以下 **ssh-add** 命令添加分别来自 **/home/user/.ssh/id_rsa**（默认）和 **/home/user/.ssh/key-with-pass** 文件的私钥。

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

将私钥成功添加到 ssh-agent 进程后，您可以使用 **ssh** 命令来调用 SSH 连接。如果您使用的是默认 **/home/user/.ssh/id_rsa** 文件之外的私钥文件，您必须将 **-i** 选项用于 **ssh**命令来指定私钥文件的路径。

以下示例中的 **ssh** 命令使用默认私钥文件对 SSH 服务器进行身份验证。

```
[user@host ~]$ ssh user@remotehost
Last login: Fri Apr  5 10:53:50 2019 from host.example.com
[user@remotehost ~]$
```

以下示例中的 **ssh** 命令使用 **/home/user/.ssh/key-with-pass**（非默认）私钥文件对 SSH 服务器进行身份验证。以下示例中的私钥已被解密并添加到其父级 **ssh-agent** 进程，所以 **ssh** 命令提示您通过交互式输入其密语来解密私钥。

```
[user@host ~]$ ssh -i .ssh/key-with-pass user@remotehost
Last login: Mon Apr  8 09:44:20 2019 from host.example.com
[user@remotehost ~]$
```

当您注销启动了 **ssh-agent** 的会话时，将退出进程，并且您的私钥密语也将从内存中清除。

参考文献

ssh-keygen(1)、ssh-copy-id(1)、ssh-agent(1)、ssh-add(1) man page

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

指导练习

配置基于 SSH 密钥的身份验证

在本练习中，您要对用户进行配置，以便其将基于密钥的身份验证用于 SSH。

成果

您应能够：

- 生成无密语保护的 SSH 密钥对。
- 生成受密语保护的 SSH 密钥对。
- 使用无密钥保护和受密语保护的 SSH 密钥进行身份验证。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 **`lab ssh-configure start`** 脚本来开始本练习。此脚本会创建必要的用户帐户。

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

1. 从 `workstation`，以 `student` 用户身份打开连接 `serverb` 的 SSH 会话。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

2. 使用 **su** 命令，切换为 `serverb`上的 `operator1` 用户。使用 `redhat` 作为 `operator1` 的密码。

```
[student@serverb ~]$ su - operator1
Password: redhat
[operator1@serverb ~]$
```

3. 使用 **ssh-keygen** 命令来生成 SSH 密钥。不要输入密语。

```
[operator1@serverb ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/operator1/.ssh/id_rsa): Enter
Created directory '/home/operator1/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/operator1/.ssh/id_rsa.
Your public key has been saved in /home/operator1/.ssh/id_rsa.pub.
The key fingerprint is:
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
SHA256:JainiQdnRosC+xXhOqsJQQLzBNUldb+jJbyrCZQBERI
 operator1@serverb.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|E+*+ooo .        |
|.= o.o o .       |
|o.. = . . o      |
|+. + * . o .     |
|+ = X . S +      |
| + @ +   = .     |
|. + =   o        |
|.o . . . .       |
|o     o..        |
+----[SHA256]-----+
```

4. 使用 **ssh-copy-id** 命令，将 SSH 密钥对的公钥发送给 servera 上的 operator1。在 servera 上，使用 redhat 作为 operator1 的密码。

```
[operator1@serverb ~]$ ssh-copy-id operator1@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
operator1/.ssh/id_rsa.pub"
The authenticity of host 'servera (172.25.250.10)' can't be established.
ECDSA key fingerprint is SHA256:ERTdjooOIrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.
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
operator1@servera's password: redhat
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'operator1@servera'"
and check to make sure that only the key(s) you wanted were added.
```

5. 在 servera 上，使用 SSH 远程执行 **hostname** 命令，而不访问远程交互式 shell。

```
[operator1@serverb ~]$ ssh operator1@servera hostname
servera.lab.example.com
```

注意前面的 **ssh** 命令没有提示您输入密码，因为它将无密语保护私钥与导出的公钥比对，从而以 servera 上的 operator1 用户进行身份验证。这种做法不安全，因为有权访问私钥文件的任何人都能以 operator1 身份登录 servera。安全的替代方法是使用密语保护私钥，而这是下一步。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

