第一部分 概述

**场景一：** 设想这样一个场景，当你挤了很久的公交，又辗转了几班地铁，终于拖着疲惫的身躯回到家中的时候，突然间接到了领导的电话，公司服务器上有某个重要的服务需要立马被关闭，而公司的内部网络与外部网络之间有防火墙，禁止telnet远程直接链接，这时你会怎么办呢？（吐槽一下，运维好苦逼/(ㄒoㄒ)/~~）

**场景二：**  设想另外一种场景，在本地的一个服务器上运行着SMTP服务，而远程的一个主机需要访问这个SMTP服务，但是防火墙不允许不信任的外部访问，进入到局域网内部，而内部的主机却是可以正常的访问防火墙之外的网络，这种情况下，又应该如何去处理呢？

SSH端口转发概述

首先，我们来了解一下SSH端口转发的概念。当我们通过SSH协议建立的通信通道之后，SSH会自动给加密和解密客户端与服务器端之间的所有通信。同时，SSH 还提供了一个非常有用的功能，这就是端口转发。它能够将其他 TCP 端口的网络数据通过 SSH 链接来转发，并且自动提供了相应的加密及解密服务。这一过程有时也被叫做“隧道”（tunneling），这是因为 SSH 为其他 TCP 链接提供了一个安全的通道来进行传输而得名。例如，Telnet，SMTP，LDAP 这些 TCP 应用均能够从中得益，避免了用户名，密码以及隐私信息的明文传输。可以将SSH建立的通信链路理解为管道（隧道），不安全的TCP协议在隧道中传输数据，避免了明文的传输。而与此同时，如果您工作环境中的防火墙限制了一些网络端口的使用，但是允许 SSH 的连接，那么也是能够通过将 TCP 端口转发来使用 SSH 进行通讯。总的来说 SSH 端口转发能够提供两大功能：

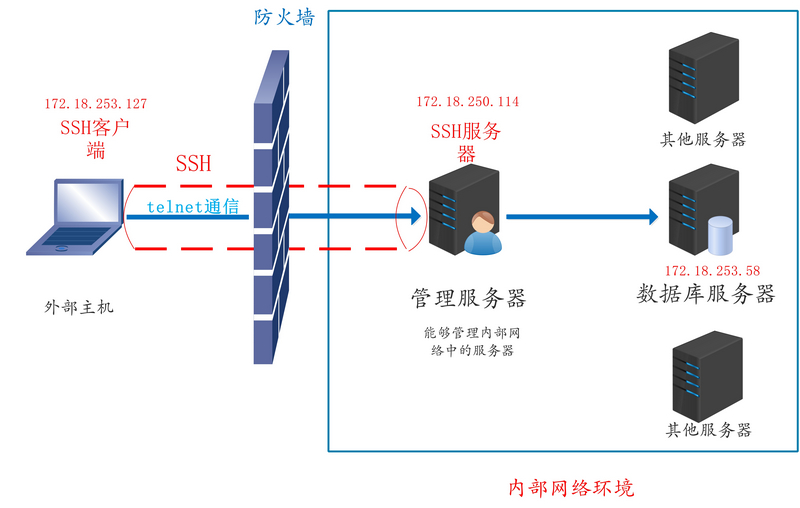
* 加密 SSH Client 端至 SSH Server 端之间的通讯数据。
* 突破防火墙的限制完成一些之前无法建立的 TCP 连接。

SSH 的端口转发有着多种类型，同时在不同的场景中有着多种不同的应用，单靠文字解释，是说不明白的，所以接下来，我们通过几个实验来详细的解释一下SSH的端口转发功能。

第二部分 本地转发与远程转发

本地转发实例

在场景一中，我们描述了这样一个场景。我们需要在远程使用telnet协议链接到某个被防火墙保护的内部网络中去，那么我们应该如何实现呢？  
答案无疑就是使用SSH的本地端口转发了。 首先看一下下面的网络拓扑结构。



