安全外壳

安全外壳 (SSH) 是一种允许您远程登录另一台计算机（例如实验室机器）的协议。几乎每个使用 SSH 的人都使用免费的 OpenSSH 实现，该实现几乎是每个 Linux 发行版的标准配置，也适用于 Windows 和 Mac，甚至适用于移动操作系统 iOS 和 Android。

稍后我们将更详细地了解 SSH 如何管理连接，但现在想象一下它在您自己的计算机与在另一台计算机上运行的 shell 之间打开网络连接。当您输入某些内容时，SSH 会对其进行加密并将其发送到另一台计算机，另一台计算机将其解密并将其传递给 shell（或您正在运行的任何其他程序）；当 shell 回复时，SSH 会对其进行加密并将其发送回给您。为此，(Open)SSH 实际上是两个程序：

- `ssh` 是客户端，您在自己的计算机上运行它以连接到另一台计算机。
- `sshd` 是服务器，或者是 UNIX 中的*守护进程*。它在您要连接的计算机的后台运行，需要由系统管理员安装。

Advanced note

SSH 默认使用 TCP 端口 22。

检查你的客户

首先，让我们检查 ssh 客户端是否正常工作。

- 在你自己的机器上打开一个终端：Linux、Mac OS 和 Linux 的 Windows 子系统应该没问题。如果您安装了 Windows 版本的 openssh（例如，如果您安装了在后台使用 ssh 的 git），Windows 10 CMD 也可能可以工作。
- 键入 `ssh localhost` 并按 Enter 键。可能会发生几种不同的情况：
  - 如果它要求输入密码，则 ssh 客户端正在工作，并且 ssh 服务器正在您当前的计算机上运行。该密码将是您的用户帐户密码，但我们实际上不想再次登录，因此请使用 Control+C 取消。
  - 如果在没有密码的情况下成功，则客户端正在工作并且 ssh 服务器正在您的计算机上运行，并且您没有密码，或者您已经设置了密钥。键入 `exit` 并按 ENTER 键返回到之前的 shell。
  - 如果它显示“连接被拒绝”，那么您的 ssh 客户端正常工作，但您自己的计算机上没有运行服务器。这不是问题，因为我们正在尝试登录实验室机器，因此我们的机器上需要一个客户端，实验室机器上需要一个服务器。
  - 如果它显示未找到 ssh 的错误，则表明您没有安装 (Open)SSH，这在 Windows CMD 上除外 - 在这种情况下，请切换到使用适用于 Linux 的 Windows 子系统。

连接到实验室

实验室机器的名称 `it######.wks.bris.ac.uk` 中的哈希值代表从 075637 到 075912 的数字。但是，并非所有机器都能同时工作，如果每个人都连接到同一台机器，那么它很快就会过载，并且出于安全原因，实验室机器无法直接通过互联网访问。相反，我们将使用另外两台机器进行连接：

- 堡垒主机 `seis.bris.ac.uk` 。这可以通过互联网上的 SSH 进行访问，并且位于大学网络上，可让您进一步连接到实验室计算机。您不应该尝试对 seis 本身进行任何工作，因为您想要使用的大多数软件（如编译器）都没有安装在那里。但是，您在 seis 上确实有一个主目录，用于存储 SSH 密钥等内容。
- 负载平衡器 `rd-mvb-linuxlab.bristol.ac.uk` 。这会将您连接到当前正在运行的实验室计算机，并确保如果每个人都使用此方法进行连接，那么他们将或多或少均匀地分布在正在运行的计算机之间。