6. 使用 **ssh-keygen** 命令，生成另一组具有密语保护的 SSH 密钥。将密钥保存为 **/home/operator1/.ssh/key2**。使用 redhatpass 作为私钥的密语。

> **警告**
> 如果未指定用于保存密钥的文件，则使用默认文件 (**/home/user/.ssh/id_rsa**)。您在前面步骤中生成 SSH 密钥时使用了默认文件名，因此务必要指定一个非默认文件，否则会覆盖现有的 SSH 密钥。

```
[operator1@serverb ~]$ ssh-keygen -f .ssh/key2
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase): redhatpass
Enter same passphrase again: redhatpass
Your identification has been saved in .ssh/key2.
Your public key has been saved in .ssh/key2.pub.
The key fingerprint is:
SHA256:OCtCjfPm5QrbPBgqbEIWCcw5AI4oSlMEbgLrBQ1HWKI
 operator1@serverb.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|O=X*             |
|OB=.             |
|E*o.             |
|Booo   .         |
|..= . o S        |
| +.o   o         |
|+.oo+ o          |
|+o.O.+           |
|+ . =o.          |
+----[SHA256]-----+
```

7. 使用 **ssh-copy-id** 命令，将受密语保护的密钥对的公钥发送给 servera 上的 operator1。

```
[operator1@serverb ~]$ ssh-copy-id -i .ssh/key2.pub operator1@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/key2.pub"
/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

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'operator1@servera'"
and check to make sure that only the key(s) you wanted were added.
```

注意前面的 **ssh-copy-id** 命令没有提示您输入密码，因为它使用了您在之前步骤中导出到 servera 的无密语保护私钥的公钥。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

8. 在 servera 上，通过 SSH 远程执行 **hostname** 命令，而不访问远程交互式 shell。使用 **/home/operator1/.ssh/key2** 作为身份文件。指定 redhatpass 作为密语，您在之前步骤中为私钥设置了该密语。

```
[operator1@serverb ~]$ ssh -i .ssh/key2 operator1@servera hostname
Enter passphrase for key '.ssh/key2': redhatpass
servera.lab.example.com
```

注意前面的 **ssh** 命令提示您输入用于保护 SSH 密钥对的私钥的密语。该密语保护私钥。如果攻击者获得私钥的访问权限，攻击者无法使用它访问其他系统，因为私钥本身受到了密语的保护。**ssh** 命令使用了不同于 servera 上的 operator1 的密语，要求用户同时知道这两者。

您可以在下一步中使用 **ssh-agent**，以避免在使用 SSH 登录时以交互方式键入密语。在管理员经常登录远程系统的情形中，使用 **ssh-agent** 不仅更加方便，也更为安全。

9. 在您的 Bash shell 中运行 **ssh-agent**，并将 SSH 密钥对的受密语保护的私钥 (**/home/operator1/.ssh/key2**) 添加到 shell 会话。

```
[operator1@serverb ~]$ eval $(ssh-agent)
Agent pid 21032
[operator1@serverb ~]$ ssh-add .ssh/key2
Enter passphrase for .ssh/key2: redhatpass
Identity added: .ssh/key2 (operator1@serverb.lab.example.com)
```

前面的 **eval** 命令启动了 **ssh-agent** 并将此 shell 会话配置为使用此私钥。随后，您使用 **ssh-add** 将解锁的私钥提供给 **ssh-agent**。

10. 在 servera 上，远程执行 **hostname** 命令，而不访问远程交互式 shell。使用 **/home/operator1/.ssh/key2** 作为身份文件。

```
[operator1@serverb ~]$ ssh -i .ssh/key2 operator1@servera hostname
servera.lab.example.com
```

注意前面的 **ssh** 命令没有提示您以交互方式输入密语。

11. 在 workstation 上打开另一个终端，再以 student 用户身份打开连接 serverb 的 SSH 会话。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

12. 在 serverb 上，使用 **su** 命令切换到 operator1，再调用连接到 servera 的 SSH 连接，使用 **/home/operator1/.ssh/key2** 作为身份文件，以利用 SSH 密钥进行身份验证。

12.1.使用 **su** 命令，切换到 operator1。使用 redhat 作为 operator1 的密码。

```
[student@serverb ~]$ su - operator1
Password: redhat
[operator1@serverb ~]$
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

12.2.以 operator1 身份，打开连接到 servera 的 SSH 会话。

