
# Autenticação PAM

# Sumário



# Índice de tabelas

# Índice de Figuras

# Capítulo 1

# Autenticação PAM

•Formas de autenticação que trabalham com o PAM;

•Descrição dos principais arquivos de configuração.

# 1.1. Mãos a obra

A autenticação de usuários no Linux é feita por padrão, através dos aquivos passwd e shadow ambos encontrados no diretório /etc. O responsável em fazer essa tarefa é o programa login, vamos acompanhar o passo a passo:

1 - O usuário digita o seu nome e o programa login verifica se o nome existe no arquivo passwd;

2 – Caso o usuário exista, é verificado que o próprio tem uma senha no arquivo shadow;

3 – Neste ponto o programa login pede a senha ao usuário, que a digita;

4 – Então a senha é criptografada e comparada com a senha no arquivo shadow;

5 – Se a comparação der sucesso o usuário tem acesso ao sistema.

*Como posso alterar o funcionamento de autenticação de meus usuários?*

Isso é possível através do PAM (Pluggable Authentication Modules) que nada mais é que uma suíte de bibliotecas, onde adicionadas e configuradas alteram a maneira que as aplicações autenticam os usuários no sistema.

Um exemplo é o programa login que pode trabalhar com o PAM, e assim usar uma base remota de usuários, aos invés de usar os arquivo passwd e shadow.

Vamos ver algumas informações relevantes ao PAM.

- Suporte vários tipos de autenticação;
- Possui mecanismos de controle de login;

- Trabalha com controles e módulos;
- Desenvolvimento inicial 1996 pela Sun Microsystems;
- Pode ser utilizado em AIX /HP-UX /Solaris /Linux /FreeBSD /Mac OS X /NetBSD
- Temos também Linux-PAM, OpenPAM e Java PAM ou JPam

*Como posso descobrir se um determinado programa tem suporte ao PAM?*

Isso pode ser feito de duas maneiras, a primeira verificando se o programa em questão tem em sua lista de bibliotecas a libpam. Vamos a prática:

*# ldd $(which login) | grep libpam*

```
        libpam.so.0 => /lib/libpam.so.0 (0xb7fa2000)
        libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb7f9f000)
```

Veja que em nosso exemplo o comando ldd foi utilizado para listar as bibliotecas do programa login. Veja na lista a biblioteca libpam.so.0 do PAM.

A outra maneira é listar o conteúdo do diretório /etc/pam.d, onde cada arquivo é usado para configurar o PAM em cada aplicação. Vamos a prática:

```
atd    common-account   common-session  gdm                 login   samba  sudo
chfn   common-auth      cron            gdm-autologin       other   sshd
chsh   common-password  cups            gnome-screensaver   passwd  su
```

Cada arquivo é criado conforme a instalação das aplicações, no exemplo temos o ssh, samba e gdm, isso que dizer que essas aplicações estão instaladas.

Configuração do PAM

A configuração do PAM pode ser feita de duas maneiras, uma através do arquivo /etc/pam.conf onde você pode centralizar a configuração de todas as aplicações, ou através de arquivos individuas no diretório /etc/pam.d. Vamos a um exemplo prático:

Programa login configurado através do arquivo /etc/pam.conf

```
# vim /etc/pam.conf
```

```
# ------------------------------------------------------------------------#
# /etc/pam.conf                                                           #
# ------------------------------------------------------------------------#
#
# NOTE
# ----
#
# NOTE: Most program use a file under the /etc/pam.d/ directory to setup their
# PAM service modules. This file is used only if that directory does not exist.
# ------------------------------------------------------------------------#

# Format:
# serv. module  ctrl            module [path]   ...[args..]               #
# name  type    flag                                                      #
login   account requisite       pam_time.so
```

Programa login configurado através de um arquivo individual no diretório /etc/pam.d

```
# vim /etc/pam.d/login
```

```
# Uncomment and edit /etc/security/time.conf if you need to set
# time restrainst on logins.
# (Replaces the 'PORTTIME_CHECKS_ENAB' option from login.defs
# as well as /etc/porttime)
account    requisite  pam_time.so
```

**Módulos do PAM**

Os módulos implementam funções para cada etapa que acontece em uma autenticação feita no sistema. Veja a descrição de cada módulo.

