第 5 章　会话层安全与 SSH

5.1　SSH 协议简介

在实际工作中，远程管理是一种极为常用的设备管理方式。在各类远程管理协议中，Telnet 是最为常用的协议之一。尽管 Telnet 协议可以通过用户名和密码对用户的身份进行认证，但这种协议无法在用户设备和被管理设备之间进行加密。通过前面几章的介绍，这种协议的弊病已经不言自明。

使用 SSH 协议可以实现安全的远程管理。它不仅可以让服务器对用户的身份进行认证，更可以对通信的信息进行加密、校验和压缩。图 5-1 所示为分别通过 Telnet 和 SSH 向被管理设备提供用户名和密码信息。

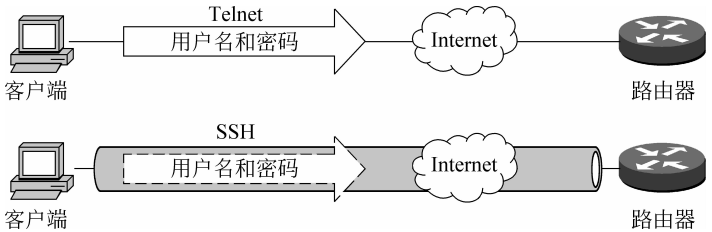

图 5-1　通过 Telnet 和 SSH 发送认证信息

SSH 最早的版本是由芬兰名校赫尔辛基工业大学（Helsingin Teknillinen korkeakoulu）的副博士 Tatu Ylönen 于 1995 年研发的，其目的旨在取代 rlogin、Telnet 和 rsh 等协议，对跨越公共网络的远程管理提供加密的认证。

SSH 和 SSL 一样都是分层协议，但 SSH 协议分为 3 层：传输层协议（定义在 RFC 4253 中）、用户认证协议（定义在 RFC 4252 中）和连接协议（定义在 RFC 4254 中），如图 5-2 所示。