本地端口转发

SSH 的本地端口转发的格式如下所示

ssh  -L local\_port:remote\_host:remote\_host\_port sshserver

本地端口转发有如下的常用选项。

* -f 后台启用
* -N 不打开远程shell，处于等待状态
* -g 启用网关功能

我们的实例实现如下所示。 这一个过程是在外部主机（172.18.253.127）上进行的操作

# 这里进行一下解释

# 172.18.253.58  是我们需要远程到的服务器 ，也就是图中的数据库服务器

# 172.18.250.114 是我们的转发服务器，同时也是SSH-Server

# 9527 是我们选择的本地监听端口

[root@localhost ~]# ssh -L 9527:172.18.253.58:23  -Nf  172.18.250.114

# 通过本地监听端口，使用telnet 协议就可以访问到 远程的数据库服务器了

[root@localhost ~]#telnet 127.0.0.1 9527

我们选择了本地主机的9527端口作为监听端口，是因为，非管理员用户，是没有权限，管理1-1023端口的，所以一般是选择1024-65535之间尚未被占用的端口来进行监听就可以了。 这样的话，整个通信的流程，大体上就是下面的样子

* 我们在外部主机(127.0.0.1) 上的telnet 应用在使用的时候将数据发送到本机的9527端口上
* 外部主机(127.0.0.1)上的SSH-Client会将9527端口收集到的所有的数据加密发送到SSH-Server(也就是图中的管理服务器 172.18.250.114)上
* SSH-Server(管理服务器 172.18.250.114) 会将解密后的数据，通过局域网发送到目标服务器（172.18.253.58）上
* 目标服务器 上的数据再原路返回，就完成了整个流程(图中省略了数据返回这一流程，避免引起理解上的歧义)。

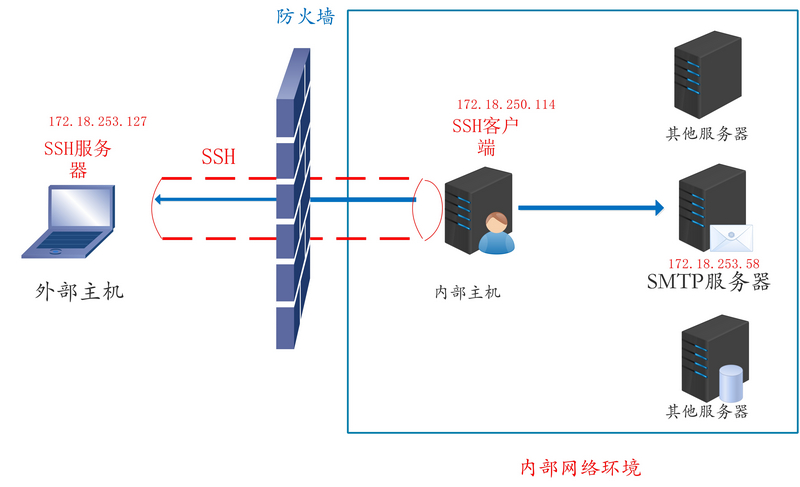
这样，我们就解决了**场景一**中所面临的困境。而且，外部主机并没有直接链接到数据库服务器上，而是监听了一个本地端口，由SSH完成了剩下的所有的事情，比如加密，通讯，解密，转发等等。

同时有下面的注意点

* SSH 端口转发是通过 SSH 连接建立起来的，我们必须保持这个 SSH 连接以使端口转发保持生效。一旦关闭了此连接，相应的端口转发也会随之关闭。
* 我们只能在建立 SSH 连接的同时创建端口转发，而不能给一个已经存在的 SSH 连接增加端口转发。
* 在主流的SSH实现中，谁建立的端口监听，谁就可以通过端口进行通讯，其他主机不能通过已经建立的端口进行链接。
* 为什么最后要使用，127.0.0.1 来进行远程链接呢？我们可以形象地理解为，一面墙上有一个通道，外面地人只能通过这个隧道的入口，与隧道另一端之外的人进行通信，那么就必须先进入这个隧道入口，而隧道入口就是端口9527 ，那我们能够打开的门就是摆在我们面前(127.0.0.1)的门（9527）了。
* 这个实验的前提是，通过放火墙不能直接远程到我们需要链接的主机，但是防火墙却能够允许，SSH协议进入到局域网内部,SSH协议是安全加密通信协议。

