Segurança Informática e nas Organizações

Guiões das Aulas Práticas

João Paulo Barraca[1] e Hélder Gomes[2]

[1]Departamento de Eletrónica, Telecomunicações e Informática
[2]Escola Superior de Tecnologia e Gestão de Águeda
Universidade de Aveiro

2015–2016

Conteúdo

2

Segurança em sistemas operativos Linux

Resumo:

- Login, identificação de utilizadores e grupos.
- Limitações à interação entre processos.
- Utilizador `root`.
- Administração com `sudo`.
- Elevação de privilégios com o mecanismo set-UID.
- Proteção de ficheiros.
- Confinamento com `chroot`.
- Gestão de senhas de *login* com módulos PAM (*Pluggable Authentication Modules*).

2.1 Introdução

Com este trabalho pretende-se estudar os mecanismos de segurança elementares em sistemas operativos Linux. Neste trabalho deverá apenas ser usado um sistema Linux, quer instalado nativamente numa máquina quer numa máquina virtual. Pode ser usada uma distribuição Linux Live.

2.2 Login (entrada no sistema)

Para entrar no sistema é primeiro preciso fazer *login*. Para isso carregue simultaneamente nas teclas Ctl+Alt+F1 para mudar para uma consola de comandos[1] e faça *login* com as credenciais (nome e senha) que o sistema já reconhece.

Como resultado de um processo de *login* correto é criada uma sessão, que consiste na execução de um interpretador de comandos (*shell* na terminologia Linux).

2.3 Identidade

Execute o comando

```
id
```

Este comando indica a identidade subjacente a quem está a executar o interpretador de comandos. Analise os dados apresentados.

Execute o comando

```
cat /etc/passwd
```

Este comando lista o ficheiro que possui as *identidades individuais* reconhecidas pelo sistema operativo. Verifique no conteúdo apresentado a presença do utilizador com cujo nome fez *login*.

Execute o comando

```
cat /etc/group
```

Este comando lista o ficheiro que possui as *identidades de grupos* reconhecidas pelo sistema operativo. Verifique no conteúdo apresentado a presença do utilizador que está a usar em diversos grupos e confronte o que observa com o resultado apresentado pelo comando `id`.

[1] Antes de executar esse sequência de teclas pode ser necessário fazer Ctl+Alt+Espaço

2.4 Gestão de utilizadores

Execute o comando

```
adduser nome
```

para acrescentar o utilizador *nome* à lista dos reconhecidos pelo sistema. Como este comando requer privilégios de administração, não irá conseguir que o mesmo seja executado, recebendo uma mensagem de erro.

Execute o mesmo comando precedendo-o de outro, `sudo`, que permite alterar os privilégios de execução para os de um administrador:

```
sudo adduser nome
```

Ao executar este comando terá que introduzir a senha do utilizador que o está a executar, para verificar que é mesmo esse utilizador que o está a requerer. Introduza os elementos pedidos pelo comando para acrescentar o novo utilizador ao sistema.

Execute o comando

```
cat /etc/passwd
```

e procure a informação que introduziu respeitante ao novo utilizador.

Execute o comando

```
cat /etc/group
```

e procure a informação que introduziu respeitante ao novo utilizador.

Execute o comando

```
finger nome
```

usando como `nome` o nome do utilizador que criou e verifique a informação que introduziu respeitante ao novo utilizador.

2.5 Mudança de utilizador

Mude para outra consola[2], carregando simultaneamente nas teclas Alt+F2 (consola 2). Entre no sistema usando a identidade e a senha do utilizador

[2]Para mudar entre consolas de texto apenas é necessário as duas teclas, Alt e Fx, onde x representa o número da consola (de 1 a 6). A consola gráfica é obtida com Alt+F7 e, opcionalmente, uma segunda consola gráfica, normalmente inativa, com Alt+F8. Para mudar de uma consola gráfica para outra qualquer usa-se Ctl+Alt+Fx.

que criou no passo anterior. Execute o comando

```
id
```

para verificar a identidade do utilizador que está a executar o interpretador de comandos.

2.6 Interação entre processos

Na consola 2 execute o comando

```
ps -au
```

para listar todos os processos em execução nas consolas no sistema. Obtenha o PID (*Process ID*) do interpretador de comandos do utilizador da consola 1 e use-o para terminar o seu interpretador de comandos. Para isso deverá ser usado o comando

```
kill PID
```

onde `PID` representa o identificador antes obtido. Observe o resultado.

