

Falando sobre senhas no Linux

By [admin](#) on 10 de julho de 2012 in [Linux](#)

7 Flares Twitter 1 Facebook 4 Google+ 1 LinkedIn 1 Email -- Filament.io 7 Flares [×](#)

Neste Post falarei um pouco sobre as senhas do Linux usando o padrão que já vem instalado, no caso o PAM (Pluggable Authentication Modules), através do passwd, group e shadow.

Muitos amigos, e administradores vêm até mim, pedindo informações de como alterar a senha do root, seja por pegar um servidor novo, ou por outro administrador alterar sem consultar com antecedência. Então neste Post mostrarei como fazer isso, através de mudança de runlevel do Grub, e de Mídia Bootável, como um CD-ROM do Debian, e neste caso usando o jail (chroot) ou a simples mudança de um arquivo passwd ou shadow.

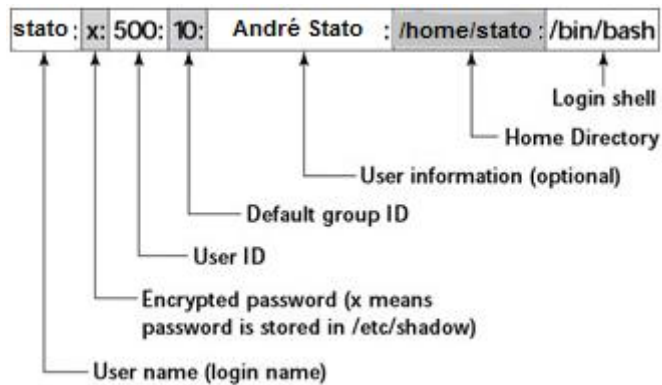
Mas não somente isso abordaremos como por criar usuários em lote, quais os campos do passwd, group e shadow, de forma que fiquemos mais familiarizados com tais arquivos.

Apesar de ser um tópico básico, é de extrema importância para o dia-a-dia de nossa administração.

Os arquivos de autenticação

Os arquivos no Linux responsáveis pela autenticação dos usuários são: passwd, group e shadow. Sendo:

- passwd – O arquivo com o cadastro do usuário, ele possui os seguintes campos
 - login, senha (*), Uid, Gid, Gecos(*), Home, Shell
 - * A senha antigamente era colocada em clear text, ou seja, texto puro, considerada hoje em dia insegura. Atualmente esse campo possui uma letra “x”, informando que está senha está criptografada no shadow. O campo Gecos, também conhecido como comentário, é utilizado para comentários, nada a mais. O campo home armazena o local onde será armazenado o diretório pessoal do usuário e por fim o shell, informa qual o shell padrão do usuário. Este campo também pode ser conhecido como command, e não obrigatoriamente deve ter um shell, poderá ter um path para um comando, ou até um prompt falso, como /bin/false.
 - Ex:



- `group` – O arquivo de cadastro de grupo, bem mais simples que o anterior, possui poucos campos sendo:
 - nome do grupo, senha (`x` , como no arquivo `passwd` usa o `gshadow`, `Gid`). Os próximos campos são os nomes dos usuários que fazem parte do grupo separado por vírgula.
 - `shadow` – Este arquivo é responsável por armazenar as senhas dos usuários, e um dos mais importantes. Vejamos os campos:
 - nome do usuário, senha criptografada, dias desde 1 de Janeiro de 1970 que a senha foi alterada, dias necessários para que a senha possa ser alterada, dias para alteração da senha, dias para aviso da troca da senha, dias que a conta está desabilitada desde que a senha expirou , dias desde 1 de Janeiro desde 1970 que a senha expirou e campo reservado, dando um total de nove campo.

Então temos aqui os principais arquivos relacionados à autenticação padrão do Linux usando o Pam. Lembrando que existem outros modelos de autenticação como Ad através do Winbind ou Ldap, Samba com Winbind, Ldap, Mysql, enfim, muitos outros, mas todos passam logicamente pelo Pam. O Pam é o centralizador de autenticação, de forma que, mesmo que usando outro método, que gerenciara e centralizará será ainda o Pam.

Não entraremos em detalhes do mesmo, já que o intuito não é aprendermos detalhes sobre como o Pam funciona, mas sim como tirar proveito em algumas situações como resetar senhas, quando não sabemos.

