Configuração de autenticação baseada em chave SSH

Autenticação baseada em chaves SSH

Você pode configurar sua conta para acesso sem senha a servidores SSH que habilitaram a autenticação baseada em chave, que é baseada em criptografia de chave pública (PKI).

Para preparar sua conta, gere um par de arquivos de chave relacionados criptograficamente. Uma chave é privada e mantida apenas por você.

A segunda chave é sua chave pública relacionada, que não é secreta.

A chave privada atua como sua credencial de autenticação e deve ser armazenada com segurança. A chave pública é copiada para sua conta em servidores que você acessará remotamente e verifica o uso de sua chave privada.

Quando você faz login na sua conta em um servidor remoto, o servidor remoto usa sua chave pública para criptografar uma mensagem de desafio e enviá-la ao seu cliente SSH.

Seu cliente SSH deve provar que pode descriptografar a mensagem, o que demonstra que você tem a chave privada associada. Se a verificação tiver êxito, isso significará que sua solicitação é confiável, e você receberá acesso sem fornecer uma senha.

As senhas podem ser facilmente aprendidas ou roubadas, mas as chaves privadas armazenadas com segurança são mais difíceis de comprometer.

Geração de chaves SSH

Use o comando `ssh-keygen` para criar um par de chaves. Por padrão, o comando `ssh-keygen` salva suas chaves privadas e públicas nos arquivos `~/.ssh/id_rsa` e `~/.ssh/id_rsa.pub`, mas você pode especificar um nome diferente.

```
[user@host ~]$ssh-keygen
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/user/.ssh/id_rs
a):Enter
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):Enter
Enter same passphrase again:Enter
Your identification has been saved in /home/user/.ssh/id_rs
a.
Your public key has been saved in /home/user/.ssh/id_rsa.pu
b.
The key fingerprint is:
SHA256:vxutUNPio3QDCyvkYm1 user@host.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|   .     .       |
|  o o     o      |
| . = o   o .     |
|  o + = S E .    |
| ..O o + * +     |
|.+% O . + B .    |
|=*oO . . + *     |
|++.     . +.     |
+----[SHA256]-----+
```

Você pode optar por fornecer uma senha para `ssh-keygen`, que é usada para criptografar sua chave privada. O uso de uma senha é recomendado para que sua chave privada não possa ser usada por alguém para obter acesso.

Se você definir uma senha, deverá inseri-la toda vez que usar a chave privada. A senha é usada localmente para descriptografar sua chave privada antes do uso, ao contrário de sua senha, que deve ser enviada em texto não criptografado pela rede para uso.

Você pode usar o gerenciador de chaves `ssh-agent` localmente, que armazena sua senha em cache no primeiro uso em uma sessão de login e, em seguida, fornece a senha para todos os usos de chave privada subsequentes na mesma sessão de login. O comando `ssh-agent` é discutido mais tarde nesta seção.

No exemplo a seguir, uma chave privada protegida por senha é criada com a chave pública.

```
[user@host ~]$ssh-keygen -f .ssh/key-with-pass
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):your_passphrase
Enter same passphrase again:your_passphrase
Your identification has been saved in .ssh/key-with-pass.
Your public key has been saved in .ssh/key-with-pass.pub.
The key fingerprint is:
SHA256:w3GGB7EyHUry4aOcNPKmhNKS7dl1YsMVLvFZJ77VxAo user@hos
t.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|    . + =.o ...  |
|     = B XEo o.  |
|  . o O X =....  |
| = = = B = o.    |
|= + * * S .      |
|.+ = o + .       |
|  + .            |
|                 |
|                 |
+----[SHA256]-----+
```

A opção `-f` do comando `ssh-keygen` especifica os arquivos nos quais salvar as chaves. No exemplo anterior, o comando `ssh-keygen` salvou o par de chaves nos arquivos `/home/user/.ssh/key-with-pass` e `/home/user/.ssh/key-with-pass.pub`.

As chaves SSH geradas são armazenadas por padrão no subdiretório `.ssh` de seu diretório pessoal. Para funcionar corretamente, a chave privada deve ser legível e gravável somente pelo usuário ao qual ela pertence (permissões octais 600). A chave pública não é segura e também pode ser lida por qualquer pessoa no sistema (permissões octais 644).

