第4章　网络地址转换

本章内容

- 什么是NAT，它是如何工作的。
- 通过示例讲述如何实现NAT。
- NAT如何应用于网络安全。
- 什么时候恰当地使用NAT。

4.1　概述

这章讲述的是网络地址转换（NAT）。从最简单的方式来看，NAT就是要通过某些设备来转换网络层（第三层）地址，这些设备包括路由器、防火墙等。理论上讲，其他的第三层协议，如AppleTalk或IPX协议，或其他层协议（如第二层协议）都能被转换。实际上，目前一般仅用于第三层的IP地址转换。由于这是一本有关TCP/IP的书籍，所以本章仅讨论IP问题。

我们将通过示例来说明仅转换第三层的地址是不够的，运输层（第四层）及更高层上的信息也可能被影响。所以我们这里的讨论也包括TCP、UDP以及应用层（第七层）协议。我们不仅要讨论什么是NAT、NAT是如何工作的，还要讨论它的问题和缺陷。

尽管这一章不是讲述网络安全的，然而NAT的问题时常与一些安全应用有很密切的关系。在某些情况下，NAT的一些特殊类型对安全应用的非常有用，许多安全包中都包含有商业NAT的实现。这也就是说，本章我们将讲述一些与NAT相关的安全问题，尽管NAT本身并不需要安全技术。

4.2　在路由器或防火墙的后面

在早期的防火墙解决方案中，经常会使用到NAT。这些早期的防火墙几乎都是基于代理的。一个很好的示例是FireWallToolKit(FWTK)软件。代理存在于防火墙内部，它负责为客户提供一些信息，例如，Web页面。客户计算机向代理请求一个特定的Web页面（要给出URL），并等待应答。此时，代理将会找到Web页，并将它返回给客户。

这个代理具体是什么？首先，代理的管理员经常要编制一些内容列表，这些内容是不允许客户访问的。例如，如果在一个公司中有一个Web代理，那么代理管理员也许会禁止公司内部计算机对www.playboy.com的访问。其次，代理还能够完成一些高速缓存和其他优化工作。如果每天有50个人访问www.syngress.com，代理就能够将这个Web页的拷贝下来。当一个客户请求这个Web页面时，代理所做的所有工作就是检查这个页面是否发生了一些变化。如果没有发

生变化，则代理只传送它已存储的拷贝，这样客户就能够更快地看到这些页面。

一般来说，对于这种类型的代理配置，主要是阻止客户直接从 Internet上查看某些Web页面。如果它们想要查看某些页面的话，则必须使用这个代理。这个工作通常是通过路由器上的包过滤功能来实现的。简单的来说，路由器被配置成仅允许代理访问在 Internet上的Web页面，而不允许其他机器访问Internet上的Web页面。

这种设计的结果就是内部的客户只能与代理进行通信，而不能同在 Internet上的其他主机进行通信。代理需要接收内部客户的请求，然后完成这个请求。这也就意味着在 Internet上的其他主机也不能直接同内部主机进行通信，甚至不能直接应答。所以，防火墙管理员通过配置路由器或防火墙能够隔断内部和外部机器间的通信。这种方法将强制所有的通信都通过代理来完成。现在，如果配置正确的话，能够与外部通信的唯一机器是代理，这会大大减少受到外部直接攻击的机器数量。代理的管理员应尽可能谨慎，以保证代理机器尽可能安全。图 4-1是有关这方面的逻辑示意图。