**auth** → Usado para autenticar usuários;

**account** → Usado para gerenciar a conta do usuários, como por exemplo expiração de conta e se o usuário pode acessar algum serviço;

**password** → Responsável por cuidar dos aspectos relacionados a tarefas envolvendo senhas (senha fraca ou pertence ao algum dicionario);

**session** → Responsável por tarefas antes e depois que o usuário for autenticado, com por exemplo dispositivos de hardware, diretório pessoal e sistema de arquivos remotos;

**Controle do PAM**

Os controles indicam à biblioteca PAM como reagir em caso de sucesso ou falha. Veja a descrição de cada controle.

**required** → Falha somente ao final do processo de autenticação;

**requisite** → Falha durante o processo de autenticação;

**sufficient** → Somente sua existência é suficiente para a autenticação;

**optional** → Sua falha não interfere no processo de autenticação.

As bibliotecas do PAM estão localizadas em /lib/security. Vamos a prática:

```
# ls /etc/security
```

```
pam_access.so      pam_keyinit.so    pam_permit.so       pam_time.so
pam_debug.so       pam_lastlog.so    pam_rhosts_auth.so  pam_umask.so
pam_deny.so        pam_limits.so     pam_rhosts.so       pam_unix_acct.so
pam_echo.so        pam_listfile.so   pam_rootok.so       pam_unix_auth.so
pam_env.so         pam_localuser.so  pam_securetty.so    pam_unix_passwd.so
pam_exec.so        pam_loginuid.so   pam_selinux.so      pam_unix_session.so
pam_faildelay.so   pam_mail.so       pam_sepermit.so     pam_unix.so
pam_filter.so      pam_mkhomedir.so  pam_shells.so       pam_userdb.so
pam_ftp.so         pam_motd.so       pam_stress.so       pam_warn.so
pam_group.so       pam_namespace.so  pam_succeed_if.so   pam_wheel.so
pam_issue.so       pam_nologin.so    pam_tally.so        pam_xauth.so
```

**Veja a descrição de algumas bibliotecas:**

**pam_access.so** → Controle de acesso de login dependendo das regras predefinidas no arquivo /etc/security/access.conf;

**pam_cracklib.so** → Verifica as senhas em relação às regras de senha;

**pam_env.so** → Verifica as variáveis de ambiente a partir de /etc/security/pam_env_conf;

**pam_debug.so** → Depura o PAM;

**pam_deny.so** → Bloqueia algum módulo do PAM;

**pam_echo** → Imprime mensagens;

**pam_exec.so** → Executa um comando externo;

**pam_ftp.so** → Módulo para acesso anônimo;

**pam_localuser.so →** Requer que o usuário seja listado em /etc/passwd;

**pam_unix.so →** Fornece autenticação de senha tradicional de /etc/passwd;

**pam_nologin.so →** Com a existência do arquivo /etc/nologin impede que qualquer usuário, com exceção do root faça login no console;

**pam_time.so →** Controle de acesso de login por horário dependendo das regras predefinidas no arquivo /etc/security/time.conf;

**pam_limits.so →** Limita o uso de recursos do sistema através das regras predefinidas no arquivo /etc/security/limits.conf.

**Formas de autenticação que trabalham com o PAM**

O PAM usa como padrão o pam_unix.so para autenticar os usuários através dos arquivos /etc/passwd e /etc/shadow. Em cada arquivo de configuração presente em /etc/pam.d, é incluído um ou mais arquivos usados para cada módulo o PAM. Veja um exemplo.

```
# tail -5  /etc/pam.d/login
```

```
# Standard Un*x account and session
@include common-account
@include common-session
@include common-password
```

Os arquivos que iniciam com common no diretório /etc/pam.d são utilizados para cada módulo do PAM. Veja a lista dos quatros arquivos.

```
# ls -l  /etc/pam.d/common-*
```

```
-rw-r--r-- 1 root root  392 Fev 24  2010 /etc/pam.d/common-account
-rw-r--r-- 1 root root  436 Fev 24  2010 /etc/pam.d/common-auth
-rw-r--r-- 1 root root 1212 Fev 24  2010 /etc/pam.d/common-password
-rw-r--r-- 1 root root  372 Fev 24  2010 /etc/pam.d/common-session
```