Compartilhamento da chave pública

Para configurar sua conta remota para acesso, copie sua chave pública para o sistema remoto. O comando `ssh-copy-id` copia a chave pública do par de chaves SSH para o sistema remoto.

Para configurar sua conta remota para acesso, copie sua chave pública para o sistema remoto. O comando `ssh-copy-id` copia a chave pública do par de chaves SSH para o sistema remoto.

```
[user@host ~]$ssh-copy-id -i .ssh/key-with-pass.pub user@re
motehost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installe
d: "/home/user/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the n
ew 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
user@remotehost's password:redhat
Number of key(s) added: 1

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

Depois de colocar a chave pública, teste o acesso remoto com a chave privada correspondente.

Se a configuração estiver correta, você poderá acessar sua conta no sistema remoto sem que a senha da conta seja solicitada. Se você não especificar um arquivo de chave privada, o comando `ssh` usará o arquivo padrão `~/.ssh/id_rsa` se ele existir.

Importante

Se você configurou uma senha para proteger sua chave privada, o SSH solicitará a senha no primeiro uso. No entanto, se a autenticação da chave tiver êxito, a senha da conta não será solicitada.

```
[user@host ~]$ssh -i .ssh/key-with-pass user@remotehost
Enter passphrase for key '.ssh/key-with-pass':your_passphra
se...output omitted...
[user@remotehost ~]$
```

Autenticação não interativa com o gerenciador de chaves

Se você criptografar sua chave privada com uma senha, deverá inseri-la toda vez que usar a chave privada para autenticação. No entanto, você pode configurar o gerenciador de chaves `ssh-agent` para armazenar em cache as senhas. Então, cada vez que você usar o SSH, o gerenciador de chaves `ssh-agent` fornecerá a senha para você. Usar um gerenciador de chaves é conveniente e pode melhorar a segurança, fornecendo menos oportunidades para outras pessoas observarem sua senha.

O gerenciador de chaves `ssh-agent` pode ser configurado para iniciar automaticamente quando você fizer login. O ambiente gráfico de área de trabalho GNOME pode iniciar e configurar automaticamente o gerenciador de chaves `ssh-agent`. Se você fizer login em um ambiente de texto, deverá iniciar o programa `ssh-agent` manualmente para cada sessão.

Inicie o programa `ssh-agent` com o seguinte comando:

```
[user@host ~]$eval $(ssh-agent)
Agent pid 10155
```

Quando você inicia manualmente o comando `ssh-agent`, ele executa comandos de shell adicionais para definir variáveis de ambiente que são necessárias para uso com o comando `ssh-add`.

Você pode carregar manualmente sua senha de chave privada para o gerenciador de chaves usando o comando `ssh-add`.

Os seguintes comandos de exemplo `ssh-add` adicionam as chaves privadas do arquivo `~/.ssh/id_rsa` padrão e, em seguida, de um arquivo `~/.ssh/key-with-pass`:

```
[user@host ~]$ssh-add
Identity added: /home/user/.ssh/id_rsa (user@host.lab.examp
le.com)
[user@host ~]$ssh-add .ssh/key-with-pass
Enter passphrase for .ssh/key-with-pass:your_passphrase
Identity added: .ssh/key-with-pass (user@host.lab.example.c
om)
```

O seguinte comando `ssh` usa o arquivo de chave privada padrão para acessar sua conta em um servidor SSH remoto:

```
[user@host ~]$ssh user@remotehost
Last login: Mon Mar 14 06:51:36 2022 from host.example.com
[user@remotehost ~]$
```

O comando `ssh` a seguir usa a chave privada `~/.ssh/key-with-pass` para acessar sua conta no servidor remoto. A chave privada neste exemplo foi descriptografada anteriormente e adicionada ao gerenciador de chaves `ssh-agent`; portanto, o comando `ssh` não solicita a senha para descriptografar a chave privada.