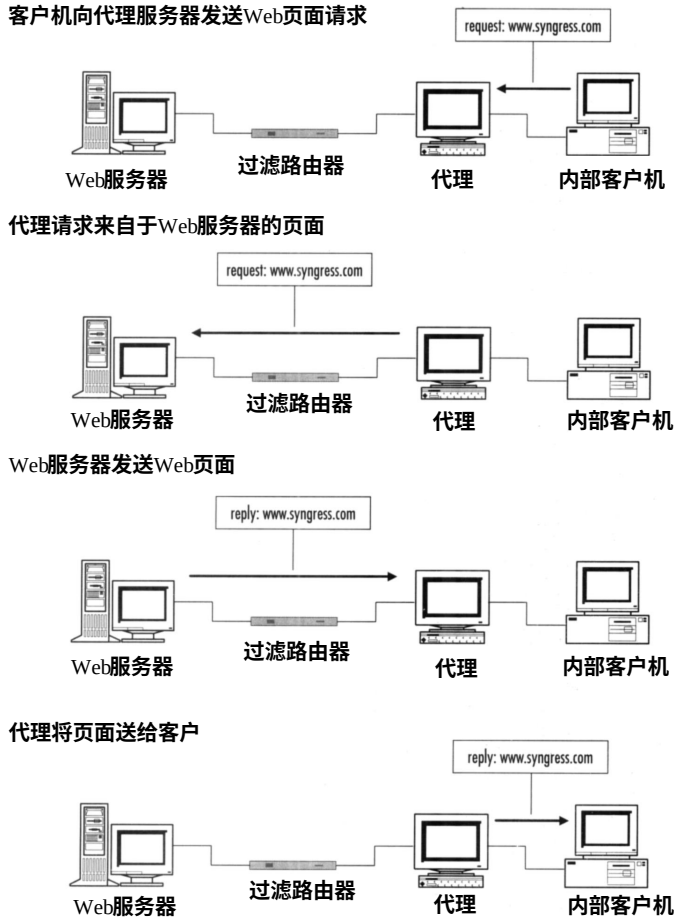

图4-1 通过代理检索Web页面

为了达到我们讨论的目的，这个过程已经被大大地简化。但要注意下面的原则：内部和外部的清楚划分以及它们之间的交接点。它们之间的交接点有时也被叫做阻塞点。在我们的图中，阻塞点就是代理和过滤路由器的组合。

这是一种最简单的防火墙结构。当你要设计一个真正的防火墙时，还需要了解许多本章以外问题，例如：

- 代理软件是否支持所有所需的协议。
- 如何在路由器上配置分组过滤。
- 在客户端的Web浏览器软件如何与代理进行通信。
- 代理如何知道哪个机器在内部，哪个机器在外部。

本章讨论的重点不是代理防火墙的结构，而是讨论它的影响。我们已经知道，从这个网络到Internet上的所有流量实际上都来自于代理。这也就是说， Internet仅能看到代理服务器的IP地址。我们已经知道， Internet不能到达内部的客户机器上。

就Internet而言，这个位置仅需要一个IP地址，它就是代理的IP地址。

回想一下第3章，目前的IP地址空间是很少的。像私有地址空间那样，一些 IP地址空间已留做它用。RFC 1918文档列出了这些地址空间，具体内容请查看网址：

http://www.cis.ohio-state.edu/htbin/rfc/rfc1918.html

还有许多站点也有这个文档。

如果恰巧你已经读过 RFC，你会看到RFC1627和1597（RFC 1918的老版本）已经过时。RFC 1627反对使用私有IP地址空间。很明显，私有地址空间的使用也就意味着 RFC1627已经不再使用。其他的RFC文档可以从前面的URL上获得（这些连接在Web页的顶部）。

下面的这段文字来自于RFC1918，它定义了私有地址空间，以及在什么时候使用它们。

“出于安全上的原因，许多企业使用应用层网关将内部网络连接到 Internet上。内部网络通常不能直接存取Internet，而Internet也只能看到一个或多个网关。在这种情况下，内部网络可以使用不唯一的IP网络号。”

RFC认为许多公司都已经使用了应用层网关（代理），这也是使用私有地址的部分原因。所以，如果内部的机器不同其他机器进行直接通信，则留出一部分地址作为内部使用是非常有意义的。

RFC还认为，希望使用代理的公司应该使用从 Internet服务提供商（ISP）那里得到地址空间。在近些年，大部分地址空间都分配给了 ISP，而没有直接给公司。之所以这样做的最大原因就是要保持Internet核心路由器中的路由表尽量小。如果一个地址块分配给了 ISP，尽管块中的不同地址范围分配给了不同的公司，其他的 ISP只需对这个单一地址块进行路由就可以了，而不需要为地址块中的每个空间分配一个入口点。按现在的规则，你最好从 ISP那里得到一个固定的地址空间。有关ISP获得和分配地址的信息，请阅读第6章。