```
[operator1@serverb ~]$ ssh -i .ssh/key2 operator1@servera
Enter passphrase for key '.ssh/key2': redhatpass
...output omitted...
[operator1@servera ~]$
```

注意前面的 **ssh** 命令提示您以交互方式输入密语，因为您没有从用于启动 **ssh-agent** 的 shell 调用 SSH 连接。

13. 退出第二个终端中正在使用的所有 shell。

13.1.从 servera 注销。

```
[operator1@servera ~]$ exit
logout
Connection to servera closed.
[operator1@serverb ~]$
```

13.2.退出 serverb 上的 operator1 和 student shell，以返回到 workstation 上的 student 用户的 shell。

```
[operator1@serverb ~]$ exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

13.3.关闭 workstation 上的第二个终端。

```
[student@workstation ~]$ exit
```

14. 在第一个终端上，从 serverb 注销并结束本练习。

14.1.从第一个终端中，退出 serverb 上 root 用户的 shell。

```
[operator1@serverb ~]$ exit
logout
[student@serverb ~]$
```

exit 命令使您退出 operator1 用户的 shell，从而终止 **ssh-agent** 处于活动状态的 shell 会话，并返回到 serverb 上 student 用户的 shell。

14.2.退出 serverb 上 student 用户的 shell，以返回到 workstation 上 student 用户的 shell。

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

海量视频题库 myitpub.com QQ:556462
www.52myit.com

完成

在 workstation 上，运行 **lab ssh-configure finish** 来完成本练习。

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

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

自定义 OPENSSH 服务配置

培训目标

学完本节后，您应能够限制直接以 root 身份登录，并为 OpenSSH 服务禁用基于密码的身份验证。

配置 OPENSSH 服务器

OpenSSH 服务由一个名为 sshd 的守护进程提供。它的主配置文件为 **/etc/ssh/sshd_config**。

OpenSSH 服务器的默认配置运行良好。但是，您可能希望进行一些更改以增强系统的安全性。有两种您可能希望进行的常见更改。您可能希望禁止直接远程登录 root 帐户，并且您可能希望禁止使用基于密码的身份验证（偏向于使用 SSH 私钥身份验证）。

禁止超级用户使用 SSH 登录

最好禁止从远程系统直接登录 root 用户帐户。允许以 root 用户身份直接登录的一些风险包括：

- 所有 Linux 系统上都默认存在用户名 root，因此潜在的攻击者只需要猜测其密码，而不必猜测有效的用户名与密码组合。这为攻击者降低了复杂度。
- root 用户具有不受限制的特权，因此它的泄露可能会给系统造成最大的损坏。
- 从审核的角度看，很难跟踪是哪个授权用户以 root 的身份登录并做出了更改。如果用户必须以普通用户身份登录并切换到 root 帐户，则会生成一个日志事件，可用于帮助确定责任。

OpenSSH 服务器使用 **/etc/ssh/sshd_config** 配置文件中的 **PermitRootLogin** 配置设置，以允许或禁止用户以root 身份登录系统。

```
PermitRootLogin yes
```

当 **PermitRootLogin** 参数设为 **yes** 时，这也是默认设置，用户被允许以 root身份登录系统。要防止这种情况，可将该值设为 **no**。或者，若要禁止基于密码的身份验证，但允许对 root 执行基于私钥的身份验证，可将 **PermitRootLogin** 参数设为 **without-password**。

SSH 服务器 (sshd) 必须重新加载才能使更改生效。

```
[root@host ~]# systemctl reload sshd
```

> **重要**
> 使用 **systemctl reload sshd** 命令的好处在于它会让 sshd 重新读取其配置文件，而不是完全重启服务。尽管 **systemctl restart sshd** 命令同样会应用更改，但它也会停止和启动服务，从而中断与该主机的所有活动 SSH 连接。

禁止对 SSH 进行基于密码的身份验证

仅允许基于私钥登录远程命令行有诸多优点：

海量视频题库 myitpub.com QQ:556462
www.52myit.com

- 攻击者无法使用密码猜测攻击来远程入侵系统上的已知帐户。
- 对于受密语保护的私钥而言，攻击者同时需要密语和私钥的副本。对于密码而言，攻击者则只需要密码。
- 通过将受密语保护的私钥与 `ssh-agent` 配合使用，由于密语的输入频率较低，因此被泄露的几率会较小，而且对用户来说会更便于登录。

