 is never allowed to connect. user12 can log on Wednesday from 1:00 p.m. to 4:00 p.m.:

```
* ; * ; user11 ; !Al0000-2400
* ; * ; user12 ; We1300-1600
```

2 different lines, for two different time for each user Expiration of a Session

When a session expires (it exceeds the time while the user is already connected), the PAM can reach the user. While user7 connects during the hours of time allowed, it is perfectly free to exceed these hours! For this, we will use a new program: ´cron´. This application executes commands at intervals of time. In our case, we will make use of command ´skill -KILL -u´ to disconnect the user when the session expires. Handling is very simple. Simply edit the file ´/etc/crontab´. Then add the following line of code:

```
Minute Hour Day * * (s) root skill -KILL -u User
```

As before, replacing the Minute field schedules and time desired. Then fill in the day (s) by (s) day (s) banned (s), or simply type an asterisk (*) to indicate all days of the week. Finally, change the field used by the login account to be blocked, and voila!

Days do not notice the same way with the `cron` jobs! Here is the list of abbreviations to be used with this program:

```
mon : monday    fri : friday
tue : tuesday   sat : saturday
wed : wednesady sun : sunday
thu : thursday   *  : all hours
```

Some Examples of `cron` jobs (with examples of times in the previous section)

user12 can log on Wednesday from 1:00 p.m. to 4:00 p.m.

> -> Disconnect: Tuesday at 4:00 p.m..

```
00 16 * root * wed skill -KILL -u user12
```

user7 is allowed to connect every day from 1:20 p.m. to 3:20 p.m. and from 4:00 p.m. to 8:30 p.m..

> -> Disconnecting: Daily, 8:30 p.m. to 3:20 p.m. ET.

```
20 15 * * * root skill -KILL -u user7
30 20 * * * root skill -KILL -u user7
```

user8, user9 and user10 are allowed to connect to 2:00 p.m. to 6:45 p.m. during the weekdays, and 2:00 p.m. to 10:15 p.m. for the weekend

> -> Disconnect (1): Monday, Tuesday, Wednesday, Thursday and Friday, at 18:45. -> Disconnect (2): Saturday and Sunday at 10:15 p.m..

```
45 18    * * mon,tue,wed,thu,fri    root    skill -KILL -u user8 && skill -KILL -u user9 && skil
l -KILL -u user10
15 22    * * sat,sun                root    skill -KILL -u user8 && skill -KILL -u user9 && skil
l -KILL -u user10
```

The command skill -KILL -u disconnects the user from the GUI, as well as TTY. It is perfectly usable for server administrators. However, this command is immediate and the disconnection will be made without notice. It would therefore be preferable to prevent the installation of this device users of the computer or network in question!

It is possible to prevent users with a `wall` command launched by `cron` few minutes before the end of the *timeframe*, that will be displayed in the terminals of all users.

```
40 18 * * Mon,Tue,wed,thu,fri root echo "end of session in 5 minutes" | wall
```

To prevent users from GUI can be used in place of the wall command `notify-send`.

```
40 18 * * Mon,Tue,wed,thu,fri user8 DISPLAY=:0 notify-send "end of session in 5 minutes"
```

Crontab :

Command prompt:

```
crontab -l lists out the cron jobs.
crontab -e brings up the jobs in vi so you can change them.
```

image description

or if you want a graphics interface for CRON use gnome-schedule :

```
yum install gnome-schedule
```

add a comment link

Your answer

Please start posting your answer anonymously - your answer will be saved within the current session and published after you log in or create a new account. Please try to give a **substantial answer**, for discussions, **please use comments** and **please do remember to vote** (after you log in)!

Add answer