如果你运行一个代理结构，从 ISP得到地址是相当容易的，并且需要的数量也相对较少。对于这种结构，你可以在你的网络内部自由使用 RFC1918文档所提供的地址，并且内部客户机器还能实现对Internet的访问。

今天，这种类型的结构用得非常广泛。许多公司，特别是大公司都有一些能够直接同Internet进行通信的防火墙或代理设备。尽管公司已经使用 Internet很长时间，并且有自己的地址空间，但出于对安全的考虑，通常还使用这种类型的结构。

现在，我们对代理已经有了一些认识，那么它与 NAT又有什么关系呢？实际上它们之间关系不大—代理不是 NAT。在本章的最后，我们将做详细的解释。由于代理是 NAT存在的重要原因之一，所以对它的讨论显得也很重要。

什么是NAT

NAT的设计思想与代理的一个优点非常相类似：隐藏你的内部地址。我们已经谈到过想隐藏地址的一般原因 ----内部客户机对 Internet的访问。从高层次的观点来看，它们的最终结果都是一样的。 Internet应该能够看到由 ISP分配的一个有效的 Internet地址（公共地址），而内部机器全部使用私有地址。

如果你正在使用 RFC1918中的地址，则至少有一个原因使你想使用 NAT：如果你的公司同其他公司合并，该怎么办？一般情况下，两个公司都希望彼此连接到内部网，以实现商业上的信息交流。然而，如果两个公司在以前都使用了相同的 RFC1918地址空间，则将产生地址冲突。此时，你不得不进行地址的重新分配。现在有一种更简洁的办法，那就是使用 NAT实现两个公司间的地址转换，以解决冲突。我们将在后面讨论这个例子。

为了理解NAT与代理的区别，我们首先要仔细看一看 NAT是如何工作的。

4.3 NAT如何工作

NAT的工作就是修改独立的分组。它至少要修改第三层的头，以便使源地址、目的地址或两者有一个新地址。我们也将会看到一个修改第四层头结构以及修改第七层的数据口的例子。

我们不仅将会看到，在地址转换过程中一些小的变化将会产生一系列行为和特征上的变化，而且也能看到，对于某些协议在实现 NAT时，要比简单的在第三层做地址转换做更多的工作，甚至有些协议不能够实现 NAT。

NAT的功能通常是由路由器或防火墙来完成的。从原理上来说，一个第 2层网桥设备也能实现第三层转换，至少市场上的防火墙新产品能够做到这一点。然而，大多数 NAT设备和包含NAT功能的软件都是根据一般 IP的路由来传递分组。大部分 NAT设备都支持IP路由功能。

4.3.1 静态网络地址转换

我们从最简单的静态或叫 1对1转换开始。这是一种最直接的方式：简单地说，在静态 NAT

中，一个特定的IP地址在某一方向上被转换成其他地址。同样，反方向也要转换。对于向外的分组，通常要对源地址进行转换，图 4-2将有助于清楚地了解这个过程。在图 4-2中，箭头描述了分组流动的方向（被路由的地方），S表示源地址，D表示目的地址。