OpenSSH 服务器使用 **`/etc/ssh/sshd_config`** 配置文件中的 `PasswordAuthentication` 参数，用于控制用户在登录系统时能否使用基于密码的身份验证。

```
PasswordAuthentication yes
```

`/etc/ssh/sshd_config` 配置文件中 `PasswordAuthentication` 参数的默认值是 **`yes`**，这会使 SSH 服务器允许用户在登录系统时使用基于密码的身份验证。`PasswordAuthentication` 的值为 **`no`** 时禁止用户使用基于密码的身份验证。

请记住，每当您更改 **`/etc/ssh/sshd_config`** 文件时，都必须重新加载 `sshd` 服务让更改生效。

重要

请记住，如果您为 **`ssh`** 关闭基于密码的身份验证，则需要有一种办法来确保用户在远程服务器上的 **`~/.ssh/authorized_keys`** 文件中填充有公钥，以便可以登录。

参考文献

`ssh`(1)、`sshd_config`(5) man page

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

指导练习

自定义 OPENSSH 服务配置

在本练习中，您将在您的一台服务器上禁止直接以 root 身份登录，并且为 OpenSSH 服务禁用基于密码的身份验证。

成果

您应能够：

- 禁止通过 **ssh** 直接以 root 身份登录。
- 禁止远程用户通过基于密码的身份验证进行 SSH 连接。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab ssh-customize start** 脚本来开始本练习。此脚本会创建必要的用户帐户和文件。

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

1. 从 workstation，以 student 用户身份打开连接 serverb 的 SSH 会话。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

2. 使用 **su** 命令，切换为 serverb上的 operator2。使用 redhat 作为 operator2 的密码。

```
[student@serverb ~]$ su - operator2
Password: redhat
[operator2@serverb ~]$
```

3. 使用 **ssh-keygen** 命令来生成 SSH 密钥。不要为密钥输入任何密语。

```
[operator2@serverb ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/operator2/.ssh/id_rsa): Enter
Created directory '/home/operator2/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/operator2/.ssh/id_rsa.
Your public key has been saved in /home/operator2/.ssh/id_rsa.pub.
The key fingerprint is:
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
SHA256:JainiQdnRosC+xXhOqsJQQLzBNUldb+jJbyrCZQBERI
 operator1@serverb.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|E+*+ooo .        |
|.= o.o o .       |
|o.. = . . o      |
|+. + * . o .     |
|+ = X . S +      |
| + @ +   = .     |
|. + =   o        |
|.o . . . .       |
|o     o..        |
+----[SHA256]-----+
```

4. 使用 **ssh-copy-id** 命令，将 SSH 密钥对的公钥发送给 servera 上的 operator2。在 servera 上，使用 redhat 作为 operator2 的密码。

```
[operator2@serverb ~]$ ssh-copy-id operator2@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
operator1/.ssh/id_rsa.pub"
The authenticity of host 'servera (172.25.250.10)' can't be established.
ECDSA key fingerprint is SHA256:ERTdjooOIrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.
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
operator2@servera's password: redhat
Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'operator2@servera'"
and check to make sure that only the key(s) you wanted were added.
```

5. 确认您可以使用 SSH 密钥以 operator2 身份成功登录 servera。

5.1. 以 operator2 身份，打开连接到 servera 的 SSH 会话。

```
[operator2@serverb ~]$ ssh operator2@servera
...output omitted...
[operator2@servera ~]$
```

注意前面的 **ssh** 命令使用 SSH 密钥进行身份验证。

5.2. 从 servera 注销。

```
[operator2@servera ~]$ exit
logout
Connection to servera closed.
```

6. 确认您可以使用密码 redhat 以 root 身份成功登录 servera。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

6.1. 使用 redhat 作为密码，以 root 身份打开连接到 servera 的 SSH 会话。

```
[operator2@serverb ~]$ ssh root@servera
root@servera's password: redhat
...output omitted...
[root@servera ~]#
```