图 5-2　SSH 协议的组成

SSH 协议的工作方式如图 5-3 所示。

通过图 5-3 可以清楚地看出 SSH 协议 3 层的作用与关系：

（1）传输层协议阶段：这一阶段的作用是为了在通信双方之间建立一条安全的加密通

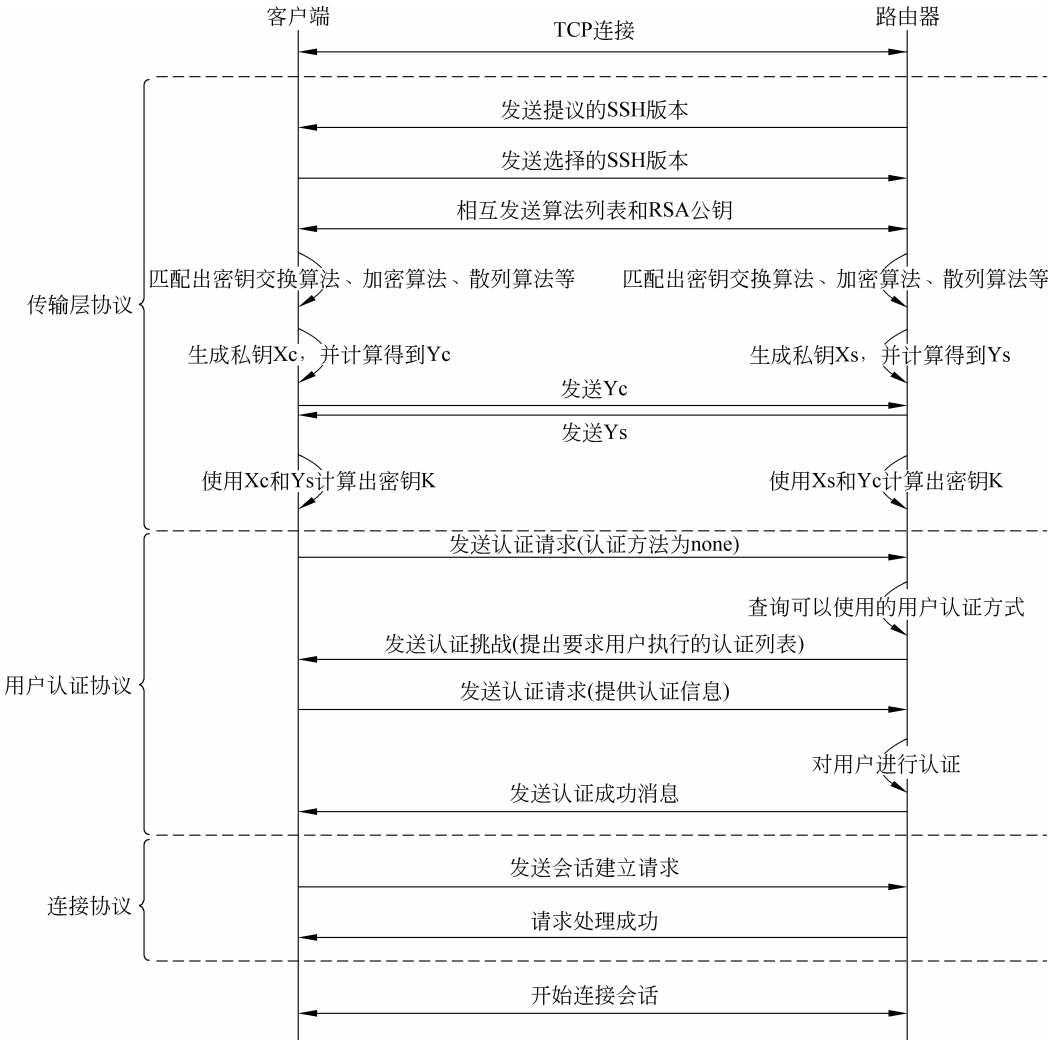

图 5-3　SSH 的协商过程

道。为用户数据提供私密性、完整性方面的保护。

- 版本协商：在服务器(即路由器)与客户端已经建立了 TCP 连接的基础上，由服务器向客户端发送自己支持的版本号。客户端会解析该数据包，如果客户端发现服务器端的协议版本号比自己的低，且自己可以支持服务器的低版本，那么使用服务器端的低版本协议号，否则使用自己的协议版本。(若服务器不支持客户端的版本，版本协商就会终止，连接也会断开。)
- 算法协商与密钥交换：接下来，服务器和客户端会分别向对方发送自己支持的算法列表(密钥交换算法、加密算法、散列算法等)和 RSA 公钥。在接收到对方发送过来的算法之后，双方会独立地对两边的算法列表进行匹配，选出双方在后续通信过程中使用的算法。然后，双方会通过密钥协商，计算出加密密钥。

(2) 用户认证协议阶段：从这一阶段开始，传输层协议阶段协商的算法就会开始对双方的消息提供保护，这一阶段的通信又可以分为 5 步：

- 第 1 步：客户端向服务器发送认证请求消息，其中携带的认证方式为 none。
- 第 2 步：客户端从配置的认证方式中找到需要这名用户完成的认证方式，并通过认证挑战的方式要求用户进行认证。
- 第 3 步：用户提供服务器所请求的信息。
- 第 4 步：服务器通过用户提供的信息对用户进行认证。
- 第 5 步：根据认证结果，服务器向客户端发送认证成功或认证失败消息。需要说明的是，如果服务器发现该用户还需要完成其他的认证，即使上述方式认证成功，服务器也会回复认证失败消息，并继续向客户端发送认证挑战，直至客户端完成认证列表中所有认证方式为止。

(3) 连接协议阶段：在用户完成认证后，客户端就可以向服务器发起服务器请求，要求服务器建立会话通道。服务器在接收到请求消息后，如果自己支持该类型的通道，那么它就会向客户端发送确认消息，会话通道就会建立起来。为了让连接协议可以更好地享用前两个协议阶段所建立的安全环境，连接协议可以在通信双方之间所建立的一条连接上复用出多条信道，分别处理不同的会话。

顺便一提，SSH 传输层协议具有认证功能，其作用是让客户端对服务器进行认证，这项功能是可选的，不要将其与用户认证协议混淆。

注意：通过上文的介绍，不难发现 SSH 协议不仅可以实现安全的远程访问，而且也可以为其他缺乏私密性和完整性保障的应用层会话提供保护。但这种做法在实际环境中的使用并不广泛，而且与此相关的内容也实在超出了本书的范畴。

5.2 使用 SSH 对远程登录用户进行认证

尽管 SSH 不仅能够提供安全的网络管理，也能够对一些其他的协议提供私密性和完全性保护，但使用 SSH 实现安全网络管理仍为 SSH 使用最为广泛的做法，因此，本实验只对通过 SSH 协议实现安全网络管理进行测试。

5.2.1 实验拓扑

图 5-4 为 SSH 实验的环境。

图 5-4 SSH 实验拓扑

如图 5-4 所示，在这个简单的环境中只有两台路由器。顾名思义，其中 SSHClient 为 SSH 客户端，而 SSHServer 则为 SSH 服务器，这两台设备使用串行接口直接相连。而本实验的目的即为 SSHClient 能够通过 SSH 协议，在完成了服务器本地认证后对 SSHServer 进行管理。