请尝试以下操作：

- 在您的终端上，输入 `ssh USERNAME@seis.bris.ac.uk` 将 USERNAME 替换为您的大学用户名的位置，例如 `aa20123` 。显然，为此您需要有效的互联网连接。
- 如果询问您是否确定，请键入 `yes` 并按 ENTER。SSH 仅在您第一次连接到以前从未使用过的计算机时才会执行此操作。
- 出现提示时，输入您的大学密码，然后按 ENTER。
- 您现在应该会看到 seis 上的提示，看起来类似于 `-bash-4.2$` 。尝试使用该命令 `uname -a` 来打印有关系统的信息（uname 本身会打印操作系统名称， `-a` 显示“所有”信息）。回复行应该以 开头 `Linux seis-shell` ，这是操作系统和主机名。
- 在 seis 提示符下，键入 `ssh rd-mvb-linuxlab.bristol.ac.uk` 。这可能需要几秒钟；如果询问您是否确定，请说“是”，然后在出现提示时再次输入密码。我们不必再次提供用户名，因为您已经使用您的大学用户名登录到 seis（ `whoami` 显示这一点），并且当您在没有提供用户名的情况下进行 ssh 时，它会使用您当前登录的用户名。
- 您现在应该连接到实验室机器，并出现以下形式的提示 `USERNAME@it######:~$` 。
- 尝试 `whoami` 检查 `uname -a` 您的登录身份和位置；也尝试 `hostname` 只打印机器名称。
- 输入 `exit` 两次即可返回到您自己的机器。（一次让您返回 seis，两次则完全关闭 ssh 连接。）

通过另一台机器（在本例中为 seis）作为代理连接到一台机器是一种常见的用例，ssh 实际上有一个选项。但请注意，尽管它可以在适用于 Linux 的 Windows 子系统（以及 Mac 和 Linux）上运行，但通常无法在 Windows CMD 终端上运行。

```
ssh -J USERNAME@seis.bris.ac.uk USERNAME@rd-mvb-linuxlab.bristol.ac.uk
```

`-J` 如果您需要通过多个主机进行连接，“跳转通过此主机”甚至可以接受以逗号分隔的主机列表。但是，您需要为每台计算机重复您的用户名。

您现在知道如何登录实验室机器，但在这两种方法中您都必须输入密码两次 - 让我们让这变得更容易。答案不是将密码存储在文件中，而是使用密钥。

设置 ssh 密钥

当您连接到一台计算机时，您计算机上的客户端和您登录的计算机上的守护程序将运行加密协议来交换密钥并为会话设置共享密钥，以便一方加密另一方方可以再次解密。它还使用多种方法之一向服务器验证您的身份。

您可能从安全消息来源听说，存在三个主要的身份验证因素：您知道的东西（密码或 PIN）、您拥有的东西（物理密钥、数字密钥、身份证、护照）以及您的身份（生物识别）。需要其中两项的身份验证方法称为双因素身份验证，这被认为是良好的安全实践。对于 ssh，这意味着：

- 您可以使用用户名和密码登录，即"您知道的内容"。这是默认设置，但不是最安全的。
- 您可以使用（数字）密钥登录，即"您拥有的东西"。这是更安全的，只要您的密钥（只是一个文件）不会落入坏人之手，而且也是最方便的，因为您可以登录实验室机器或复制文件而无需输入密码。
- 您可以使用本身受密码保护的密钥文件登录。这将为您提供双因素身份验证。

SSH 使用的密钥实现数字签名。每个密钥都作为一对文件出现：

- 通常命名为 CIPHER 的文件中的私钥（也称为秘密密钥） `id_CIPHER` 是正在使用的密码。您需要保证其安全，并且仅将其存储在只有您有权访问的地方。
- 通常名为 的文件中的公钥 `id_CIPHER.pub` 。你可以与全世界共享它，并且你需要在任何机器上或任何你想要登录的服务上存储它的副本（对于实验室，因为实验室机器都共享一个文件系统，你只需要存储一次 - 但 seis 有一个单独的文件系统，因此您需要在那里有一个单独的副本）。

让我们创建一个密钥对：

- 在您自己的计算机上，确保您未连接到实验室计算机或 seis，然后键入命令 `ssh-keygen -t ed25519` 。（如果您收到"未知密钥类型"错误，则说明您使用的是过时版本的 OpenSSH，出于安全原因，您应该立即升级。）*注意： `ed25519` 直接键入，不要将其替换为您的用户名。 `2^255-19` 如果您想知道的话，它代表"素数上的爱德华兹曲线"加密群。*
- 当它询问您保存文件的位置时，只需按 ENTER 接受默认值，但记下路径 - 通常它是 `.ssh` 您主目录中的一个文件夹。
- 如果它询问您"覆盖（y/n）"，请说"否"（n，然后按 ENTER），因为这意味着您已经拥有其他东西的密钥 - 可以直接 ssh 或使用它的东西，例如 github。重新启动密钥生成，但选择不同的文件名。
- 当它要求您输入密码时，我们建议您只需按 ENTER 键即可，这不会设置密码（安全性好，方便）。如果您设置了密码，它会要求您输入两次，然后您将需要密码和密钥文件才能使用此密钥（最大安全性，不太方便）。

