Laboratório Aberto - Configuração e proteção do SSH

Neste laboratório, você define a autenticação baseada em senha para os usuários, e desativa o login direto como `root` e a autenticação de senha para todos os usuários para o serviço OpenSSH em um dos seus servidores.

Resultados

- Autenticar com chaves SSH.
- Impedir que os usuários façam login diretamente como o usuário `root` no serviço `ssh`.
- Impedir que os usuários façam login no sistema usando a autenticação baseada em senha do SSH.

Com o usuário `student` na máquina `workstation`, use o comando `lab` para preparar seu sistema para este exercício.

Esse comando prepara seu ambiente e garante que todos os recursos necessários estejam disponíveis.

```
[student@workstation ~]$lab start ssh-review
```

Instruções

1. Na máquina `workstation`, faça login na máquina `servera` como o usuário `student`.

   `[student@workstation ~]$ ssh student@servera`
   `[student@servera ~]$`

   Ocultar solução

2. Alterne para o usuário `production1` na máquina `servera`. Digite `redhat` com a senha.

   `[student@servera ~]$ su - production1`
   `Password:`
   `redhat`
   `[production1@servera ~]$`

Ocultar solução

3. Gere chaves SSH sem senha para o usuário `production1` na máquina `servera`.

```
[production1@servera ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/production1/.ssh/id_rsa):
Enter
Created directory '/home/production1/.ssh'.
Enter passphrase (empty for no passphrase):
Enter
Enter same passphrase again:
Enter
Your identification has been saved in /home/production1/.ssh/id_rsa.
Your public key has been saved in /home/production1/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:MCQ8nXClDFSlJV0i5IouUzlzFrbsdz+jO8ZIMeSTOuQ production1@servera.lab.example.com
The key's randomart image is:
+---[RSA 3072]----+
|   o==B==..      |
|    oB+*..       |
|    o+B.         |
|   =.+*o         |
|  *o*. +S        |
| o *E .          |
|o . .o.o.        |
| o   ...+.o      |
|       .o+.o     |
+----[SHA256]-----+
```

Ocultar solução

4. Envie a chave pública do par de chaves SSH para o usuário `production1` na máquina `serverb`.

```
[production1@servera ~]$ ssh-copy-id production1@serverb
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/production1/.ssh/id_rsa.pub"
The authenticity of host 'serverb (172.25.250.11)' can't be established.
ED25519 key fingerprint is SHA256:h/hEJa/anxp6AP7BmB5azIPVbPNqieh0oKi4KWOTK80.
Are you sure you want to continue connecting (yes/no)?
yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
production1@serverb's password:
redhat
Number of key(s) added: 1

Now try logging in to the machine, with:   "ssh 'production1@serverb'"
and check to make sure that only the key(s) you wanted were added.
```

Ocultar solução

5. Confirme que o usuário `production1` consegue fazer login com êxito na máquina `serverb` como com as chaves SSH.

```
[production1@servera ~]$ ssh production1@serverb ...output omitted...
[production1@serverb ~]$
```

Ocultar solução

6. Configure o serviço `sshd` no `serverb` para impedir que os usuários façam login como o usuário `root`. Use `redhat` como a senha de `root`.

1. Alterne para o usuário `root` na máquina `serverb`.

```
[production1@serverb ~]$ su -
Password:
redhat
[root@serverb ~]#
```

2. Defina o parâmetro `PermitRootLogin` como `no` no arquivo `/etc/ssh/sshd_config` e recarregue o serviço `sshd`. Edite o parâmetro ativo não comentado e não um exemplo comentado.

```
...output omitted...
PermitRootLogin
no ...output omitted...
[root@serverb ~]#
systemctl reload sshd.service
```

3. Abra outro terminal na máquina `workstation` e faça login na máquina `servera` como o usuário `production1`. Em `servera`, tente fazer login na máquina `serverb` com o usuário `root`. Esse comando deve falhar, porque você desativou o login de usuário `root` de SSH.
O comando saiu após três tentativas malsucedidas de fazer login na máquina
`servera` como o usuário `root`. Por padrão, o comando `ssh` prefere usar chaves SSH para autenticação e, se ele não encontrar as chaves necessárias do usuário, solicita a senha do usuário para autenticação.
Nota

