Capitulo 10. Configuração e proteção do SSH

Acesso à linha de comando remoto com o SSH

Objetivos

Fazer login em um sistema remoto e executar comandos com `ssh`.

Descrição do Secure Shell

Com o protocolo SSH, os sistemas podem se comunicar de maneira criptografada e segura em uma rede insegura.

Use o comando `ssh` para criar uma conexão segura com um sistema remoto, autenticar como um usuário específico e obter uma sessão de shell interativa no sistema remoto.

O comando `ssh` pode executar uma sessão em um sistema remoto sem executar um shell interativo.

Exemplos do Secure Shell

O comando `ssh` a seguir faz seu login no servidor remoto `hosta` usando o mesmo nome de usuário do usuário local atual.

Neste exemplo, o sistema remoto solicita a autenticação com a senha do usuário `developer1`.

```
[developer1@host ~]$ssh hosta
developer1@hosta's password:redhat...output omitted...
[developer1@hosta ~]$
```

Use o comando `exit` para fazer o logout do sistema remoto.

```
[developer1@hosta ~]$exit
logout
Connection to hosta closed.
[developer1@host ~]$
```

O comando `ssh` a seguir faz o seu login no servidor remoto `hosta` com o nome de usuário `developer2`. O sistema remoto solicita a autenticação com a senha do usuário `developer2`.

```
[developer1@host ~]$ssh developer2@hosta
developer2@hosta's password:shadowman...output omitted...
[developer2@hosta ~]$
```

O comando `ssh` a seguir executa o comando `hostname` no sistema remoto `hosta` como o usuário `developer2` sem acessar o shell interativo remoto.

```
[developer1@host ~]$ssh developer2@hosta hostname
developer2@hosta's password:shadowman
hosta.lab.example.com
[developer1@host ~]$
```

Esse comando exibe a saída no terminal do sistema local.

Identificação de usuários remotos

O comando `w` exibe uma lista de usuários atualmente conectados ao sistema. Ele também exibe o local do sistema remoto e os comandos que o usuário executou.

```
[developer1@host ~]$ssh developer1@hosta
developer1@hosta's password:redhat
[developer1@hosta ~]$w
 16:13:38 up 36 min,  1 user,  load average: 0.00, 0.00, 0.
00
```

```
USER     TTY      FROM             LOGIN@   IDLE   JCPU   P
CPU WHAT
developer2   pts/0    172.25.250.10    16:13    7:30   0.01
s  0.01s -bash
developer1   pts/1    172.25.250.10    16:24    3.00s  0.01
s  0.00s w
[developer2@hosta ~]$
```

A saída mostra que o usuário `developer2` fez login no sistema no pseudoterminal `0` às `16:13` hoje a partir do host com o endereço IP `172.25.250.10` e ficou ocioso em um prompt do shell por sete minutos e trinta segundos.

A saída também mostra que o usuário `developer1` fez login no sistema no pseudoterminal `1` e ficou ocioso desde os últimos três segundos após a execução do comando `w`.

Chaves do host do SSH

O SSH garante a segurança da comunicação por meio da criptografia de chaves públicas.

Quando um cliente SSH se conecta a um servidor SSH, o servidor envia uma cópia da chave pública ao cliente antes de o cliente fazer login.

Essa chave ajuda a configurar a criptografia segura do canal de comunicação e a autenticar o sistema do cliente.

Quando um usuário executa o comando `ssh` para se conectar a um servidor SSH, o comando verifica se há uma cópia da chave pública desse servidor em seu arquivo host local conhecido.

A chave pode estar pré-configurada no arquivo `/etc/ssh/ssh_known_hosts` ou o usuário pode ter o arquivo `~/.ssh/known_hosts` que contém a chave em seu diretório pessoal.

Se o cliente tiver uma cópia da chave, o comando `ssh` comparará a chave dos arquivos de servidor host conhecidos com a chave recebida. Se as chaves não corresponderem, o cliente entenderá que o tráfego de rede ao servidor está comprometido e solicitará que o usuário confirme se deve ou não continuar com a conexão.

Verificação estrita de chave de host

O parâmetro `StrictHostKeyChecking` é definido no arquivo `~/.ssh/config` específico do usuário, no arquivo de todo o sistema `/etc/ssh/ssh_config` ou pela especificação da opção `-o StrictHostKeyChecking=` do comando `ssh`.

- **Se o parametro `StrictHostKeyChecking` estiver definido como `yes`, o comando ssh sempre anula a conexão por SSH se as chaves publicas não corresponderem.**
- **Se o parametro `StrictHostKeyChecking` estiver difinindo como `no`, o comando SSH habilitara a conexão e adicionara a chave do host de destino ao arquivo `~/.ssh/known_hosts`.**

Se a chave SSH do host de destino tiver mudado desde a última vez que você se conectou a ela, o comando `ssh` solicitará confirmação para fazer login e aceitar a nova chave.

Se você aceitar a nova chave, uma cópia da chave pública será salva no arquivo `~/.ssh/known_hosts` para confirmar automaticamente a identidade do servidor nas conexões subsequentes.