Advanced note

该 `-t` 参数选择要使用的加密算法，在本例中 `ed25519` ，该算法是现代的、经过同行评审的，并且通常被认为是可用的最安全的公钥算法之一。然而，一些旧的 ssh 版本不接受 ed25519。

如果您需要对不喜欢 ed25519 的计算机使用 SSH 密钥，请改用密钥类型"rsa"。我们建议您尽可能避免使用"dsa"和"ecdsa"，因为密码学家猜测设计中可能存在缺陷。

例如，虽然 seis 支持 ed25519，但旧的 cs 堡垒主机 `snowy.cs.bris.ac.uk` 仍然使用旧版本的 SSH，因此您需要生成 rsa 密钥才能连接到该主机。

查看存储密钥的文件夹。 `ls -l ~/.ssh` 将会执行此操作，除非您在创建它们时选择其他位置来存储它们：

```
-rw-------. 1 vagrant vagrant  411 Oct  7 10:50 id_ed25519
-rw-r--r--. 1 vagrant vagrant   98 Oct  7 10:50 id_ed25519.pub
-rw-r--r--. 1 vagrant vagrant 1597 Oct  7 11:54 known_hosts
```

请注意我的示例中这些文件的权限。私钥（第一行）在开头有一个权限行， `(-)(rw-)(---)(---)` 我在其中添加了括号以更清楚地了解发生了什么。第一个括号仅适用于特殊文件类型（例如 `d` 目录）。接下来，所有者权限在本例中是读取和写入（第三个权限是 `x` 如果文件是可执行程序）。最后两个括号是该组和其他所有人的权限，这些都已关闭，因此除了您自己（和根）之外没有人可以读取您的密钥文件。OpenSSH 对此很挑剔，并且会拒绝使用其他人可以访问的私钥。

公钥权限 `(-)(rw-)(r--)(r--)` 意味着所有者可以读取和写入，并且组和其他所有人（假设他们有权访问该文件夹）可以读取公钥，这很好。毕竟这是一个公钥。

`known_hosts` SSH 存储您已连接的计算机的公钥的位置：每次您对“您确定要连接吗？”回答“是”时当您第一次连接到新计算机时，它会将结果存储在该文件中，并且下次不会再询问您。文件格式是每行一键，如果需要，您可以自己编辑该文件。

在 SEIS 上设置密钥访问

首先，我们需要将公钥上传到 `~/.ssh` seis 上的目录。在此之前，我们需要确保该目录存在：

- 使用 ssh 和您的密码登录 seis。
- 尝试 `ls -al ~/.ssh` 。如果它抱怨该文件夹不存在，请使用 `mkdir ~/.ssh` .
- 使用 再次注销 seis `exit` 。

复制文件的命令用于 `scp` 安全复制，其工作方式类似于 `cp` 但允许您包含远程主机并通过 SSH 进行复制。从您自己的计算机运行此命令：

```
scp ~/.ssh/id_ed25519.pub "USERNAME@seis.bris.ac.uk:~/.ssh/"
```

显然，将 USERNAME 替换为您的大学用户名。这将再次要求您输入密码。这里请注意两件事：首先，为了设置对 seis 的访问，我们上传的是公钥 - 而不是私钥！- 其次，我们在目的地周围加上双引号。这是因为 ~ 表示主目录的字符是由我们的 shell 处理的，但我们不希望本地 shell 扩展它，而是希望由 scp 启动的 seis 上的 shell 将其扩展到该机器上的主目录。

scp 的一般语法是 `scp source destination` ，源或目标可以采用以下形式 `[USERNAME@]HOSTNAME:PATH` - 如果它包含冒号 (`:`)，则它引用不同计算机上的文件。

现在通过 ssh 登录 seis 并最后一次输入您的密码。然后运行以下命令：

```
cd .ssh
cat id_ed25519.pub >> authorized_keys
chmod 600 authorized_keys
```