Resetando senhas

Bom sabemos que as senhas podem ficar armazenadas em dois locais, no arquivo `/etc/passwd` (isso antigamente) no 2º campo, logo após o nome do usuário, ou ainda no `/etc/shadow`, também localizado no 2º campo.

Tempos então duas possibilidades num primeiro momento. Podemos apagar o valor do campo senha do passwd, ou a senha criptografada do shadow.

No arquivo /etc/passwd, atualmente encontraremos o valor “x”, informando que a senha está criptografada e localizado no /etc/shadow. Mas uma vez que ela esteja vazia, ou seja, sem o “x”, o Linux não irá procurar nenhuma senha, e já cairá no prompt de comando, se logicamente o campo 7 (shell) permitir.

O grande problema é: Como chegar até lá, já que não tenho a senha do Administrador root?

Obviamente se você já tivesse a senha do root, usaria o comando “**passwd**” , para gerar uma nova senha, não é? rsrs

Neste caso temos duas, ou três opções. Uma opção é iniciar um nível de inicialização diferente através do GRUB (Gerenciador de Boot padrão do Linux na maioria das distribuições) se o mesmo não possuir senha para travar essa alteração. E a segunda opção é iniciar outro sistema através de um CD/DVD Live, ou até atachando o HD em outra máquina com outro Linux. Falaremos de ambas.

Vamos ao primeiro procedimento.

Resetando a senha usando um Runlevel diferente no Grub

O primeiro passo é iniciar o Linux e escolher qualquer um dos Kernels que tenham acesso ao file system (Sistema de arquivos) onde o arquivo /etc/passwd está localizado. Então nesta primeira tela do grub deveremos escolher o kernel e digitar a letra “e”.
Veja a Tela Inicial antes de digitarmos qualquer coisa:

```
GNU GRUB version 0.97 (638K lower / 522176K upper memory)

Debian GNU/Linux, kernel 2.6.26-1-686
Debian GNU/Linux, kernel 2.6.26-1-686 (single-user mode)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

Tela padrão do Grub

Após selecionar o kernel, devemos então digitar a letra “e”, que significa **Editar**, para podermos editar o nível de inicialização que queremos.

```
GNU GRUB version 0.97 (638K lower / 522176K upper memory)

root (hd0,0)
kernel /boot/vmlinuz-2.6.26-1-686 root=/dev/sda1 ro quiet
initrd /boot/initrd.img-2.6.26-1-686

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

Podemos ver na figura acima que a tela possui três linhas, a primeira faz referencia a partição onde está a nossa partição de boot, a segunda onde se encontra nosso kernel (Essa é a mais importante para nós), e a terceira onde se localiza nossa imagem inicial.

Devemos então alterar a segunda linha, adicionando qual será o novo nível de inicialização ao final do texto já existente.

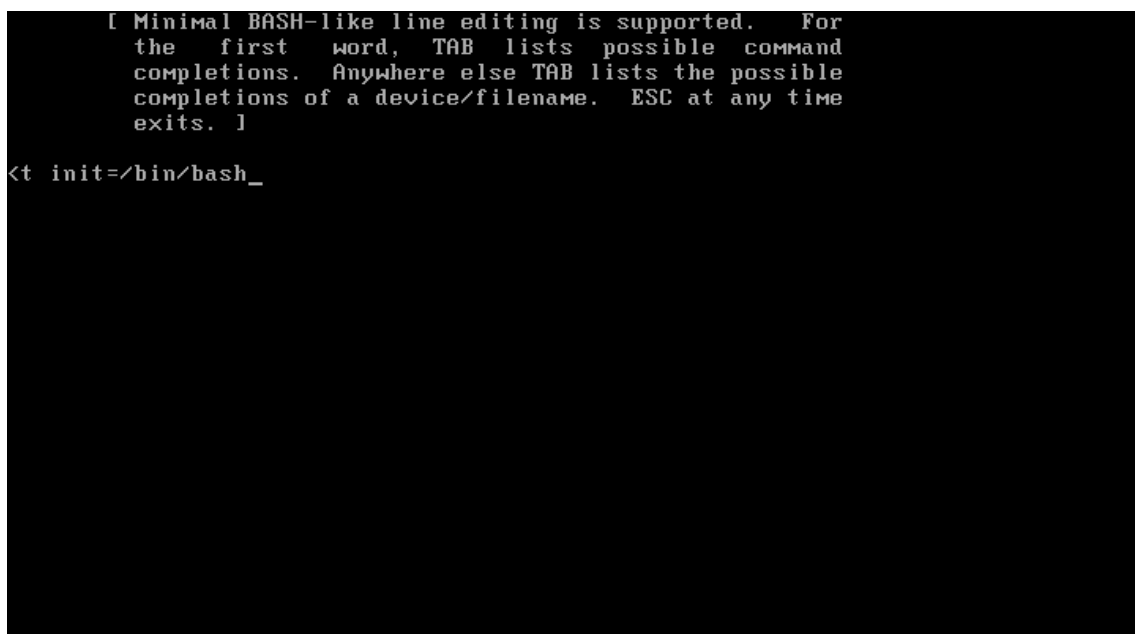
Através do valor **init**, podemos especificar qual o runlevel que desejamos iniciar nesse momento. Ao final deste deveremos adicionar:

