[PAM](http://shabbathster.blogspot.com/2013/09/pam.html)

                           PAM  
                     --------------------  
  
  
       PLUGGABLE AUTHENTICATION MODULES  LDUP : 18.Dec.2k3 Thu 0100  
                   [Adv SysAdmin - System Security]  
                               by Sun                                                                       
  
[See also checkList at EOF]  
  
[Always keep an eye on the Linux system log file : /var/log/messages]  
  
- What is PAM ?  
  
  The glue between authentication methods [eg one-time pwds, kerberos,  
  smart cards] and applications requiring authentication services  
  [eg ftpd, sshd, imapd] etc  
  
  It is another layer on 1st-Level defence in Linux  
  
- The 4 directories we are concerned with :  
  
    /lib/libpam\*      [3]     PAM System Libs  
    /lib/security/.so [40]    PAM Loadable Modules [PLM]  
     
    /etc/pam.d/       [94]  
    /etc/security/    [ 6 .conf files and some other files/dirs]  
         
           =======================================================  
           1. PAM System Libs : /lib/  or "PAM System Libraries"  
                        or libpams  
           =======================================================  
  
# cd /lib  
# ll|grep pam|grep -v ^l  
   or  
# ll|grep pam|grep ^-  
  
The Actual PAM System Libs :  
----------------------------  
  
-rwxr-xr-x    1 root     root        10064 Feb 11  2003 libpamc.so.0.75  
-rwxr-xr-x    1 root     root         8548 Feb 11  2003 libpam\_misc.so.0.75  
-rwxr-xr-x    1 root     root        30448 Feb 11  2003 libpam.so.0.75  
  
SymLinks to the above System Libs :  
-----------------------------------  
  
lrwxrwxrwx 1 root root 15 Apr 13  2003 libpamc.so.0 -> libpamc.so.0.75  
lrwxrwxrwx 1 root root 19 Apr 13  2003 libpam\_misc.so.0 -> libpam\_misc.so.0.75  
lrwxrwxrwx 1 root root 14 Apr 13  2003 libpam.so.0 -> libpam.so.0.75  
  
  
You will see 6 files :  
  
                           libpamc.so.0 -> libpamc.so.0.75  
      libpamc.so.0.75  
                           libpam\_misc.so.0 -> libpam\_misc.so.0.75  
      libpam\_misc.so.0.75  
                           libpam.so.0 -> libpam.so.0.75  
      libpam.so.0.75  
       
      ===================================================================  
      2. /lib/security/ -   "PAM Loadable Modules" - PLMs [40]  
                             All are shared objects [.so]  
      ===================================================================  
     1 pam\_access.so  
     2 pam\_chroot.so  
     3 pam\_console.so  
     4 pam\_cracklib.so  
     5 pam\_deny.so  
     6 pam\_env.so  
     7 pam\_filter.so  
     8 pam\_ftp.so  
     9 pam\_group.so  
    10 pam\_issue.so  
    11 pam\_krb5afs.so  
    12 pam\_krb5.so  
    13 pam\_lastlog.so  
    14 pam\_ldap.so  
    15 pam\_limits.so  
    16 pam\_listfile.so  
    17 pam\_localuser.so  
    18 pam\_mail.so  
    19 pam\_mkhomedir.so  
    20 pam\_motd.so  
    21 pam\_nologin.so  
    22 pam\_permit.so  
    23 pam\_pwdb.so  
    24 pam\_rhosts\_auth.so  
    25 pam\_rootok.so  
    26 pam\_securetty.so  
    27 pam\_shells.so  
    28 pam\_smb\_auth.so  
    29 pam\_smbpass.so  
    30 pam\_stack.so  
    31 pam\_stress.so  
    32 pam\_tally.so  
    33 pam\_time.so  
    34 pam\_timestamp.so  
    35 pam\_unix.so  
    36 pam\_userdb.so  
    37 pam\_warn.so  
    38 pam\_wheel.so  
    39 pam\_winbind.so  
    40 pam\_xauth.so  
  
                        ====================  
                        3. /etc/security/ -  
                        ====================  
  
limits.conf  
time.conf  
access.conf  
pam\_env.conf  
console.perms  
console.apps/................  
  