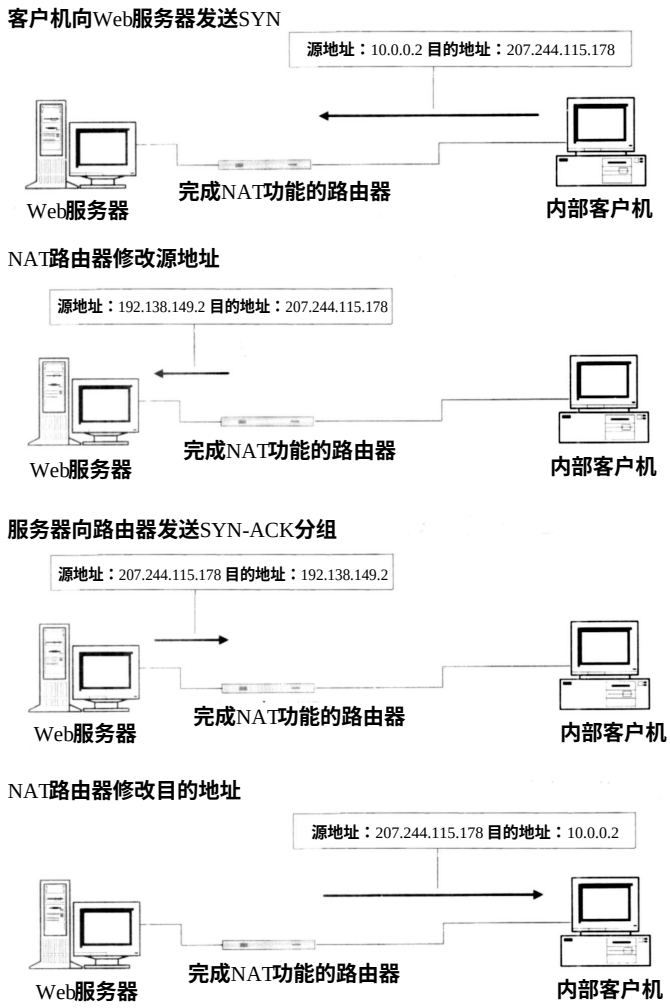

图4-2 在TCP握手中头两个分组的静态NAT

4.3.2 静态NAT是如何工作的

让我们假设这是一个头脑简单的 NAT，这也就是说，它所做的一切就是在适当的时候修改源或目的地址。此时， NAT路由器要完成哪些工作呢？首先，它要知道分组向哪个方向传送，这与NAT配置直接相关。注意，在示例中，路由在一个方向转换源地址，在相反方向转换目的地址。根据网络接口上获得的标记“ to ”或“ from ”的不同，路由器完成不同的工作。下面的

配置示例将会使这个问题更加清楚。路由器也许还要降低 TTL电平，并且还做一些必要的校验，但无论如何都要完成上述工作。

这个示例是通用的。路由器并不需要知道前面的分组情况，它只修改当前分组。在当前的分组和配置中包含有修改分组的所有信息。注意，这种类型的 NAT没有安全特征----除了进行地址转换外，所有的分组都全部通过。状态信息对以后的 NAT示例以及防火墙都是非常重要的。请记住，后面将会讨论这些问题。

这种类型的NAT非常简单，容易理解，但不是非常实用。考虑到我们的研究目标，目前使用一些IP地址来表示一组内部计算机。我们的示例是一对一的，没有节省 IP地址。每一个内部IP地址都要与一个外部地址相匹配，这样没有节省 IP地址，难道它没有什么用吗？不是，在许多情况下要使用一对一的 IP地址映射。

一种情况是你有一个拥有内部 IP地址的内部计算机，现在由于某种原因想接入 Internet。一种方法是不改变内部计算机的任意信息，只需要定义一个静态转换就可以了。我们的示例就是这样。如果这样做的话，你只需要发布被转换的 IP地址就可以了（也许可以使用 DNS来对它进行命名）。

让我们考虑另外一个示例，除了第一个分组被转换的地址不是源地址，而是目的地址外，它与图4-2非常相似。什么情况下会转换目的地址而不转换源地址呢？至少有一种类型的服务器是这样，它就是 DNS服务器。它是通过 IP地址进行索引的。想象一下这种情况， DNS服务器目前不能工作，也可能是临时的，此时内部客户机在做 DNS请求时，NAT会自动找到一个新的DNS，而不需要在客户机上进行任何配置。当原始的 DNS服务器恢复正常时， NAT再把它恢复回去。

4.3.3 双NAT

我们想讨论的最后一个静态 NAT示例被叫做“双 NAT”。简单地来说，就是同时转换一个分组的源地址和目的地址。许多支持 NAT的产品并不支持这种配置。除非你有两个 NAT。

在什么环境下使用双 NAT呢？一种可能是将前面两个例子进行组合：你的内部计算机使用的是私有地址，并且在不需要进行重新配置的情况下可连接到不同的 DNS服务器上。这个示例尽管不太实际，但作为示例是很好的。