- **init=/bin/bash**

Para podermos adicionar, deveremos novamente “editar”, e para isso deveremos novamente teclar a letra “e” do teclado. Após isso, entraremos em modo edição na linha do kernel, e após isso poderemos fazer a nossa alteração.

Devemos tomar cuidado, pois algumas teclas padrões não irão funcionar inclusive a barra (/) do teclado padrão. Somente a barra, do teclado numérico, provavelmente funcionará.

Após inserir esses dados, a tela será parecida com a tela abaixo.

A screenshot of a terminal window with a black background and white text. The text reads: "[Minimal BASH-like line editing is supported. For the first word, TAB lists possible command completions. Anywhere else TAB lists the possible completions of a device/filename. ESC at any time exits.]". Below this, the prompt "<t init=/bin/bash_" is visible, indicating that the user has entered the command to set the root shell to /bin/bash.

Finalizando o processo, basta teclar enter, e voltaremos para a tela anterior, mas com os dados já alterados. Agora basta iniciarmos com essa alteração.

É muito importante agora prestar atenção, para não digitar “Enter” e iniciar normalmente.

Para iniciarmos com as opções solicitadas, devemos digitar a tecla “B”, que significa Boot. A partir daí ele iniciará o boot, e finalizará no prompt de comando, já como super-usuário, ou seja , como root.

Mas ainda sim, outros procedimentos são necessários, já que o sistema está em modo somente leitura, dessa forma é necessário alterar para modo gravação, para podermos alterar a senha.

Vejamos os procedimentos abaixo.

```
mount -o remount,rw /passwd
```

Com o comando mount, estamos remontando o sistema no modo read-write. Após isso usando o comando o passwd, estamos alterando a própria senha do root. Para finalizar basta alterar novamente o sistema para somente leitura (alterar o comando mount o rw para ro) e enfim rodar o comando reboot.

É possível apagar a letra “x” do campo senha do arquivo passwd, bem como a senha do shadow, mas isso deixarei para utilizar na próxima forma...

Resetando a senha usando outros sistemas Linux

Neste caso, tanto faz se você vai usar um Live CD/DVD (Sistema Bootável através de uma mídia), ou um sistema já instalado em uma máquina. Em ambos os caso podemos usar duas formas de recuperação da senha:

- Entrando no ambiente chroot e alterando a senha
- Alterando o arquivo passwd ou shadow

Mas em ambos os casos deveremos acessar a partição, montar o sistema, para então tomar a decisão de qual das duas acessar.

Existe ainda a possibilidade de usar o “Modo Rescue” de algumas distribuições, como do Debian por exemplo. Apesar de ser funcional em ambos os casos, no primeiro caso, pode ser um pouco complicado, pois os dispositivos de hardware como HD sata, ou scsi, não são criados automaticamente, de forma a forçar que você os crie manualmente usando o comando **mknod**. Gerando um pouco mais de trabalho, por isso, recomendo os dois citados anteriormente.