远程转发实例

回到场景二中，如果我们使用SSH协议也无法进入到局域网内部怎么办？我们想要访问，某个局域网内部的SMTP服务器，但是防火墙隔离了我们的所有请求。而局域网内部的主机却可以通过防火墙访问到外部的主机，此时我们应该如何是好呢？ 答案应该就是使用SSH的远程端口转发了。首先看一下，远程端口转发的网络拓扑。



远程端口转发

SSH 的远程端口转发的格式如下所示

ssh  -R sshserver\_port:remote\_host:remotehost\_port sshserver

9527 端口将在远程主机上打开

我们的实例实现如下所示。 这一个过程是在内部主机（172.18.250.114）上进行的操作

# 这里进行一下解释

# 172.18.253.58  是我们的SMTP 服务器

# 172.18.253.127 是防火墙隔离之外的主机，在远程端口转发的案例中，承担了SSHSERVER的角色

# 9527 是我们选择的远程的端口，也就是说，当隧道建立成功之后，会在172.18.253.127的主机上开启9527端口，并进行通信

[root@centos6 ~]$ssh -R 9527:172.18.253.58:25 -Nf 172.18.253.127

此时我们切换到远程主机（172.18.253.127） 使用ss -nlt 命令查看一下当前的端口，就会发现开启了9527端口。 如果此时我们使用telnet 命令连接一下本地的端口 ，就能够像 SMTP服务器发起SMTP请求了。

# 通过本地监听端口，使用telnet 协议就可以访问到 远程的SMTP服务器了

[root@localhost ~]#telnet 127.0.0.1 9527

Trying 127.0.0.1...

Connected to 127.0.0.1.

Escape character is '^]'.

220 localhost.localdomain ESMTP Postfix

helo localdomain    #开始进行邮件输入了250 localhost.localdomain

mail from:mage@magedu.com

250 2.1.0 Ok

rcpt to:root

250 2.1.5 Ok

data

354 End data with <CR><LF>.<CR><LF>

subject:hello,ssh

ssh remote link

.

250 2.0.0 Ok: queued as A5B6C2003944

quit

221 2.0.0 Bye

Connection closed by foreign host.

和本地端口转发相比，这次的图中，SSH的server主和Client相互换了换了位置。但是数据流还是类似的，还是从外部主机发送到内部网络上的服务器。 数据的流程变成了下面的这个样子

* 由SSH Client 也就是内部主机（172.18.250.114）发起SSH请求，建立SSH链接。
* 此时SSH Server 也就是外部主机(172.18.253.127) 上就开启了9527端口，此时局域网内部就建立了一个与远程主机之间通信的管道。
* 远程主机（172.18.253.127）通过telnet 链接就能够建立与SMTP服务器（172.18.253.58）的链接，并访问其SMTP服务，为了避免混淆，图中并没有画出访问SMTP的这条线。