回忆一下有关使用私有 IP地址可能产生冲突问题。当连接的网络也使用相同的私有地址时，冲突就发生。双 NAT做为一个临时手段，能够解决这个问题。

请看这种情况，当你需要将你的网络连接到其他公司时，你突然发现两个网络都使用了 C类地址192.168.1。除非被重新编址，否则这两个网络不能够进行通信。这种情况的出现不是不可能的。一些防火墙 /NAT产品的缺省情况都使用这个地址。

很明显，你需要两个能够实现 NAT的路由器----每个网络都连接一个相同的路由器。在我们的示例中，我们将集中在两个机器上，每个网络一台，并有相同的 IP地址（见图4-3）。

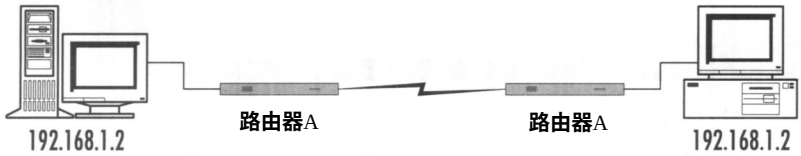

图4-3 有RFC1918地址冲突的两个网络

在这个示例中，如果两个路由器间连接的 IP地址没有其他冲突的话，这两个地址就显得不是很重要。

这个技巧就是使每个机器都相信，其他的机器有不同的 IP地址。我们让左边的机器认为右边机器上的IP地址是192.168.2.2，右边的机器认为左边的机器是 192.168.3.2。

这还是一个静态 NAT：每个机器都采用一对一方式，映射到其他 IP地址上。然而，在这个例子中，信息要通过两个路由器，所以它们要被转换两次。第一个路由器将转换分组的源地址，而第二个路由器将转换分组的目的地址，这就是双 NAT。

图4-4的示例说明一个左边的机器将一个分组发送给右边的机器。

现在假设左边的机器同另一个 IP地址为192.168.2.12机器进行通信。它将分组送到本地路由器上进行转换，与正常的没有区别。在这个路由点上，路由器 A将转换在分组中的源地址，以便隐藏如下事实：分组实际来自于 192.1.68.1网络（见图4-5）。

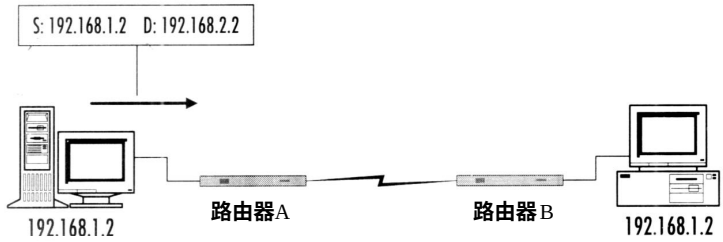

图4-4 源地址是192.168.1.2，目的地址是192.168.2.2

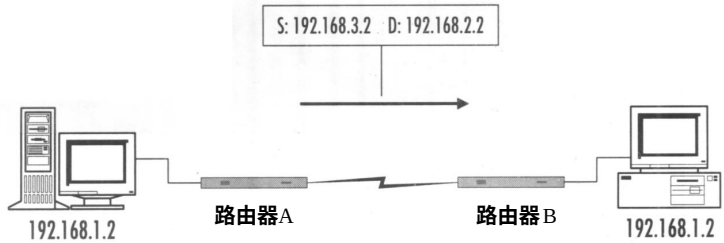

图4-5 源地址是192.168.1.2，目的地址还是192.168.2.2

在这个路由器上，目的地址仍保持 192.168.2.2。路由器 A使用正常的路由表来确定192.168.2网络在什么地方，然后转发分组。在这种情况下，分组被转发到路由器 B。路由器B将进行下一步的转换，将目的地址从 192.168.2.2转换成192.168.1.2（见图4-6）。