Repita o comando anterior usando o PID do interpretador de comandos do utilizador atual da consola 2. Observe o resultado.

Repita o comando anterior usando a opção `-9`:

```
kill -9 PID
```

Observe o resultado e explique o que aconteceu.

2.7 Utilizador `root`

O utilizador `root` é o *super-user* do Linux. A este utilizador nada é negado, tudo é permitido. Ele é o administrador omnipotente do Linux. O seu ID é 0 (por definição).

2.7.1 Comando `sudo`

O comando `sudo <comando>` permite executar o comando indicado como parâmetro com a identidade de `root`. Para que este aumento de privilégios seja autorizado a quem não é `root`, quem requer este comando precisa de introduzir a sua senha para verificar a sua identidade.

Volte à consola 1 com as teclas Alt+F1.

Execute o comando

```
sudo id
```

e verifique o resultado obtido. Execute novamente comando

```
sudo id
```

e verifique que já não precisou de introduzir a senha do utilizador corrente[3]. Explique a lógica à luz da definição de políticas de segurança.

Execute o comando

```
cat /etc/sudoers
```

e verifique que não consegue observar o conteúdo do ficheiro indicado. Use o comando `sudo` para ultrapassar a limitação encontrada. Observe e interprete o conteúdo do ficheiro.

2.7.2 Mecanismo de Set-UID

Certos comandos podem ser executados por qualquer utilizador mas realizam ações que só estão reservadas para administradores. Um exemplo de tais comandos é o `passwd`, que permite alterar a senha de um determinado utente. Este comando irá alterar o conteúdo de um ficheiro protegido onde estão guardadas informações sobre as senhas dos utilizadores reconhecidos pelo sistema, o qual não pode ser alterado livremente por qualquer utilizador.

Outro exemplo de comando que precisa de fazer ações privilegiadas mas sob invocação de utilizador qualquer é o comando `sudo`. Este comando permite mudar o UID de um processo para o UID 0 (`root`), o que é uma operação privilegiada.

A maneira como se lida com este dilema no Linux é com o mecanismo de Set-UID. Se a proteção associada a um comando possuir este bit ativo, então o UID do processo onde o comando será executado o UID do ficheiro com o programa do comando.

Observe a proteção do comando `passwd` usando o comando

```
ls -la /usr/bin/passwd
```

Faça o mesmo em relação ao comando `sudo`, usando

[3]Eventualmente tal poderá ter acontecido na primeira vez que executou este comando.

```
ls -la /usr/bin/sudo
```

Para estabelecer um termo de comparação, liste as proteções de um comando banal, como o `ls` que usou anteriormente, fazendo

```
ls -la /bin/ls
```

Compare as proteções indicadas e observe as diferenças.

Execute o comando

```
passwd
```

na consola corrente (1). Mude para a consola 2, faça *login*, e execute o comando

```
ps -au
```

Verifique qual é o USER (UID) dos processos associados ao TTY tty1 (consola 1), os quais deverão estar a executar os comandos `bash` e `passwd`.

Volte à consola 1 e termine o comando em curso, alterando ou não a senha do utilizador corrente.

2.8 Proteção de ficheiros

Os ficheiros têm uma proteção base que permite indicar se podem ser lidos, alterados ou executados (rwx). Este último só tem interesse para ficheiros que possuam comandos (ou aplicações).

A proteção de um ficheiro é indicada para 3 entidades distintas, não mais nem menos: o utilizador seu dono (identificado por um UID), o grupo seu dono (identificado por um GID) e os demais. O comando `ls -la` permite observar a proteção de um dado ficheiro.

Crie um ficheiro na diretoria corrente através do comando

```
cat > nome
```

onde `nome` é o nome do ficheiro que quer criar. Escreva o conteúdo do ficheiro (não edite!) e termine com Ctr-D (código Unix de fim de ficheiro). Vamos assumir que o ficheiro se chama `tralha`.

2.8.1 Pertença e proteções

Verifique a proteção do ficheiro `tralha` com o comando

```
ls -la
```

ou

```
ls -la tralha
```

Verifique quem é o utilizador seu dono, o grupo seu dono, e quais as proteções do ficheiro para estas duas entidades e para os demais.