                =======================================  
                4.  /etc/pam.d/  - The PAM Config Files  
                =======================================  
  
Mgmt Group aka  
stacked modules  
  
module-type  control-flag   module-path                       [args]  
===========  ============   ===========                       ======           
auth         required      defaults to /lib/security/\*.so     debug  
account      requisite                                        no\_warn  
password     sufficient                                       use\_first\_pass  
session      optional                                         try\_first\_pass  
                                                              expose\_account  
                      Module Types  
       ============  
  
  1. auth :    Serves 2 fns - first authenticating users to be who they claim  
               to be, and second allowing other privileges to be granted to the  
        users  
  
  2. account : Provides account-mgmt options not related to authentication.  
               Typically used to restrict based on factors such as origin  
        [eg only non-root users allow remote login, or time of day]  
  
  3. password: Called only when updating the auth token associated with the  
               user  
  
  4. session : Tasks to be performed before/after the user is allowed access.  
  
                       Consequences of control flags:  
                       =============================  
  
control     
flag       What ?  
=======    =========================================================  
  
required   Compulsory to succeed. If it does not, executing of other  
           modules of the same module-type still continues  
  
requisite  Similar to required. Compulsory to succeed. However, in the case  
           of failure, control is passed straight back to the app, rather  
           than other modules being executed  
  
sufficient If a module with the control flag sufficient succeeds, no further  
           modules of the same module-type are called and the entire  
    module-type succeeds  
                     PROVIDED  
    all other modules before it have succeeded - in the same module  
    type, of course  
  
optional   Success of this module is optional, success or failure is  
           irrelevant  
  
                               Args  
        ====  
   debug   : Log debug info using syslog         
  
   no\_warn : Suppress warning msgs  
  
   use\_first\_pass : If a pwd has been previously entered, use it and do not  
                    prompt user another time  
  
   try\_first\_pass : Auth should be tried with the previously entered pwd.  
                    If a pwd was not entered or is invalid, the user is  
     prompted for a password  
  
   expose\_account : By default, PAM attempt to hide a/c info like user's full  
                    name or default shell [a cracker could use this]  
     This arg allows such info to be displayed, making for a  
     more friendly login experience [eg "Pl enter your password,  
     Mr Vada Pavji), but should be used in a secure env  
  
   nullok           If you remove the 'x' in /etc/passwd, a user can login  
                    w/o a passwd. That is bcos of this.  
                    Remove nullok and it will not allow it  
      
   likeauth  
    
   retry=N          The default number of times this module will request a new  
                    password (for strength-checking) from the user is 1.  
     Using this argument this can be increased to N  
  
     This happens when you are changing the passwd from the CLI  
     and not when you are logging in, obviously!  
    
   type=            The default action is for the module to use the following  
                    prompts when requesting passwords: ``New UNIX password: ''  
     and ``Retype UNIX password: ''.  
     Using this option you can replace the word UNIX with XXX  
      
                    See Exercise IV  
    
   use\_authok  
   md5  
   shadow  
  
    
Q: When you log in, what does login\* binary do  ?  
   
   1. Accepts your un/pwd  
   2. Loads the 3 PAM system software libs into the RAM  
   3. Hands over this un/pwd to these libs  
   4. Then instructs the libs to loads login's PAM config file -  
      /etc/pam.d/login  
      [The PAM config file of the login\* service which is by the  
       same name i.e. /etc/pam.d/login]   
   5. Hands over control to the PAM libs  
   6. PAM System Libs start processing the file /etc/pam.d/login  
   7. libpam loads and unloads the PLMs as given in the config file  
  
                             ==========  
                             Exercise I  
                             ==========  
  
     Add the following lines to the existing /etc/pam.d/login file :  
  
     [For explanation of what adding these lines does, see Exercise II]  
  