`authorized_keys` 如果公钥列在用户文件夹的文件中 `.ssh`， SSH 将接受该公钥，格式为每个密钥一行。我们不只是复制 `id_ed25519.pub` 到 `authorized_keys`，这会覆盖后一个文件（如果它已经存在），而是使用该结构 `cat SOURCE >> DEST` 让 shell 将源文件附加到目标。*注意：它是**授权的**，美国拼写 - 如果您在这里使用英国拼写，OpenSSH 将不知道读取该文件。*

但是，如果授权密钥文件尚不存在，那么它现在已使用默认权限创建，出于安全原因，SSH 将不会接受该文件。 `chmod` 意味着更改权限（也称为“mod 位”），600 是我们想要的 8 进制位模式，因为由于历史原因，这就是权限的工作方式。权限是一个 9 位的位字段，前三位对于所有者来说是读/写/执行，接下来的三位对于组来说是相同的，然后是其他人。如果您 `ls -l` 会以更易于理解的格式看到此内容，即 `rw-------` 减号表示关闭了某个位。

现在键入 `exit` 以返回到您自己的计算机，然后 `ssh USERNAME@seis.bris.ac.uk` 重新登录到 seis。它应该让您登录而无需输入密码，并且您现在已经为 seis 设置了基于密钥的 SSH 身份验证。

注意：如果您之前在 SSH 密钥上设置了密码，那么它会要求您输入密码，并且它会期望密钥密码而不是您的 uni 密码。您知道不要将您的 uni 密码重复用于其他用途，对吧？

如果由于某种原因某些东西无法使用 ssh，首先要尝试的是添加 `-v` 开关启用调试信息（您甚至可以执行 `-vv` 或 `-vvv` 查看更多详细信息，但我们不需要这样做）。例如，如果您的私钥文件的权限存在问题，那么您将在调试信息中看到 SSH 抱怨。

设置实验室机器的密钥

您现在可以使用密钥进入 seis，但您希望在实验室机器上完成工作。

要从您的计算机连接到 seis，您需要计算机上的私钥和 seis 上的公钥。要从 seis 连接到实验室机器，您似乎需要实验室机器上的公钥和 seis 上的私钥。但出于安全原因，您不想将私钥上传到 seis，因此我们将使用称为*代理转发*的 SSH 功能，这意味着如果您通过 SSH 连接到一台计算机，那么当您尝试通过 SSH 进一步连接到另一台计算机时将重复使用相同的密钥。执行此操作的方法是使用 `-A` 命令行标志。

Advanced note

基于密钥的身份验证的要点是，您的私钥永远不会离开您自己的计算机，因此即使是大学管理员也永远看不到它，如果您在大学计算机上存储了副本，则无法保证这一点。

登录到计算机不会将密钥发送到该计算机。相反，机器会向您发送一个质询（一个长随机数），然后 SSH 使用您自己机器上的私钥对其进行数字签名，然后将签名发回，远程机器可以

使用公钥对其进行验证。看到这样的签名不会让机器代表您创建进一步的签名，并且它绝对不会泄露密钥。

代理转发的作用是允许跨多个连接转发质询和签名，但密钥永远不会离开您自己的计算机。

通过这种方式，您可以创建一个 SSH 密钥并将其用于大学、github 以及您通过 SSH 访问的任何其他内容，即使一项服务被破坏，攻击者也无法访问您在其他服务上的帐户。

- 使用 登录 seis `ssh USERNAME@seis.bris.ac.uk` 。您不应该再需要密码了。
- 登录实验室机器并 `ssh rd-mvb-linuxlab.bristol.ac.uk` 输入您的密码。检查该 `~/.ssh` 文件夹是否存在，如果不存在则创建它，就像您之前在 seis 上所做的那样，然后 `exit` 再次到 seis。
- 将您的公钥文件从 seis 复制到实验室机器上 `scp ~/.ssh/id_ed25519.pub "rd-mvb-linuxlab.bristol.ac.uk:~/.ssh/"` 。这将再次要求您输入密码。
- `ssh rd-mvb-linuxlab.bristol.ac.uk` 最后一次登录到实验室机器并输入您的密码。在实验室机器上，使用以下命令安装公钥：

```
cd .ssh
cat id_ed25519.pub >> authorized_keys
chmod 600 authorized_keys
```