5.2.2 SSH 的配置

首先，需要在这两台设备上完成主机名和 IP 地址的初始化配置，如例 5-1 和例 5-2

所示。

例 5-1 SSHClient 上的基本配置

```
enable
configure terminal
!
hostname SSHClient
!
interface Serial 0
  ip address 10.1.1.10 255.255.255.0
  no shutdown
!
end
```

例 5-2 SSHServer 上的基本配置

```
enable
configure terminal
!
hostname SSHServer
!
interface Serial 1
  ip address 1.1.1.1 255.255.255.0
  no shutdown
!
end
```

相关的 SSH 配置全部集中在 SSH 服务器端,全部配置过程可以分为如下 5 步。

第 1 步:使用命令 ip domain name 配置域名。

第 2 步:使用命令 crypto key generate rsa 创建一个 RSA 密钥对。

第 3 步:使用命令 username username privilege privilege password password 创建本地的用户名和密码。

第 4 步:在 vty 线路下配置本地认证,即使用命令 login local 指定当有用户发起远程管理连接时,通过本地数据库对用户进行认证。

第 5 步:在 vty 线路下使用命令 transport input ssh 规定只能使用 SSH 进入 vty 线路。

全部的配置过程如例 5-3 所示。

例 5-3 SSH 的配置

```
SSHServer(config)# ip domain name cisco.com
SSHServer(config)# crypto key generate rsa
The name for the keys will be: SSHServer.cisco
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
```

```
%Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)SSHServer(config)# username admin privilege 15
password cisco
SSHServer (config)# line vty 0 15
SSHServer(config-line)# login local
SSHServer(config-line)# transport input ssh
```

根据需要,管理员也可以通过访问控制列表来限制用户能够从哪些地址发起 SSH 连接。具体的配置方法由两步组成。

第 1 步:配置一个 ACL。

第 2 步:在 vty 线路下调用这个 ACL。

具体的配置方式如例 5-4 所示。

例 5-4 用访问控制列表限制可以发起 SSH 访问的地址

```
SSHServer(config)# access-list 10 permit 1.1.1.10
SSHServer (config)# line vty 0 15
SSHServer (config-line)# access-class 10 in
```

此外,管理员也可以在全局配置模式下通过命令 ip ssh timeout seconds 来指定 SSH 服务器等待 SSH 客户端提供密码的时间;也可以通过命令 ip ssh authentication-retries *number* 来设置输入密码的次数,超出设定次数,服务器就会丢弃该连接,默认的次数为 3 次。

5.2.3 SSH 的测试

在完成 SSH 的配置后,需要在 SSHClient 上发起连接请求进行测试。

发起测试需要在 EXEC 模式下使用命令 ssh 来实现,具体的命令如例 5-5 所示。在本例中,使用 3DES 加密方式(-c 3des),且登录的用户名为例 5-3 中所示的 admin(-l admin),目的地址为 SSHServer 的 S1 接口地址,即 1.1.1.1。

例 5-5 在 SSHClient 上发起测试

```
SSHClient# ssh -c 3des -l admin 1.1.1.1
Password:
SSHServer#
```

在输入密码后,可以在 SSHServer 上通过命令 show ssh 来进行验证,如例 5-6 所示。

例 5-6 命令 show ssh 的输出信息

```
SSHServer# show ssh
Connection  Version  Mode  Encryption  Hmac       State            Username
     0        1.99   IN    3des-cbc    hmac-sha1  Session started  admin
     0        1.99   OUT   3des-cbc    hmac-sha1  Session started  admin
%No SSHv1 server connections running.
```

根据上述命令的输出信息,可以看到该 SSH 会话的版本为 1.99,加密方式为 3DES。

此外,命令 show ip ssh 可以查看到 SSH 的版本、密码响应超时时间和输入密码的次

数，如例 5-7 所示。

例 5-7 命令 show ip ssh 的输出信息

```
SSHServer# show ip ssh
SSH Enabled -version 1.99
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded):
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQDCVodExaFJEMEZXRBYwF6GXboUtzWcxJVBx0Pl8/ra
M3gL4UWBrv9waNSVIdzVCo0niD1DxxiEOhuauKc0MbZxUthdNumX5fMZh7huWPWugG
9ssZ7fbC0F+9r7
QomU6U9KGxrtJZSE6APreO8CW/3ufBY7VI8weiA1kEhqrb/qgw==
```

思 考 题

1. 既然 Telnet 协议也可以对远程管理用户进行身份认证，为什么还要使用 SSH 协议来保护远程管理访问？

2. SSH 协议旨在实现 CIA 三原则中的哪个(或哪几个)原则？