#%PAM-1.0  
auth       required pam\_securetty.so  
auth       required pam\_stack.so service=system-auth  
auth       required pam\_nologin.so                      
auth       required pam\_pwdb.so                       <================  
account    required pam\_stack.so service=system-auth  
account    required pam\_time.so                       <================  
account    required pam\_access.so                     <================  
password   required pam\_stack.so service=system-auth  
session    required pam\_stack.so service=system-auth  
session    required pam\_limits.so                     <================  
session    optional pam\_console.so  
  
      ===========  
                             Exercise II  
                             ===========  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
                     auth       required pam\_pwdb.so    
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
\*   Will prompt you twice [once more] for your passwd on login  
  
\*   Repeat the line if you wish to be prompted for the pwd again...etc etc  
  
\*   Now put the arg use\_first\_pass with this, viz :  
  
            auth       required pam\_pwdb.so   use\_first\_pass  
  
    Now will behave just like before, i.e. will ask for the pwd only once  
    and accept the pwd that has been previously entered  
  
\*   Now put the arg nodelay with this :  
  
            auth       required pam\_pwdb.so   use\_first\_pass nodelay  
  
     If you change your passwd, and give a wrong one, it will wait for one  
     whole irritating second, by default.  
  
     This arg - nodelay - will disable this one sec delay  
  
  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
                      /etc/security/time.conf  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
  
These are all ALLOW rules which when FALSE become DENY ones !  
-----------------------------------------------------------------------------  
         login;tty5;foo;Al1205-1206  
  
  Allow foo to login only on any day between 1205 to 1206 but only on tty5  
  Outside the time range, foo will not be allowed to login on tty5  
  
  foo can however login anytime on any other terminal since those are not  
  mentioned in the above rule, and hence, do not obviously get affected.  
-----------------------------------------------------------------------------  
         login;\*;foo;Al1205-1206  
  
  foo can login from any tty but only within the time range.  
  Outside the time range, she is effectively logged out of the system.  
-----------------------------------------------------------------------------  
         login;\*;foo;Al1205-1106  
  
  Similar to above - since the outer range is smaller than the former,  
  the outer time range is considered to be for the next day  
-----------------------------------------------------------------------------  
         login;\*;foo;AlMo1205-1106  
  
 Similar to above - All days except Mondays i.e. the other 6 days of the week  
-----------------------------------------------------------------------------  
         login;\*;foo;WkMo1205-1106  
  
  Similar to above - All weekdays except Mondays i.e. Tue-Fri  
-----------------------------------------------------------------------------  
         blank;tty\* & !ttyp\*;foo|bar;!Al0000-2400  
  
   Running blank on tty\* (any ttyXXX device), the users 'you' and 'me' are  
   denied service all of the time  
-----------------------------------------------------------------------------  
         login;\*;root;MoWd0000-2400  
  
  'root' is denied bash access  from pseudo terminals at the weekend and on  
   mondays.  
-----------------------------------------------------------------------------  
  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
                        /etc/security/limits.conf  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
-----------------------------------------------------------------------------  
             foo               hard    nproc          5  
  
Login as 'foo' and try doing 'man passwd'.  
Error!  
Bcos only max 5 processes allowed !  
Check with 'ulimit -a'  
  
-----------------------------------------------------------------------------  
             foo                -     maxlogins       0  
  
 Now try logging in as foo. Not allowed !  
 'too many logins' error.  
  
-----------------------------------------------------------------------------  
             foo                -     maxlogins      1  
  
 Now try logging in as foo. Allowed ! From two ttys only!  
  
-----------------------------------------------------------------------------  
              foo              hard    fsize          50  
  
  Now foo can create a file of max 50 KB only  
-----------------------------------------------------------------------------  
  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
                        /etc/security/access.conf  
///////////////////////////////////////////////////////////////////////////////  
///////////////////////////////////////////////////////////////////////////////  
  
-:foo:ALL EXCEPT tty6  
  
 DO not allow foo to login from any tty except from tty6  
 + means the opposite -> to allow  
-----------------------------------------------------------------------------  
                              
      ============  
                             Exercise III  
                             ============  
  
    Examine /etc/pam.d/login and do this  :  
  
a. Change the 2nd line from required to sufficient ?  
  
Q: Now can root and foo login ?  
  
   Now do this :  
  
b. Delete /etc/securetty and touch /etc/nologin ?  
  