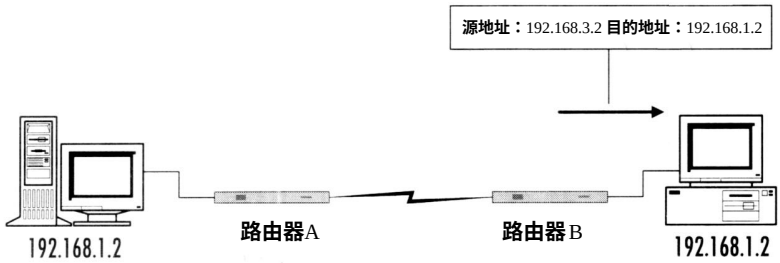

图4-6 源地址是192.168.3.2，目的地址改为192.168.1.2。

现在，右边的机器接到了分组，并且它相信接到的分组来自于 192.168.3.2。分组从右边机器到左边机器的过程与这个过程相似，但转换过程相反。

在这种方法中，两个具有相同地址，且不能够进行通信的机器却实现了通信功能。一般来说，如果在实际中使用这种结构，这就需要对 DNS进行巧妙的设置。当在 DNS中配置完左边机器后，右边机器上的名字将能够被解析到 192.168.3这个地址中。

4.3.4 静态NAT的问题

到目前为止，我们忽略了 NAT带来的问题，这些问题也是比较重要的。最基本的问题是：并不是所有网络的地址信息全部放在网络地址头中（ IP层）。由于各种不同原因，造成一些协议的地址信息可能被放到分组的数据部分。我们看下面一些例子。

给NAT带来最大问题的协议是文件传送协议（ File Transfer Protocol：FTP）。然而，由于FTP非常通用，大部分 NAT能够正确解决这个问题。

FTP协议有什么问题呢？首先，它传送的 IP地址以ASCII方式存在于数据流中。其次，它通过传送这些地址来通知其他机器将按相反的顺序监听源 IP地址和端口号。在缺省方式下，当一个FTP客户想接收一个文件时，它一直监听由操作系统分配的端口号，并通知服务器它的端口号和它的IP地址。服务器然后连接客户并且传送文件。我们将在后面看到当要考虑安全或其他类型NAT时，问题会变的更糟糕。

这就意味着 NAT软件要注意捕捉被发送的 IP地址，并且能够修改它们。 FTP也引出了一个状态问题。对于 NAT软件设计者来说，非常不幸的是 IP地址信息经常被分割到多个分组上，这就意味着NAT软件要从当前分组开始一直跟踪到最后一分组。这通常被叫做维护状态信息，大部分NAT设备使用状态表来维护这类信息。

图4-7就包含着一个分组捕获的问题。

```
IP：---------IP header -----------
    IP：
    IP：Version=4,header length=20bytes
    IP：Type of service=00
    IP：    000. ....  =routine
    IP：    ...0 ....  =normal delay
```

```
    IP:     .... 0...  =normal throughput
    IP:     .... .0..  =normal reliability
    IP:Total length   =66 bytes
    IP:Identification =3437
    IP:Flags=4x
    IP:     .1.. ....  =don't fragment
    IP:     ..0. ....  =last fragment
    IP:Fragment offset=0 bytes
    IP:Time to live   =128 seconds/hops
    IP:Protocol=6(TCP)
    IP:Header checksum=410F(correct)
    IP:Source address     =[208.25.87.11]
    IP:Destination address=[130.212.2.65]
    IP:No options
    IP:
TCP:---------TCP header -----------
    TCP:
    TCP:Source port=4585
    TCP:Destination port         =21(FTP)
    TCP:Sequence number          =353975087
    TCP:Next expected Seq number=353975113
    TCP:Acknowledgment number    =1947234980
    TCP:Data offset              =20 bytes
    TCP:Flags=18
    TCP:            ..0.....    =(No urgent pointer)
    TCP:            ...1....    =Acknowledgment
    TCP:            .... 1...    =Push
    TCP:            .... .0..    =(No reset)
    TCP:            .... ..0.    =(No SYN)
    TCP:            .... ...0    =(No FIN)
    TCP:Window=8030
    TCP:Checksum=1377(correct)
    TCP:No TCP options
    TCP:[26 Bytes of data]
    TCP:
FTP:---------File Transfer Data Protocol -----------
    FTP:
    FTP:Line 1:PORT 208,25,87,11,17,234
    FTP:
```