**那么到底，本地端口转发，和远程端口转发的区别到底在什么地方呢？**

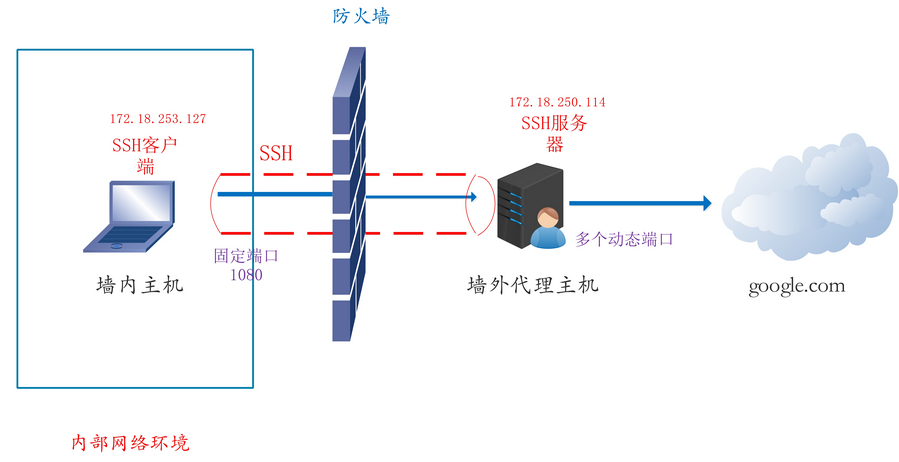
本地转发与远程转发的对比分析

我们还是可以结合上面的两个场景的实际用例来进行分析。 首先，我们知道，SSH的链接是有方向的，都是从Client 到Server，而我们的实例中有一个很明显的不同就是，9527端口到底在哪个主机上打开了。因为我们都是通过9527端口发送的数据，所以我们应用（telnet）的方向，都是从Client端发送到Server端。如果我们通过9527端口发送的数据的方向与SSH Client端发送数据的方向一致的话，就是本地端口转发，否则就是远程端口转发。

第三部分 其他类型的转发

动态转发实例

设想这样一种场景。我们在防火墙内部想要访问放火墙外面的网站，但是防火墙给我们开放了很少的端口。那么我们如何访问墙外的美好世界呢？ 先来看一下下面的拓扑结构



如何解决上面场景中遇到的问题呢？答案是使用动态端口转发。 为什么要使用动态端口转发呢？我们知道防火墙外面的主机能够访问到互联网上的多个服务，若干个服务可能需要不同的端口号来进行提供，而防火墙给我们提供的端口有限，所以我们不能直接墙外主机上的服务。另外，逐个建立本地端口转发也是意见非常麻烦的事情。

SSH 的动态端口转发的格式如下所示

# ssh server 指的就是我们在放火墙之外的代理服务器

# 1080 也可以是其他可用端口

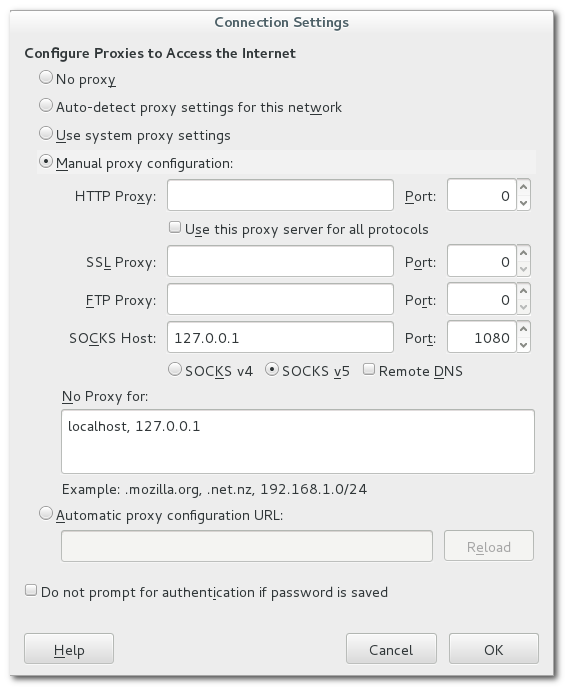
ssh -D 1080 root@sshserver

我们为网内的主机设置一下动态代理

# 执行这条命令的是172.18.253.127

[root@localhost ~]#ssh -D 1080 root@172.18.250.114

这时通过动态转发，可以将在本地主机发起的请求，转发到远程主机，而由远程主机去真正地发起请求。而在本地发起的请求，需要由Socket代理(Socket Proxy)转发到SSH绑定的1080端口。 我们以火狐浏览器为例，配置本地的网络访问代理。找到设置=>高级=>网络=>代理，然后设置成如下的内容。



这样的话，Firefox浏览器发起的请求都会转发到1080端口，然后通过SSH转发到真正地请求地址。动态代理也就设置成功了。

X协议转发实例

所有图形化应用程序都是X客户程序。

* 能够通过tcp/ip连接远程X服务器。
* 数据没有加密机，但是它通过ssh连接隧道安全进行

SSH X协议转发的格式如下所示

# 所有的图形话应用程序都是X客户程序  例如gedit

ssh -X user@remotehost    X\_APP

**通过上面的介绍，我们就应该能够简单的了解了如何使用SSH转发机制来解决我们日常生产中的问题。比方说，解决防火墙或者网络应用带来的一些限制，以及在不是很安全的场景下如何加密自己的上网信息，从而达到自己安全上网的目的。**