Como em ambos os casos, deveremos montar o sistema, os passos abaixo, servirão para os dois casos:

```
mkdir /mnt/linux
```

```
mount /dev/sdb1 /mnt/linux
```

Os comandos acima foram utilizados para acessar a primeira partição do disco em questão, no caso o sdb, que é o segundo disco da minha máquina. É bem provável que se você estiver usando um Live CD, este seja um sda, ou seja, o primeiro disco, e não o segundo. No primeiro comando criamos o ponto de montagem e no segundo comando estamos montando propriamente dito a partição onde está a nossa raiz. Então é importante sabermos de antemão onde se encontra nosso disco e partição. Na dúvida execute o **dmesg**.

Os próximos passos dependerão da forma que você escolher. Sem sombra de dúvida apagar parte dos arquivos /etc/passwd ou /etc/shadow é bem mais fácil que qualquer outra coisa. Então vamos começar com esse. Vamos pegar como exemplo o arquivo /etc/passwd, com a seguinte entrada:

```
root:x:0:0:root:/root:/bin/bash
```

Devemos alterar para:

```
root:0:0:root:/root:/bin/bash
```

No caso acima, só removemos o valor do segundo campo, o “x”.

No arquivo /etc/shadow, também devemos remover o segundo campo, que é onde fica a senha criptografada.

```
root:$1$DPgV.H67$HEMDjPGFX0ZW36YTfM.b6/:11326:0:99999:7:::
```

Nesse caso acima, temos o segundo campo em negrito, é onde fica a senha, basta alterarmos, melhor apagar, para que possamos logar sem senha.

```
root::11326:0:99999:7:::
```

Com uma dessas alterações já é o suficiente, para fazer logon sem uso de senha.

A segunda forma usando outro sistema , é montando o sistema como foi feito anteriormente, e depois disso informando ao sistema que a raiz (/) do nosso sistema é o próprio ponto de montagem, sendo assim quando solicitarmos a troca da senha, será feita a troca em nosso sistema externo, e não na máquina que está sendo usada. Veja abaixo os comandos:

```
chroot /mnt/linux
```

```
passwd
```

```
...xit
```

Com essa alteração, basta iniciar o sistema , que nossa senha já estará alterada. É de bom grado, rsrs, verificar os valores do shadow, só para ter certeza que foram alterados.

Usando o John The Ripper para recuperar uma senha

Existe ainda uma terceira possibilidade, mas não se trata de alterar uma senha, e sim descobrir essa senha, através de um software chama John the Ripper. Esse software é capaz de descriptografar senhas não tão complexas, com certa facilidade.

O processo em si é bem simples , bastando instalar , executar o John apontando para o shadow.

```
#apt-get install john
```

```
#john /etc/shadow
```

```
Created directory: /root/.john
```

```
Loaded 2 password hashes with 2 different salts (FreeBSD MD5 [32/32])
```


Obviamente quanto mais complexa a senha mais demora, pois se trata de um brute force , ou seja, tentativa e erro. O John suporte dicionário, se você possuir um, pode acabar agilizando em muito esse processo de descoberta de senha.

Trabalhando com senhas no Linux

Nesta ultima parte do Post, vou falar um pouco sobre as senhas em si. Muitos administradores gostariam de automatizar o processo de criação de usuários, e talvez por não conhecerem o tipo de autenticação, ou até algumas ferramentas, não o fazem.

Vimos acima que os campos do `/etc/shadow` são :

- Login Nome do usuário
- Senha Senha criptografado usando algoritmo MD5
- Dias 1970 Alteração Dias desde 01 de Janeiro de 1970 desde a última alteração da senha
- Mínimo Troca Dias para que o usuário tenha permissão para alterar a senha desde a última alteração
- Máximo Troca Dias máxima para troca de senha desde a última alteração
- Aviso Dias para que o usuário comece a ser avisado que deverá trocar a senha
- Desabilitado Dias que a conta está desabilitada, desde que a senha expirou
- Dias 1970 Dias que a conta está desabilitada, desde 1 de Janeiro de 1970
- Reservado Campo Reservado

Destes campos os mais importantes , são a senha obviamente , Mínimo, Máximo, Aviso, e talvez a data de alteração dada acima como Dias 1970 Alteração.