图4-7 包含FTP PORT命令的分组

图4-7的内容来自于FTP对话中间的一个分组，里面包含着端口命令。再深一步说，FTP基本上是一个文本协议，然后在上面加入了二进制传递。你在图的底部看到的命令PORT 208.25.87.11.17.234，它的作用是客户通知服务器为了能够接收数据，它使用哪些端口进行监听。目前我已经连接到服务器上，我的客户向服务器发送一个地址号和一个端口号，以便服务器能够向客户正确发送标题为welcome的内容。

让我们看一看这个命令。PORT部分是很明显的：它告诉服务器可连接到客户的哪个端口上。其中前4个数208,25,87,11是客户的IP地址-在图顶端源地址（source address）条目中，地址也是208.25.87.11。随后的两个数是已被划分成两个字节的端口号。注意，目前的源端口号为4584。在这种情况下的客户通常使用带 Windows 98的计算机，象大多数操作系统那样，Windows 软件也是按顺序分配端口的。使用下面方法可以将 17，234这两个数合并成一个数：左边的第一个数乘以256，然后再加上第二个数，即17×256+234=4586。此时客户告诉服务器要连接的地址为208.25.87.11,端口号为4586。

每件事情工作都象我们所希望的那样，标题将被正确地显示在 FTP客户端。但是，如果使用NAT的话，它将不得不识别这个PORT命令，并且修改内部分组中的IP地址。在这个示例中，同一个分组包含着所有的域（经常是这样）。然而在有些情况下，这些域将被拆分到多个分组中，这就需要NAT软件能够处理这种可能性。

如果NAT软件能够正确地修改端口命令，这就意味着它工作得很好。头被修改后， PORT命令也要相应地被修改。现在FTP能够通过静态NAT正常工作了。

这仅仅是需要特殊处理的一个协议——还有许多这样的协议。为了满足客户要求，实际的NAT实现要解决这些问题。一般情况下， NAT的销售商都会提供一张协议列表，以便说明哪些协议能够正常工作，哪些协议不能正常工作。最根本的问题在于协议本身。如果协议要传送的地址信息和端口信息存在于分组的数据部分，则当 IP的头需要改变时，为了匹配，数据部分也要改变。如果不做改变的话，协议将不能正常地工作。

目前还有一些协议在使用静态 NAT时会产生一些问题。一些协议能够对 IP头进行检测，当发现头被修改时，协议将拒绝工作。通常这些协议都是带有安全性的协议。最典型的例子是IPSec的确认头（Authenticate Header:AH）协议。尽管不需要了解IPSec的细节，但有一点要注意，你要通信的IP地址一定要是对方声明的IP地址。当两个IP地址使用IPSec AH进行通信时，它们要使用公共密钥来校验信息的类型。当一个设备将上述内容放入到一个分组中时，分组中将包含一个大的数字，它的功能是描述在分组中的所有信息。当在另一端的设备看到这个分组时，将进行同样的处理。并且要判断分组是否已经被修改。如果发现一些修改，则将这个分组作为无效分组放弃掉。

IPSec AH会认为NAT对分组进行了修改（即对头的未授权修改），并将这个分组作为无效分组丢掉。象这种协议并不是很多，由于它们通常都非常复杂，所以经常需要网络管理员和防火墙管理员对它们进行配置。管理员应能够意识到这这个问题，并能够处理它。但要注意，一些ISP还在他们的网络中使用NAT。由于虚拟专用网（Virtual Private Network:VPN）产品使用IPSec协议，所以这些产品既不能工作在使用 NAT的ISP网络中，也不能工作在使用任何类型防火墙的网络中。