2.8.2 Efeito da proteção de leitura

Liste o conteúdo do ficheiro com o comando

```
cat tralha
```

Remova o direito de leitura do utilizador dono do ficheiro `tralha` usando o comando

```
chmod u-r tralha
```

Liste novamente o conteúdo do ficheiro com o comando

```
cat tralha
```

e verifique a impossibilidade de o fazer. Explique porquê. Reponha o direito de leitura do utilizador dono do ficheiro `tralha` usando o comando

```
chmod u+r tralha
```

2.8.3 Efeito da proteção de escrita

Acrescente conteúdo ao ficheiro `tralha` com o comando

```
cat >> tralha
```

Escreva o conteúdo extra do ficheiro (não edite!) e termine com Ctr-D. Liste o conteúdo do ficheiro com o comando

```
cat tralha
```

Remova o direito de escrita do utilizador dono do ficheiro `tralha` usando o comando

```
chmod u-w tralha
```

Acrescente novamente conteúdo ao ficheiro `tralha` com o comando

```
cat >> tralha
```

e verifique a impossibilidade de o fazer. Explique porquê. Reponha o direito de escrita do utilizador dono do ficheiro `tralha` usando o comando

```
chmod u+w tralha
```

2.8.4 Efeito da proteção de execução

Acrescente conteúdo ao ficheiro `tralha` com o comando

```
cat >> tralha
```

Copie o comando `ls` para a diretoria corrente usando o comando

```
cp /bin/ls myls
```

A cópia passou a chamar-se `myls`. Execute o comando `myls` fazendo

```
./myls -la
```

Verifique o que resultado é igual ao que acontece com o comando `ls`. Remova o direito de execução do utilizador dono do ficheiro `myls` usando o comando

```
chmod u-x myls
```

Execute novamente o comando `myls` fazendo

```
./myls -la
```

Verifique a impossibilidade de o fazer. Explique porquê. Reponha o direito de execução do utilizador dono do ficheiro `myls` usando o comando

```
chmod u+x myls
```

e execute-o novamente.

2.9 Confinamento

Por vezes interessa confinar o âmbito da atividade de um programa. Confinar significa limitar de alguma forma mais agressiva a sua liberdade de atuação, os limites do que a aplicação pode fazer no sistema.

O comando `chroot` (*change root*) permite confinar a visão que uma aplicação possui do sistema de ficheiros. Este comando altera a noção que a aplicação (o seu processo) terá da diretoria raiz (*root*) do sistema de ficheiros. Assim, a aplicação só "verá" uma hierarquia de ficheiros que começará numa dada diretoria, que verá como raiz, e não toda a hierarquia que começa na raiz real do sistema de ficheiros.

Na experiência seguinte vai-se criar um ambiente de execução de comandos onde apenas existirão os comandos `bash`, `ls`, `mkdir`, e `vi`. Neste ambiente será possível listar os ficheiros existentes com o `ls`, criar diretorias com o `mkdir` e criar novos ficheiros, ou alterar ficheiros existentes, com o `vi`. Não será possível, por exemplo, apagar ficheiros ou alterar a sua localização.

Para isso, vamos criar uma pequena hierarquia de ficheiros com duas subdiretorias, bin e lib. A primeira terá os comandos acima indicados; a segunda terá as bibliotecas dinâmicas que os mesmos usam.

NOTA: na sequência de comandos seguidamente indicada o caráter ' é uma pelica no teclado Português, e não uma acento agudo, enquanto o caráter ` é uma acento grave. Eles têm significados diferentes: o primeiro serve para delimitar uma sequência de caracteres que formam apenas um único elemento; o segundo representa o resultado do comando que delimitam (i.e., ``cmd` é o texto formado pelo que é escrito pelo comando `cmd` durante a sua execução). O comando indentado é para ser introduzido no seguimento do anterior.

```
cd /tmp
mkdir newroot
cd newroot
mkdir bin
mkdir lib
cp /bin/bash /bin/ls /bin/mkdir /usr/bin/vi bin
ldd bin/* | awk '{for (i = 1; i <= NF; i++) print $i}' |
    grep ^/ | sort | uniq | sed -e s#^/## > files
for i in `cat files`; do mkdir -p `dirname $i`; cp /$i $i; done
rm files
```

Criada a hierarquia reduzida abaixo da diretoria `newroot`, podemos listar o seu conteúdo com

```
ls -lR
```

Agora vamos executar o comando `bash` que está na diretoria `bin` indicando-lhe como diretoria raiz a diretoria atual (`/tmp/newroot`):

```
sudo chroot . /bin/bash
```

Após a execução deste comando, verifica-se que o interpretador de comandos está a trabalhar num ambiente confinado, com um sistema de ficheiros reduzido:

```
ls -l /
```

Pode verificar que apenas consegue executar os comandos anteriormente copiados e mais nenhuns.