- 注销实验室机器并通过键入 `exit` 两次再次启动。

上述步骤是必要的，因为 seis 上的主目录与实验室计算机上的主目录不同。但是，您的主目录在所有实验室计算机上都是相同的，因此您无需在每台计算机上单独安装密钥。您可能已经注意到，当从 seis 复制或 `ssh` 复制到实验室机器时，您不必重复您的用户名：这是因为它在所有这些机器上都是相同的。

从现在开始，在您自己的计算机上，您应该能够使用以下命令直接进入实验室计算机，而根本不会询问您的密码：

```
ssh -A -J USERNAME@seis.bris.ac.uk USERNAME@rd-mvb-linuxlab.bristol.ac.uk
```

不幸的是， `-J` 它不能在 Windows CMD 终端上工作，尽管它应该在适用于 Linux 的 Windows 子系统上工作。一旦我们设置了配置文件，就有办法解决这个问题。Mac 和 Linux 用户应该没问题，任何人都可以在自己的计算机上从 Linux VM 运行这些命令，无论其主机操作系统如何。

设置配置文件

您现在已经有了一个可以使用的登录命令，但您仍然需要输入很多内容，并且需要输入用户名两次。我们可以通过使用配置文件来改进这一点。

SSH 读取两个配置文件：一个适用于所有用户，位于 `/etc/ssh/ssh_config` （ `/etc` POSIX 程序通常存储全局设置的位置），另一个适用于每个用户，位于 `~/.ssh/config` 。站点

https://www.ssh.com/ssh/config/或仅 `man ssh_config | less` 在终端上包含文档（ `man` 指手册页， `less` 是一个每次在页面上显示一个文件并允许您滚动和搜索的程序）。

`config` 在您自己的计算机上的目录中创建一个名为 simple 的文件 `.ssh` 。例如，您可以使用 `touch config` （确保您 `.ssh` 首先位于目录中，然后 `cd ~/.ssh` 到达那里）来执行此操作，然后在您最喜欢的文本编辑器中对其进行编辑。添加以下行，将 USERNAME 替换为您的用户名两次：

```
Host seis
  HostName seis.bris.ac.uk
  User USERNAME

Host lab
  HostName rd-mvb-linuxlab.bristol.ac.uk
  ProxyJump seis
  User USERNAME
```

现在，您可以简单地 `ssh lab` 通过 seis 登录到实验室计算机（使用时隐含代理转发 `ProxyJump` ），或者 `ssh seis` 如果您想直接访问 seis 来更新您的密钥。

- 从您自己的机器上尝试一下 `ssh lab` 。从现在起，这将是您登录实验室机器的主要方式。

Advanced note

如果您想在学习过程中学习另一项有用的技能，这里有一种在命令行上编辑文件的方法。许多Linux发行版都有一个名为 `nano` built in的编辑器，它在终端中运行，因此 `nano config` 编辑名为config的文件（如果它不存在，则在第一次保存时创建它）。这是相当不言自明的，命令Control+X 退出，正如您在 Nano 屏幕底部的命令栏上看到的那样，如果您退出时未保存更改，它会询问您是否要保存。

Nano 安装在 SEIS 和实验室机器上，并且在 Debian 发行版中您将安装在 Vagrant 中，因此您可以使用它来远程编辑文件。

还有一些适合 Windows 用户的东西：

如果您使用的是 Windows 并通过 CMD 终端使用 OpenSSH，则 OpenSSH 中的错误会阻止该 `-J` 选项工作。但是，您可以这样编写文件：

```
# ~/.ssh/config file for WINDOWS CMD users only

Host seis
  HostName seis.bris.ac.uk
  User USERNAME

Host lab
  HostName rd-mvb-linuxlab.bristol.ac.uk
  ProxyCommand ssh.exe -W %h:%p seis
  User USERNAME
```

`ssh lab` 这也应该对你有用。

使用不同的键

您在实验室中不需要此文件，但如果您曾经管理不同的系统和帐户，那么您可能会为每个系统和帐户使用不同的密钥文件。在这种情况下，命令行上的 `ssh` 允许您 `-i FILENAME` 选择私钥文件，并且在配置文件中您可以使用该行为特定主机选择文件 `IdentityFile FILENAME` 。默认情况下，`ssh` 将搜索 `.ssh` 名称为 的文件 `id_CIPHER` ，如果您使用显示详细调试信息的参数启动连接，您可以看到 `-vv` 。