注意前面的 **ssh** 命令使用超级用户的密码进行身份验证，因为不存在针对该超级用户的 SSH 密钥。

6.2. 从 servera 注销。

```
[root@servera ~]# exit
logout
Connection to servera closed.
[operator2@serverb ~]$
```

7. 确认您可以使用密码 redhat 以 operator3 身份成功登录 servera。

7.1. 使用 redhat 作为密码，以 operator3 身份打开连接到 servera 的 SSH 会话。

```
[operator2@serverb ~]$ ssh operator3@servera
operator3@servera's password: redhat
...output omitted...
[operator3@servera ~]$
```

注意前面的 **ssh** 命令使用了 operator3 的密码进行身份验证，因为不存在针对 operator3 的 SSH 密钥。

7.2. 从 servera 注销。

```
[operator3@servera ~]$ exit
logout
Connection to servera closed.
[operator2@serverb ~]$
```

8. 配置 servera 上的 sshd，以禁止用户以 root 身份登录。必要时，使用 redhat 作为该超级用户的密码。

8.1. 使用 SSH 密钥，以 operator2 身份打开连接到 servera 的 SSH 会话。

```
[operator2@serverb ~]$ ssh operator2@servera
...output omitted...
[operator2@servera ~]$
```

8.2. 在 servera 上，切换到 root。使用 redhat 作为 root 用户的密码。

```
[operator2@servera ~]$ su -
Password: redhat
[root@servera ~]#
```

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

8.3. 将 **/etc/ssh/sshd_config** 中的 **PermitRootLogin** 设为 **no**，再重新加载 sshd。您可以使用 **vim /etc/ssh/sshd_config** 来编辑 sshd 的配置文件。

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

8.4. 在 workstation 上打开另一个终端，再以 operator2 身份打开连接 serverb 的 SSH 会话。从 serverb，尝试以 root 身份登录 servera。这应该会失败，因为您在之前步骤中禁止了 root 用户通过 SSH 登录。

> **注意**
> 为方便起见，课堂环境中的 workstation 和 serverb 之间已配置了免密码登录。

```
[student@workstation ~]$ ssh operator2@serverb
...output omitted...
[operator2@serverb ~]$ ssh root@servera
root@servera's password: redhat
Permission denied, please try again.
root@servera's password: redhat
Permission denied, please try again.
root@servera's password: redhat
root@servera: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

默认情况下，**ssh** 命令会先尝试使用基于密钥的身份验证，如果尝试失败，再使用基于密码的身份验证。

9. 配置 servera 上的 sshd，以允许用户仅使用 SSH 密钥进行身份验证，而不使用其密码。

9.1. 返回到第一个终端，此终端中含有 servera 上活跃的 root 用户 shell。将 **/etc/ssh/sshd_config** 中的 PasswordAuthentication 设为 **no**，再重新加载 sshd。您可以使用 **vim /etc/ssh/sshd_config** 来编辑 sshd 的配置文件。

```
...output omitted...
PasswordAuthentication no
...output omitted...
[root@servera ~]# systemctl reload sshd
```

9.2. 转到第二个终端，其中含有 serverb 上活跃的 operator2 用户 shell，再尝试以 operator3 身份登录 servera。这应该会失败，因为没有为 operator3 配置 SSH 密钥，而且 servera 上的 sshd 服务也不允许使用密码进行身份验证。

```
[operator2@serverb ~]$ ssh operator3@servera
operator3@servera: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
```

海量视频题库 myitpub.com QQ:556462 www.52myit.com

> **注意**
> 要获得更多粒度，您可以将显式 **-o PubkeyAuthentication=no** 和 **-o PasswordAuthentication=yes** 选项用于 **ssh** 命令。这允许您覆盖 **ssh** 命令的默认值，并根据之前步骤中在 **/etc/ssh/sshd_config** 中调整的设置，自信地确定前面的命令会失败。

9.3. 返回到第一个终端，此终端中含有 servera 上活跃的 root 用户 shell。验证 **/etc/ssh/sshd_config** 中启用了 **PubkeyAuthentication**。您可以使用 **vim /etc/ssh/sshd_config** 来查看 sshd 的配置文件。