2.10 Política de verificação de senhas de *login*

Por omissão, a distribuição de Linux que se está a usar não verifica se as senhas escolhidas pelos utentes são ou não fracas. Uma senha fraca é uma senha que pode ser descoberta com um ataque de pesquisa "inteligente", onde são feitas tentativas com um conjunto de senhas prováveis (dicionário de senhas). Uma senha fraca compromete a segurança do sistema porque permite que o seu dono seja facilmente personificado por terceiros.

Para alterar esta política pode-se alterar a forma como são controlados os procedimentos de alteração de uma senha. Esses procedimentos são controlados por uma arquitetura modular, designada por PAM (*Pluggable Authentication Modules*), a qual permite efetuar diversas operações complementares ou alternativas relacionadas com os processos de autenticação de pessoas. Vamos então alterar o procedimento de alteração de uma senha de modo a incluir a validação da robustez da nova senha proposta.

Para facilitar o processo que se segue passe a executar comandos como `root`, através do comando

```
sudo bash
```

2.10.1 Alteração das políticas de segurança via PAM: deteção de senhas fracas

Instale o módulo PAM `pam_cracklib.so`, o qual não vem instalado por omissão. Para isso é preciso primeiro configurar o instalador de *packages* para ir procurar o módulo nos locais corretos. Edite o ficheiro `/etc/apt/sources.list` e retire o comentário (retire o cardinal) do início de todas as linhas que começam por `#deb␣`.

Feito isto, ligue a máquina Linux à rede (se ainda não estiver ligada), e execute os comandos

```
apt-get update
apt-get install libpam-cracklib
```

O ficheiro `/etc/pam.d/common-password` contém os procedimentos comuns aos processos de alteração de senhas. Edite esse ficheiro e procure uma linha com o seguinte conteúdo:

```
password requisite pam_cracklib.so retry=3 minlen=8 difok=3
```

Altere o valor do parâmetro `minlen` para 10 e de `difok` para 5. Desta forma, uma senha não poderá ter um comprimento inferior a 10 caracteres e será validada através de um dicionário se é ou não robusta. Para além disso, uma nova senha deverá possuir pelo menos 5 caracteres diferentes da anterior.

Numa consola diferente faça *login* como um utente comum, e não como `root`. Mantenha a consola anterior para o `root` para poder corrigir algo de drástico que aconteça!

Altere a senha do utilizador corrente com o comando

```
passwd
```

Experimente dar uma senha nova com menos de 10 caracteres. Experimente depois com uma senha maior mas que seja um conjunto de uma ou mais palavras inglesas (“elephant”, “mypassword”, “googlerules”, “exercises”, etc.). Verifique que por vezes o módulo de verificação PAM não permite a alteração da senha por considerar que a nova é demasiado fácil de adivinhar (por ser demasiado curta ou demasiado fácil de adivinhar com um dicionário de senhas).

Experimente também alterar a senha para uma quase igual, diferindo até um máximo de 5 letras. A diferença autorizada entre as senhas é controlada pelo parâmetro `difok`, o qual possui o valor 5.

Na consola onde está como `root` edite novamente ficheiro `/etc/pam/common-password` e procure a linha

```
password [success=1 default=ignore] pam_unix.so obscure use_authtok
try_first_pass sha512
```

e acrescente à mesma o parâmetro `remember=2`.

Mude novamente de consola, para a do utente normal, e experimente alterar a senha alternando entre duas aceitáveis. Verifique que a alternância é negada, e que só se pode repetir uma senha após ter alterado com sucesso para três outras senhas. O que acontece é fruto da parametrização extra que se adicionou: o sistema passou a memorizar as duas últimas senhas para além da corrente, e impede que o utente reutilize com demasiada frequência a mesma senha.

2.11 Bibliografia

- Identificação de um utilizador, `http://en.wikipedia.org/wiki/User_ID`
- Identificação de um grupo, `http://en.wikipedia.org/wiki/Group_identifier`
- Super-utilizador, `http://en.wikipedia.org/wiki/Superuser`
- Permissões em sistemas de ficheiros, `http://en.wikipedia.org/wiki/File_system_permissions`
- Mecanismo Set-UID, `http://en.wikipedia.org/wiki/Setuid`
- Comando `sudo`, `http://en.wikipedia.org/wiki/Sudo`
- Mecanismo `chroot`, `http://en.wikipedia.org/wiki/Chroot`
- PAM (*Pluggable Authentication Modules*), `http://en.wikipedia.org/wiki/Pluggable_Authentication_Modules`