Q: Now can root and foo login ?  
  
#%PAM-1.0  
auth       sufficient pam\_securetty.so <-------- nbsp="" p="">auth       required pam\_stack.so service=system-auth  
auth       required pam\_nologin.so  
account    required pam\_stack.so service=system-auth  
password   required pam\_stack.so service=system-auth  
session    required pam\_stack.so service=system-auth  
session    optional pam\_console.so  
  
      ===========  
                             Exercise IV  
                             ===========  
  
1. Create user foo with passwd 'x'  
2. login as foo  
3. Change the passwd to 'x'  
   It will ask 3 times bcos of this and will o/p 'UNIX':  
  
   /etc/pam.d/system-auth  
   ======================  
   ....  
   ....  
   password    required      /lib/security/$ISA/pam\_cracklib.so retry=3 type=  
   ....  
   ....  
  
4. Remove the "retry" arg and change type=Windows  
  
5. Relogin as foo and now change the passwd and give some wrong passwd :  
  
   It will prompt you only ONCE and that too for a Windows passwd  
  
      ===========  
                             Exercise V  
                             ===========  
  
 Q: Login as root  
  
    # su foo  
  
    root is never asked for a password when she su's to another user  
  
    Why?  
  
A: BCos you are root ?  
   Wrong!  
  
   # ldd `which su`  
     libpam.so.0 => /lib/libpam.so.0 (0x40034000)  <------- is="" p="" pam-aware="" su="">     libpam\_misc.so.0 => /lib/libpam\_misc.so.0 (0x4003c000)  
     .....  
   The su\* binary is PAM-aware!  
   Hence su\* has to have a PAM config file in /etc/pam.d/ to consult  
   for authentication  
  
   Now, the config file name is the same as the service name [su\*]  
   hence su\* consults this file  
   /etc/pam.d/su  
   =============  
#%PAM-1.0  
auth       sufficient   /lib/security/$ISA/pam\_rootok.so  
# Uncomment the following line to implicitly trust users in the "wheel" group.  
#auth       sufficient   /lib/security/$ISA/pam\_wheel.so trust use\_uid  
# Uncomment the following line to require a user to be in the "wheel" group.  
auth       required     /lib/security/$ISA/pam\_wheel.so use\_uid  
.......  
.......  
session    optional /lib/security/$ISA/pam\_xauth.so  
  
   Now do this :  
    
   \* Change sufficient in Line 1 to required  
   \* Uncomment the "pam\_wheel.so use\_id"  
  
   Now try this :  
  
   # Login as root  
   # su - foo  
  
     And u will be asked for the passwd even if you are root  
  
   Now try this :  
  
   # Login as foo  
   # su -  
  
     And even if you give the right password for root you will not  
     be able so become root!  
  
     Actually no one can even become root now  
  
  Solution :  
  
   # usermod -G wheel foo  
  
     We make foo a member of system group wheel which the wheel module  
     consults!  
  
     Now foo ONLY will be able to su to root and no one else  
   # su -  
   #  
   Success  
  
   This increases security when root logs in remotely as a user and su's  
   to become root on the local system  
  
      ===========  
                             Exercise VI  
       Downloading and Compiling your own PLM from source code  
           see www.kernel.org/...  
                             ===========  
  
   On the 4th CD in the workingconfigs/pam/ dir, install the following RPM :  
  
        rpm -ivh pam\_alreadyloggedin-0.3-alt1.i586.rpm  
 or  
 pam\_alreadyloggedin-0.3.tar.gz  
  
 This will install the following :  
   
    pam\_alreadyloggedin.so in /lib/security/       Right Place!  
           pam\_alreadyloggedin.8     /usr/share/man/man8  Right Place!  
    login.sso                 /etc/pam.d           Example file!  
  
   \* Examine your /etc/pam.d/login :  
      
         #%PAM-1.0  
         auth       required pam\_securetty.so  
=====>   auth     sufficient pam\_alreadyloggedin.so  
         auth       required pam\_stack.so service=system-auth  
         auth       required pam\_nologin.so  
         .........  
         .........  
  