```
...output omitted...
#PubkeyAuthentication yes
...output omitted...
```

注意 PubkeyAuthentication 行已注释掉。此文件中的任何注释行都使用默认值。注释行表示参数的默认值。默认情况下，SSH 的公钥身份验证为有效状态，如注释行所示。

9.4. 返回到第二个终端，其中含有 serverb 上活跃的 operator2 用户 shell，再尝试以 operator2 身份登录 servera。这应该会成功，因为已经配置了 SSH 密钥，供 operator2 从 serverb 登录 servera。

```
[operator2@serverb ~]$ ssh operator2@servera
...output omitted...
[operator2@servera ~]$
```

9.5. 从第二个终端中，退出 servera 和 serverb 上的 operator2 用户 shell。

```
[operator2@servera ~]$ exit
logout
Connection to servera closed.
[operator2@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

9.6. 关闭 workstation 上的第二个终端。

```
[student@workstation ~]$ exit
```

9.7. 从第一个终端中，退出 servera 上 root 用户的 shell。

```
[root@servera ~]# exit
logout
```

9.8. 从第一个终端中，退出 servera 和 serverb 上的 operator2 用户 shell。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[operator2@servera ~]$ exit
logout
Connection to servera closed.
[operator2@serverb ~]$ exit
logout
[student@serverb ~]$
```

9.9. 从 `serverb` 注销，再返回到 `workstation` 上 `student` 用户的 shell。

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

完成

在 `workstation` 上，运行 **`lab ssh-customize finish`** 来完成本练习。

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

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

开放研究实验

配置和保护 SSH

任务执行清单

在本实验中，您将在您的一台服务器上为用户设置基于密钥的身份验证，禁止直接以 `root` 身份登录，并且为 OpenSSH 服务禁用密码所有用户的密码身份验证。

成果

您应能够：

- 使用 SSH 密钥进行身份验证。
- 禁止用户通过 **ssh** 直接以 `root` 身份登录。
- 禁止用户使用基于 SSH 密码的身份验证登录系统。

在你开始之前

以 `student` 用户身份并使用 `student` 作为密码登录 `workstation`。

在 `workstation` 上，运行 **lab ssh-review start** 脚本来开始本练习。此脚本会创建必要的用户帐户和文件。

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

1. 从 `workstation`，以 `student` 用户身份打开连接 `servera` 的 SSH 会话。
2. 使用 **su** 命令，切换为 `servera`上的 `producttion1`。
3. 使用 **ssh-keygen** 命令，为 `servera` 上的 `production1` 生成无密语 SSH 密钥。
4. 使用 **ssh-copy-id** 命令，将 SSH 密钥对的公钥发送给 `serverb` 上的 `production1`。
5. 确认 `production1` 可以使用 SSH 密钥成功登录 `serverb`。
6. 配置 `serverb` 上的 `sshd`，以禁止用户以 `root` 身份登录。使用 `redhat` 作为该超级用户的密码。
7. 配置 `serverb` 上的 `sshd`， 以允许用户仅使用 SSH 密钥进行身份验证，而不使用其密码。

评估

在 `workstation` 上，运行 **lab ssh-review grade** 命令以确认本练习是否成功。

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

完成

在 `workstation` 上，运行 **lab ssh-review finish** 脚本来完成本实验。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

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

本实验到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

解决方案

配置和保护 SSH

任务执行清单

在本实验中，您将在您的一台服务器上为用户设置基于密钥的身份验证，禁止直接以 root 身份登录，并且为 OpenSSH 服务禁用密码所有用户的密码身份验证。

成果

您应能够：

- 使用 SSH 密钥进行身份验证。
- 禁止用户通过 **ssh** 直接以 root 身份登录。
- 禁止用户使用基于 SSH 密码的身份验证登录系统。

在你开始之前

以 student 用户身份并使用 student 作为密码登录 workstation。

在 workstation 上，运行 **lab ssh-review start** 脚本来开始本练习。此脚本会创建必要的用户帐户和文件。

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

1. 从 workstation，以 student 用户身份打开连接 servera 的 SSH 会话。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

2. 使用 **su** 命令，切换为 servera上的 producttion1。

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

3. 使用 **ssh-keygen** 命令，为 servera 上的 production1 生成无密语 SSH 密钥。