Este valores indicam o seguinte :

- O mínimo , diz o tempo mínimo que o usuário tem para trocar a senha, logo após ter alterado, ou seja, se ele acabou de trocar a senha, quando poderá trocar a senha novamente.
- O máximo , de quanto em quanto tempo o usuário poderá alterar a senha.
- Aviso, ou Warning, é a o tempo que antecede o dia de troca de senha, padrão 7 dias, onde será exibida a mensagem que o usuário deverá trocar a senha

- Dias alteração, é através deste campo que o Linux verifica se é necessário alterar a senha, são os dias passados desde 01 de Janeiro de 1970. Sinistro!!!

Um modo fácil de controlar é através do comando “**chage**”, onde podemos passar esses valores com parâmetros , temos os seguintes parâmetros :

- -M Máximo Dias para troca de senha
- -m Mínimo Dias para troca de senha
- -W Dias para aparecer à mensagem de Warning
- -d Data da ultima troca de senha no formato YYYY/MM/DD

Um exemplo Simples:

```
#chage -M 90 -m 20 -W 5 -d 2012-06-30 stato
```

```
#passwd -S stato
```

```
stato P 06/30/2012 20 90 7 -1
```

Veja que no exemplo acima, na data usei hífen ao invés de barra. Para o comando é indiferente, tanto faz um ou outro. Já o comando “**passwd -S**”, traz o Status da conta, que neste caso, nos mostra que a senha está funcional (P) caso contrário encontraríamos um L de lock, que a senha foi trocada em 30/60/2012, e deverá ser trocada a cada 90 dias, com intervalo mínimo de 20 dias , e mensagem de Warning 5 dias antes de expirar a senha. O ultimo campo refere-se à quantidade de dias que a conta está desabilitada depois de expirada, que não é o caso acima.

Se olharmos o shadow , veremos muita similaridade:

```
getent shadow statostato:$1$BpUjmeEN$gbzwGR842.MjadFZrYidR.:15521:20:90:7:::
```

O campo 4,5,6 trazem justamente essa informação, tempo mínimo, tempo máximo, e warning. Podemos deixar um padrão através do arquivo “**/etc/login.defs**”, alterando as variáveis :

PASS_MAX_DAYS

PASS_MIN_DAYS

PASS_WARN_AGE

Alterando esses valores, qualquer usuário criado novo terá como padrão, tais valores. Mas ainda resta um problema: a senha. De qualquer forma ainda tenho que gerar uma senha como o comando passwd, e ainda por cima, me pede confirmação.

Não podemos esquecer o campo 3, que é a data da última troca de senha, mas se estamos criando um novo usuário ele é irrelevante, pois podemos zerar ele, para forçar uma troca de senha por exemplo.

Um comando que pode nos auxiliar nessa tarefa é o “**mkpasswd**”. Este comando retorna o valor de uma senha em vários formatos, mas para nós o que é importante é o algoritmo md5. Veja abaixo o comando :

```
mkpasswd -m md5 debian$1$CJQtw0io$.8PXGLT/qkvlduAyuMGeP/
```

Este valor retornado, pode ser usado diretamente dentro do shadow, como uma senha padrão. Então com alguns comandos podemos criar um usuário, usando uma senha padrão, de forma que ele altere a senha no primeiro login, e tenha que trocar a senha a cada 90 dias, com intervalo mínimo de 20 e warning de 7, vejamos como fica:

```
echo "joao:x:1010:1010:Usuario Joao:/home/joao:/bin/bash" >>/etc/passwd  
echo "joao:x:1010:" >>/etc/group  
  
echo "joao:`mkpasswd -m md5 debian`:0:20:30:5::" >>/etc/shadow  
  
mkdir /home/joao  
  
cp /etc/skel/* /home/joao  
  
usermod joao.joao /home/joao
```

No caso acima, criamos entradas no passwd, group, shadow, criamos o diretório do usuário, copiamos o skel, e alteramos a permissão do mesmo.

Lógico que deu mais trabalho que usar um simples useradd e um passwd, mas o intuito é a manipulação da ferramenta mkpasswd. Facilmente é possível criar um script, verificando o último UID e GID usado, para criação destes usuários....

E com certeza, vale o aprendizado, a manipulação de Hash MD5 no shadow.

Com este Post, acredito que desmistificamos um pouco do esquema de autenticação de senhas do Linux. Mas fica devendo ainda um Post somente PAM.

Até a próxima.

André Stato