```
[user@host ~]$ssh -i .ssh/key-with-pass user@remotehost
Last login: Mon Mar 14 06:58:43 2022 from host.example.com
[user@remotehost ~]$
```

Quando você faz o logout de uma sessão que usou um gerenciador de chaves `ssh-agent`, todas as senhas armazenadas em cache são apagadas da memória.

Solução de problemas de conexão por SSH básica

O SSH pode parecer complexo quando o acesso remoto usando a autenticação de par de chaves não tiver êxito. O comando `ssh` fornece três níveis de detalhamento com as opções `-v`, `-vv` e `-vvv`, que respectivamente fornecem cada vez mais informações de depuração durante o uso do comando `ssh`.

O próximo exemplo demonstra as informações fornecidas ao usar a opção de menor detalhamento:

```
[user@host ~]$ssh -v user@remotehost
OpenSSH_8.7p1, OpenSSL 3.0.1 14 Dec 2021 (1)
debug1: Reading configuration data /etc/ssh/ssh_config (2)
debug1: Reading configuration data /etc/ssh/ssh_config.d/01
-training.conf
```

```
debug1: /etc/ssh/ssh_config.d/01-training.conf line 1: Appl
ying options for *
debug1: Reading configuration data /etc/ssh/ssh_config.d/50
-redhat.conf
...output omitted...
debug1: Connecting to remotehost [192.168.1.10] port 22.
(3)
debug1: Connection established.
...output omitted...
debug1: Authenticating to remotehost:22 as 'user' (4)
...output omitted...
debug1: Authentications that can continue: publickey,gssapi
-keyex,gssapi-with-mic,password  (5)
...output omitted...
debug1: Next authentication method: publickey (6)
debug1: Offering public key: /home/user/.ssh/id_rsa RSA SHA
256:hDVJjD7xrUjXGZVRJQixxFV6NF/ssMjS6AuQ1+VqUc4  (7)
debug1: Server accepts key: /home/user/.ssh/id_rsa RSA SHA2
56:hDVJjD7xrUjXGZVRJQixxFV6NF/ssMjS6AuQ1+VqUc4 (8)
Authenticated to remotehost ([192.168.1.10]:22) using "publ
ickey".
...output omitted...
[user@remotehost ~]$
```

1- Versões OpenSSH e OpenSSL

2- Arquivos de configuração OpenSSH

3- Conexão com o host remoto.

4- Tentando autenticar o usuário no host remoto.

5- Metodos de autenticação que o host remoto permite

6- Tentando autenticar o usuario usando a chave SSH

7- Usando o arquivo de chave /home/user/.ssh/id_rsa para autenticar

8- Os hosts remotos aceita, a chave SSH

Se uma tentativa de método de autenticação falhar, um servidor SSH remoto fará failback para outros métodos de autenticação permitidos até que todos os métodos disponíveis sejam tentados. O próximo exemplo demonstra um acesso

remoto com uma chave SSH que falha, mas o servidor SSH oferece autenticação de senha com êxito.

```
[user@host ~]$ssh -v user@remotehost...output omitted...deb
ug1: Next authentication method: publickey
debug1: Offering public key: /home/user/.ssh/id_rsa RSA SHA
256:bsB6l5R184zvxNlrcRMmYd32oBkU1LgQj09dUBZ+Z/k
debug1: Authentications that can continue: publickey,gssapi
-keyex,gssapi-with-mic,password
...output omitted...debug1: Next authentication method: pas
sword
user@remotehost's password:password
Authenticated to remotehost ([172.25.250.10]:22) using "pas
sword".
...output omitted...
[user@remotehost ~]$
```

Configuração de cliente SSH

Você pode criar o arquivo `~/.ssh/config` para pré-configurar conexões por SSH.

No arquivo de configuração, você pode especificar parâmetros de conexão, como usuários, chaves e portas para hosts específicos.

Esse arquivo elimina a necessidade de especificar manualmente os parâmetros de comando toda vez que você se conectar a um host. Considere o seguinte arquivo `~/.ssh/config`, que pré-configura duas conexões de host com diferentes usuários e chaves:

```
[user@host ~]$cat ~/.ssh/config
host servera
     HostName                    servera.example.com
     User                        usera
     IdentityFile                ~/.ssh/id_rsa_servera

host serverb
     HostName                    serverb.example.com
```

```
    User                        userb
    IdentityFile                ~/.ssh/id_rsa_serverb
```