```
[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.
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
The key fingerprint is:
SHA256:CsWCAmWOg5qaJujLzIAcengNj3u21kbrPP4Ysl3PXCA
 production1@servera.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|..o              |
|o+ . .           |
|= o . o          |
|.+   o           |
|o.. .   E .      |
|*o.= ... . .     |
|Xo+ +oo..   .    |
|Oo .+==+ + .     |
| *o+o=*o. +      |
+----[SHA256]-----+
```

4. 使用 **ssh-copy-id** 命令，将 SSH 密钥对的公钥发送给 serverb 上的 production1。

```
[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.
ECDSA key fingerprint is SHA256:ERTdjooOIrIwVSZQnqD5or+JbXfidg0udb3DXBuHWzA.
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 into the machine, with:   "ssh 'production1@serverb'"
and check to make sure that only the key(s) you wanted were added.
```

5. 确认 production1 可以使用 SSH 密钥成功登录 serverb。

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

6. 配置 serverb 上的 sshd，以禁止用户以 root 身份登录。使用 redhat 作为该超级用户的密码。

   6.1. 使用 **su -** 命令，切换为 serverb 上的 root。

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

   6.2. 将 **/etc/ssh/sshd_config** 中的 **no** 设为 PermitRootLogin，再重新加载 sshd。您可以使用 **vim /etc/ssh/sshd_config** 来编辑 sshd 的配置文件。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

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

6.3. 在 workstation 上打开另一个终端，再以 production1 身份打开连接 servera 的 SSH 会话。从 servera，尝试以 root 身份登录 serverb。这应该会失败，因为您在之前步骤中禁止了 root 用户通过 SSH 登录。

> **注意**
> 为方便起见，课堂环境中的 workstation 和 servera 之间已配置了免密码登录。

```
[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 ~]$
```

前面的 **ssh** 命令在尝试以 root 身份登录 servera 失败三次后返回到了提示符。默认情况下，**ssh** 命令会优先使用 SSH 密钥进行身份验证，但如果找不到用户的必要密钥，它会请求用户的密码进行身份验证。

7. 配置 serverb 上的 sshd，以允许用户仅使用 SSH 密钥进行身份验证，而不使用其密码。

7.1. 返回到第一个终端，此终端中含有 serverb 上活跃的 root 用户 shell。将 **/etc/ssh/sshd_config** 中的 **no** 设为 PasswordAuthentication，再重新加载 sshd。您可以使用 **vim /etc/ssh/sshd_config** 来编辑 sshd 的配置文件。

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

7.2. 转到第二个终端，其中含有 servera 上活跃的 production1 用户 shell，再尝试以 production2 身份登录 serverb。这应该会失败，因为没有为 production2 配置 SSH 密钥，而且 serverb 上的 sshd 服务也不允许使用密码进行身份验证。

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

海量视频题库 myitpub.com QQ:556462 www.52myit.com

> **注意**
> 要获得更多粒度，您可以将显式 **-o PubkeyAuthentication=no** 和 **-o PasswordAuthentication=yes** 选项用于 **ssh** 命令。这允许您覆盖 **ssh** 命令的默认值，并根据之前步骤中在 **/etc/ssh/sshd_config** 中调整的设置，自信地确定前面的命令是否确实会失败。

7.3. 返回到第一个终端，此终端中含有 serverb 上活跃的 root 用户 shell。验证 **/etc/ssh/sshd_config** 中启用了 **PubkeyAuthentication**。您可以使用 **vim /etc/ssh/sshd_config** 来查看 sshd 的配置文件。

```
...output omitted...
#PubkeyAuthentication yes
...output omitted...
```

注意 PubkeyAuthentication 行已注释掉。此文件中的任何注释行都使用默认值。注释行表示参数的默认值。默认情况下，SSH 的公钥身份验证为有效状态，如注释行所示。

7.4. 返回到第二个终端，其中含有 servera 上活跃的 production1 用户 shell，再尝试以 production1 身份登录 serverb。这应该会成功，因为已经配置了 SSH 密钥，供 production1 从 servera 登录 serverb。

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

7.5. 从第二个终端中，退出 serverb 和 servera 上的 production1 用户 shell。

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

7.6. 关闭 workstation 上的第二个终端。