Nota

A Red Hat recomenda definir o parâmetro `StrictHostKeyChecking` como `yes` no arquivo `~/.ssh/config` específico do usuário ou no arquivo `/etc/ssh/ssh_config` válido para todo o sistema para que o comando `ssh` sempre anule a conexão por SSH se as chaves públicas não corresponderem.

```
[developer1@host ~]$ssh hostb
The authenticity of host 'hostb (172.25.250.12)' can't be e
stablished.
ECDSA key fingerprint is SHA256:qaS0PToLrqlCO2XGklA0iY7CaP7
aPKimerDoaUkv720.
Are you sure you want to continue connecting (yes/no)?no
[developer1@host ~]$
```

Verifique a impressão digital da chave de host SSH do servidor de destino usando o comando `ssh-keygen`. Neste exemplo, o comando `ssh-keygen` é executado no servidor de destino `hostb`.

O comando `ssh-keygen` exibe a impressão digital da chave para que você possa correspondê-la à saída do comando `ssh` e verificar se a chave é válida. Use as opções `-lf` para listar a impressão digital de chave pública no arquivo de chave pública padrão do host.

Como não é possível se conectar por SSH, você deve verificar a impressão digital de chave do host de destino fazendo login localmente. Use um método de comunicação fora de banda para compartilhar chaves públicas, como uma chamada telefônica ou uma videoconferência.

```
[developer1@hostb ~]$ssh-keygen -lf /etc/ssh/ssh_host_ecdsa
_key.pub
256 SHA256:qaS0PToLrqlCO2XGklA0iY7CaP7aPKimerDoaUkv720 root
@server (ECDSA)
```

Depois de verificar a chave no host de destino, você pode aceitar a chave e se conectar ao host de destino.

```
[developer1@host ~]$ssh hostb
The authenticity of host 'hostb (172.25.250.12)' can't be e
stablished.
ECDSA key fingerprint is SHA256:qaS0PToLrqlCO2XGklA0iY7CaP7
aPKimerDoaUkv720.
Are you sure you want to continue connecting (yes/no)?yes
Warning: Permanently added 'hostb,172.25.250.12' (ECDSA) to
the list of known hosts.
developer1@hostb's password:redhat...output omitted...
[developer1@hostb ~]$
```

Gerenciamento de chaves de hosts conhecidos SSH

As informações sobre sistemas remotos conhecidos e suas chaves são armazenadas em um dos seguintes locais:

- O arquivo de todo o sistema /etc/ssh/ssh_known_hosts
- O arquivo ~/.ssh/known_hosts no diretorio pessoal de cada usuario

O arquivo `/etc/ssh/ssh_known_hosts` é um arquivo de todo o sistema que armazena as chaves públicas para hosts conhecidos pelo sistema. Você deve criar e gerenciar esse arquivo, manualmente ou por algum método automatizado, como o Ansible ou um script que usa o utilitário `ssh-keyscan`.

A chave pública de um servidor pode ter sido alterada porque a chave foi perdida devido a uma falha no disco rígido ou porque ela foi substituída por algum motivo legítimo. Nesse caso, para fazer login com êxito nesse sistema, o arquivo `/etc/ssh/ssh_known_hosts` deve ser modificado para substituir a entrada de chave pública anterior pela nova chave pública.

Se você se conectar a um sistema remoto e a chave pública desse sistema não estiver no arquivo `/etc/ssh/ssh_known_hosts`, o cliente SSH pesquisará a chave no arquivo `~/.ssh/known_hosts`.

Cada entrada de chave de host conhecida consiste em uma linha contendo três campos:

- O primeiro campo é lista de nomes de host e endereço IP que compartilham a chave publica
- O segundo campo é algoritmo de criptografia da chave
- O ultimo campo é a chave em si

Solução de problemas com a chave do host

Se o endereço IP ou a chave pública do sistema remoto forem alterados e você tentar se conectar a esse sistema novamente por SSH, o cliente SSH detectará que a entrada de chave desse sistema no arquivo `~/.ssh/known_hosts` não é mais válida. Uma mensagem de aviso informa que a identificação do host remoto foi alterada e que você deve modificar a entrada de chave.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the
-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host i
s
SHA256:hxttxb/qVi1/ycUU2wXF6mfGH++Ya7WYZv0r+tIkg4I.
Please contact your system administrator.
Add correct host key in /home/user/.ssh/known_hosts to get
rid of this message.
Offending ECDSA key in /home/user/.ssh/known_hosts:12
ECDSA host key for server1.example.com has changed and you
have requested strict checking.
Host key verification failed.
```

Se você não souber por que a chave foi alterada, verifique a nova impressão digital da chave, pois essa chave pode ser um ataque real à sua rede. Use um método fora de banda para verificação, como falar com o administrador do sistema de destino.

Se você souber por que a chave foi alterada, como uma alteração de endereço IP, resolva esse problema removendo a entrada de chave relevante do arquivo `~/.ssh/known_hosts` e reconecte-se ao sistema para receber a nova entrada de chave.

O número da linha da entrada de chave relevante é especificado na mensagem de aviso. Você também pode localizar e remover a entrada de chave relevante executando o seguinte comando:

```
[developer1@host ~]$ssh-keygen -Rremotesystemname -f ~/.ss
h/known_hosts
# Host remotesystemname found: line 12
/home/user/.ssh/known_hosts updated.
Original contents retained as /home/user/.ssh/known_hosts.o
ld
```