Para a conveniência do curso, o login sem senha
`root` já está configurado entre `workstation` e `servera` no ambiente de sala de

aula. No entanto, essa configuração é altamente insegura e não é recomendada para qualquer ambiente de produção.

```
[student@workstation ~]$ ssh production1@servera ...output omitted...
[production1@servera ~]$
ssh root@serverb
root@serverb's password:
redhat
Permission denied, please try again.
root@serverb's password:
redhat
Permission denied, please try again.
root@serverb's password:
redhat
root@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
[production1@servera ~]$
```

Ocultar solução

7. Configure o serviço `sshd` na máquina `serverb` para permitir que os usuários autentiquem com somente chaves SSH, em vez de com suas senhas.

   1. Volte para o primeiro terminal com o shell ativo `root` na máquina `serverb`. Defina o parâmetro `PasswordAuthentication` como `no` no arquivo `/etc/ssh/sshd_config` e recarregue o serviço `sshd`. Edite o parâmetro ativo não comentado e não um exemplo comentado.

```
...output omitted...
PasswordAuthentication no

...output omitted...
[root@serverb ~]#
systemctl reload sshd
```

   2. Acesse o segundo terminal com o shell ativo `production1` na máquina `servera` e tente fazer login na máquina `serverb` com o usuário `production2`. Esse comando deve falhar, porque as chaves SSH não estão configuradas para o usuário `production2`, e o serviço `sshd` na máquina `serverb` não permite o uso de senhas para autenticação.

```
[production1@servera ~]$ ssh production2@serverb
production2@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

Nota

Para mais granularidade, você pode usar as opções explícitas `-o PubkeyAuthentication=no` e `-o PasswordAuthentication=yes` do comando `ssh`. Com essas opções, você pode substituir os padrões do comando `ssh` e determinar com segurança que o comando anterior falhe com base nas configurações ajustadas no arquivo `/etc/ssh/sshd_config` na etapa anterior.

3. Volte para o primeiro terminal com o shell ativo `root` na máquina `serverb`. Verifique se `PubkeyAuthentication` está ativado no arquivo `/etc/ssh/sshd_config`.

```
[root@serverb ~]$ cat /etc/ssh/sshd_config ...output omitted...
#PubkeyAuthentication yes

...output omitted...
```

A linha `PubkeyAuthentication` está comentada. As linhas comentadas indicam os valores padrão de um parâmetro. A autenticação de chave pública do SSH está ativa por padrão, como a linha comentada indica.

4. Retorne ao segundo terminal com o shell ativo `production1` na máquina `servera` e tente fazer login na máquina `serverb` com o usuário `production1`. Esse comando deve ser bem-sucedido porque as chaves SSH estão configuradas para o usuário `production1` fazer login na máquina `serverb` a partir da máquina `servera`.

```
[production1@servera ~]$ ssh production1@serverb ...output omitted...
[production1@serverb ~]$
```

5. Saia e feche o terminal adicional.

```
[production1@serverb ~]$ exit
logout
Connection to serverb closed.
[production1@servera ~]$
exit
logout
[student@workstation ~]$
exit
```

6. Retorne ao sistema `workstation` como o usuário `student`.

```
[production1@serverb ~]$ exit
logout
Connection to serverb closed.
[production1@servera ~]$ exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Ocultar solução

Avaliação

Com o usuário `student` na máquina `workstation`, use o comando `lab` para avaliar seu trabalho. Corrija todas as falhas relatadas e execute novamente o comando até que ele seja concluído com êxito.

```
[student@workstation ~]$lab grade ssh-review
```

Encerramento

Na máquina `workstation`, altere para o diretório pessoal do usuário `student` e use o comando `lab` para concluir este exercício. Essa etapa é importante para garantir que recursos de exercícios anteriores não afetem exercícios futuros.

```
[student@workstation ~]$lab finish ssh-review
```

Isso conclui a seção.