**common-account** → Sua configuração é incluída em todos os serviços que utilizam o módulo account.

**common-auth** → Sua configuração é incluída em todos os serviços que utilizam o módulo auth.

**common-passwd** → Sua configuração é incluída em todos os serviços que utilizam o módulo passwd.

**common-session** → Sua configuração é incluída em todos os serviços que utilizam o módulo session.

Então chegamos a conclusão que a alteração do conteúdo desses arquivos, muda a forma de autenticação no sistema. Veja alguns exemplos:

**Autenticação usando como base arquivos no sistema**

auth required pam_unix.so

**Autenticação usando como base um servidor de diretório LDAP**

auth sufficient pam_ldap.so

**Autenticação usando como base um servidor AD da Microsoft**

auth sufficient pam_winbind.so

Além das configurações dos quatros arquivos, é preciso também editar o arquivo /etc/nsswitch.conf usado pelo sistema para definir a ordem em que as bases de dados serão consultadas. Exemplo:

```
# vim /etc/nsswitch.conf
```

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.

passwd:         compat ldap winbind
group:          compat ldap winbind
shadow:         compat ldap winbind
```

No exemplo acima nomes de usuários, grupos e senhas serão consultadas primeiramente em compat (arquivos), depois em uma base ldap (servidor de diretório) e por ultimo usado o winbind (servidor AD da Microsoft).

# Capítulo 2

# Gerenciando

# 2.1. Objetivos

•Trobleshooting: Configurar o PAM com o SSH.

# 2.1. Troubleshooting

*Como posso configurar o acesso ao SSH usando PAM?*

Através de alguns módulos do PAM é possível limitar o acesso remoto via SSH ao usuários por data e hora, lista de quem pode ou não acessar e bloqueio de terminais. Vamos a prática:

Primeiro instale o ssh e verifique o suporte ao PAM

```
# aptitude install ssh
```

```
# ldd $(which sshd) | grep libpam
```

```
        libpam.so.0 => /lib/libpam.so.0 (0xb7ec7000)
```

**Cenários de uso do PAM com SSH**

Vamos ver dois exemplos práticos de uso das bibliotecas do PAM e seus arquivos de configuração. O arquivo principal do SSH esta localizado em /etc/pam.d/sshd, é através dele que configuramos os módulos, controles e argumentos do PAM.

Exemplo 1: Limitar acesso por horário

1 – Abra o arquivo /etc/pam.d/sshd

```
# vim /etc/pam.d/sshd
```

2 – Adicione a linha abaixo:

```
account    required     pam_time.so
```

3 – Abra o arquivo de configuração do módulo em /etc/security/time.conf, e configure os dias e horários que os usuários poderão realizar acesso via SSH. Exemplo:

```
ssh;tty1;root;Al0800-1800
```

A sintaxe de configuração do arquivo é:

serviço;terminal;usuários;data e hora

Em nosso exemplo o serviço de SSH poderá ser acessada pelo usuário root, todos os dias das oito da manhã até as dezoito horas, apenas no primeiro terminal.

Veja abaixo a descrição das abreviações aceitas para data e hora:

**Su** → Domingo;

**Mo** → Segunda;

**Tu** → Terça;

**We** → Quarta;

**Th** → Quinta;

**Fr** → Sexta;

**Sa** → Sábado;

**Wk** → Finais de semana (Sábado e Domingo);

**Wd** → Dias da semana (Segunda a Sexta);

**Al** → Todos os dias.

Exemplo 2: Bloquear acesso aos usuários usando com base uma lista

1 – Abra o arquivo /etc/pam.d/sshd

```
# vim /etc/pam.d/sshd
```

2 – Adicione a linha abaixo:

```
auth  required  pam_listfile.so  item=user  sense=deny  file=/etc/ssh/sshd.deny
onerr=succeed
```

Vamos a descrição dos argumentos utilizados:

**item** → Usado para verificar o nome de usuário;

**sense** → Nega o acesso se usuário não existir no arquivo especificado;

**file** → Nome do arquivo que contém a lista de usuário;

**onerr** → Indica sucesso no login, caso o nome exista na lista de usuários.

3 – Crie o arquivo com a lista de usuários bloqueados

```
# vim /etc/sshd/sshd.deny
```

```
tux
maria
joao
```