    \*   Now login as root on a tty  
  
       The next time you login as root again you will not be asked for the  
       password, since you are already logged in!  
    
CheckList :  
=========  
  
1 ldd command  
2 Hacking with #% , renaming pam\_unix.so etc etc  
3 /etc/pam.d/passwd called when you change passwd with passwd\*  
  since # ldd `which passwd`  
  so why is there a password module type in /etc/pam.d/login?  
  Furthermore, /etc/pam.d/login is never called and has nothing to do with  
  changing passwd on the CLI since the files which are concerned there are  
  /etc/pam.d/passwd and system-auth?  
  A: When your a/c is expired or disabled then ONLY does this module  
     come into play and prompts you for password  
4 In Single User Mode Login, no login is called, hence no PAM  
  is invoked  
5 Do Login Challenges in Essentials/PAM/  
  
                     \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
  
#################################################################################  
    PAM stands for Pluggable Authentication Modules.PAM is based on a series of library modules.Locations of PAM configuration files and library modules are:  
  
                 \* All PAM applications are configured in the directory "/etc                        /pam.d" or in a file "/etc/pam.conf".  
  
                \* The library modules are normally stored in the directory "/lib                  /security".  
  
                 \* The configuration files are located in the directory "/etc/se  
                   curity"  
  
  
    To configure PAM, on systems  you would need to edit the files for the service  
 you want to modify in the "/etc/pam.d" directory, and modify the appropriate        configuration file in the directory "/etc/security".  
  
  
      Some PAM module's behavior is controlled with configuration files (in /etc/security)as listed below:  
  
    \* access.conf - Login access control. Used for the pam\_access.so library.  
    \* group.conf - Group membership control. Used for the pam\_group.so library.  
    \* limits.conf - Set system resource limits. Used for the pam\_limits.so library.  
    \* pam\_env - Control ability to change environment variables. Used for the                      pam\_env.so library.  
    \* time - Allows time restrictions to be applied to services and user privileges            Used for the pam\_time.so library.  
  
  
            The configuration for PAM is normally in the /etc/pam.d directory which has a file for each PAM controlled application. This file or directory is used to control the behavior of applications that use the PAM modules  
  
                PAM is controlled using the configuration file /etc/pam.conf or the configuration directory, but not both.  
  
  
        A general configuration line in one of the PAM application configuration file has the following form:  
  
module-type   control-flag   module-path   arguments  
  
    1. module-type - The type name of the PAM module used which are  
  
        1. auth - Authenticates the user,usually asking a password then checking                 it.  
  
        2. account - Check to see if the authentication is allowed based on avai                    lable system resources such as the maximum number of users                       or the location of the user.  
  
       3. password - Used to set passwords  
  
        4. session - Used to make it possible for a user to use their account on                     ce they have been authenticated  
  
  
  2. control-flag:-  
       
       1. required - The success of the module is required for the module-type                       facility to succeed. It means it must succeed, if not furth                     er modules are still rxecuted.  
  
       2. requisite - If the module returns a failure, control is directly retur                      ned to the application. It means must succeed , if not no                      other furthe modeules are checked of same types.  
     
       3. sufficient - If this module succeeds and no previous required module h                       as failed, no more `stacked' modules of this type are                           invoked.  
                       It means if this module succeed, no further modules of                           same type are called.  
  
      4. optional - This module is not critical to the success or failure of the  
                   user's application for service.  
  
  3. module-path - The path and filename of the PAM library used to control the                       function.  
   4. arguments - Arguments are optional and vary from module to module.  
  
        
  
My "/etc/pam.d/login" file looks like this:  
  
#%PAM-1.0  
auth       required     /lib/security/pam\_securetty.so  
auth       required     /lib/security/pam\_pwdb.so shadow nullok  
auth       required     /lib/security/pam\_nologin.so  
account    required     /lib/security/pam\_pwdb.so  
password   required     /lib/security/pam\_cracklib.so  
password   required     /lib/security/pam\_pwdb.so nullok use\_authtok md5 shadow  
session    required     /lib/security/pam\_pwdb.so  
session    optional     /lib/security/pam\_console.so