```
[student@workstation ~]$ exit
```

7.7. 从第一个终端中，退出 serverb 上 root 用户的 shell。

```
[root@serverb ~]# exit
logout
```

7.8. 从第一个终端中，退出 serverb 和 servera 上的 production1 用户 shell。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

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

7.9. 从 servera 注销，再返回到 workstation 上 student 用户的 shell。

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

评估

在 workstation 上，运行 **lab ssh-review grade** 命令以确认本练习是否成功。

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

完成

在 workstation 上，运行 **lab ssh-review finish** 脚本来完成本实验。

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

本实验到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

总结

在本章中，您学到了：

- **ssh** 命令允许用户使用 SSH 协议安全地访问远程系统。
- 客户端系统将远程服务器的身份存储在 **~/.ssh/known_hosts** 和 **/etc/ssh/ssh_known_hosts** 中。
- SSH 同时支持基于密码和基于密钥的身份验证。
- **ssh-keygen** 命令可生成用于身份验证的 SSH 密钥对。**ssh-copy-id** 命令将公钥导出到远程系统。
- sshd 服务在红帽企业 Linux 系统上实施 SSH 协议。
- 建议您配置 sshd 以禁止以 root 身份进行远程登录，并要求使用公钥身份验证，而非基于密码的身份验证。

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

章 11

分析和存储日志

目标

查找和准确解读系统事件的日志，以满足故障排除之需。

培训目标

- 描述红帽企业 Linux 用于记录事件的基本日志架构。
- 解读相关 syslog 文件中的事件，以排除故障或查看系统状态。
- 查找和解读系统日志中的条目，以排除故障或查看系统状态。
- 配置系统日志，以在服务器系统重启时保留事件记录。
- 利用 NTP 维护准确的时间同步，并且配置时区以确保系统日志和日志记录的事件标有正确的时间戳。

章节

- 描述系统日志架构（及测验）
- 检查 Syslog 文件（及引导式练习）
- 检查系统日志条目（及引导式练习）
- 保留系统日志（及引导式练习）
- 维护准确的时间（及引导式练习）

实验

分析和存储日志

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

描述系统日志架构

培训目标

学完本节后，您应能够描述红帽企业 Linux 用于记录事件的基本日志架构。

系统日志记录

进程和操作系统内核为发生的事件记录日志。这些日志用于系统审核和问题的故障排除。

许多系统都以文本文件的方式记录事件日志，而这些文件就保存在 **/var/log** 目录中。可以使用常规文本实用程序（如 **less** 和 **tail**）检查这些日志。

红帽企业 Linux 中内建了一个基于 Syslog 协议的标准日志记录统。许多程序使用此系统记录事件，并将它们整理到日志文件中。`systemd-journald` 和 `rsyslog` 服务处理红帽企业 Linux 8 中的 syslog 消息。

`systemd-journald` 服务是操作系统事件日志架构的核心。它收集许多来源的事件消息，包括内核、引导过程早期阶段的输出、守护进程启动和运行时的标准输出及标准错误，以及 syslog 事件。然后，它会将它们重构为一种标准格式，并写进带有索引的结构化系统日志中。默认情况下，该日志存储在系统重启后不保留的文件系统上。

不过，`rsyslog` 服务会从日志中读取 `systemd-journald` 收到的 syslog 消息。之后，它将处理 syslog 事件，将它们记录到日志文件中，或根据自己的配置将它们转发给其他服务。

`rsyslog` 服务对 syslog 消息进行排序，并将它们写入到在系统重启后不保留的日志文件中 (**/var/log**)。`rsyslog` 服务会根据发送每条消息的程序类型或设备以及每条 syslog 消息的优先级，将日志消息排序到特定的日志文件。

除了 syslog 消息文件外，**/var/log** 目录中还包含系统上其他服务的日志文件。下表列出了 **/var/log** 目录中一些有用的文件。

所选的系统日志文件

日志文件	存储的消息类型
/var/log/messages	大多数系统日志消息记录在此处。例外包括与身份验证、电子邮件处理和调度作业执行相关的消息以及纯粹与调试相关的消息。
/var/log/secure	与安全性和身份验证事件相关的 syslog 消息。
/var/log/maillog	与邮件服务器相关的 syslog 消息。
/var/log/cron	与调度作业执行相关的 syslog 消息。
/var/log/boot.log	与系统启动相关的非 syslog 控制台消息。

海量视频题库 myitpub.com QQ:556462
www.52myit.com