4.3.5 示例配置

在本章，示例的配置将采用Cisco的IOS、Windows NT 2000和Linux。特别指出的是我们

将使用Cisco IOS 11.3或更高版本（在主路由器上），以及Red Hat Linux 6.0。但要注意，一些其他的Cisco设备，如77x ISDN路由器能够很好的支持NAT，但在它们的软件中使用的是不同的编址策略。我们之所以使用 Windows NT 2000是因为这是第一个内置 NAT能力的Windows NT版本。在这本书的写作过程中， NT 2000还处于Beta版测试阶段，希望在最终正式版中有这个特征，但也不能排除这个特征将被删去或修改。我们正在 Linux上使用的软件包被叫做 IP Masquerade，它来自于最新发布的 Linux版本。本章最后的“索引和资源”部分将提供一些URL，能通过这些URL获得包含NAT信息的文档、包含NAT特征的Cisco IOS版本信息；如果你的Linux版本中没有IP Masquerade软件，则可通过提供的URL获得这个软件。这章首先假设这些相应的软件已经被安装，并且你对这些操作系统已经有了基本的了解。

1. Windows NT 2000

在Windows NT 2000中的这个特征被叫做 Internet连接共享（Internet Connection Sharing:ICS）(Windows 98第二版也包含ICS)。ICS的目的是想通过拨号用户为连接到 LAN上的其他计算机提供访问 Internet的能力。这个软件做得很好，也相当专业，但不够灵活。对外的接口必须是拨号连接，这也就是说，如果你对 Internet的访问是通过LAN连接的（例如，使用的是Cable调制解调器或DSL设置），你就不能够使用ICS了。为了能满足在LAN上的内部机器要求，通过NT 2000中的工具箱可将LAN接口配置成192.168.0.1，并将它加入到DHCP服务器和DNS代理中。如果这些服务已经存在的话，对 LAN接口的配置很容易引起冲突，一定要小心。我们假设NT 2000已被正确地安装， LAN接口已经被正确配置， Internet拨号连接已经被正确地定义。我们现在打开网络的控制面板（如图 4-8）。

在图4-8中，我们能够看到LAN连接和Internet拨号连接。Internet拨号连接是灰色的，这说明此时它不能工作。

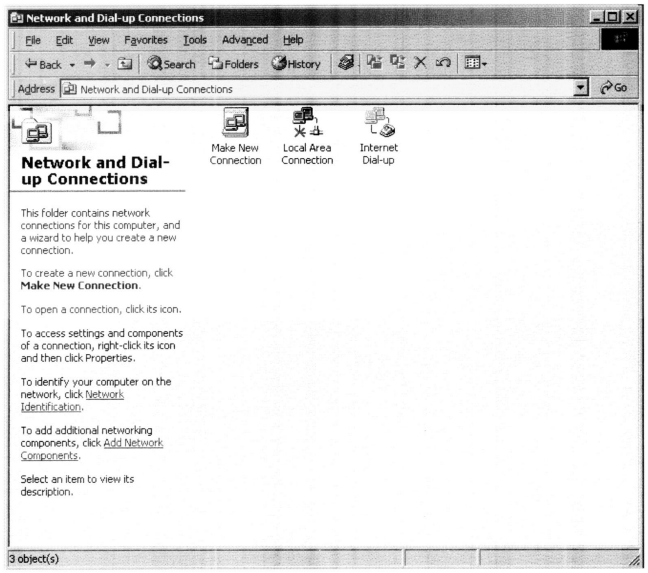

图4-8 Windows 2000 网络连接窗口

为了配置ICS，右击Internet拨号连接图标，并选择属性（Properties），当属性窗口出现后，单击Internet连接共享（Internet Connection Sharing）菜单条（见图4-9）。

选中“允许Internet连接共享”（Enable Internet Connection Sharing）选择框以允许ICS进行正常的工作。另外，当内部计算机想访问 Internet时，你要将NT 2000计算机配置成具有自动拨入Internet的功能。选中这个选项的同时也激活了 DHCP服务器。但一定要注意，在选中前，DHCP服务器没有工作。

这个简短的配置示例仅能够说明在拨号后，内部计算机能够访问 Internet。然而，由于我们正在讨论的内容是静态NAT，所以我们还要更深一步了解ICS的工作过程。严格的来说， ICS不能实现静态NAT（我们将在本章后面讨论这个问题），但它能够完成相似的功能。

注意，在屏幕底部有一个“设置”（Settings）按钮。单击它以后，选择“服务”（Services）选项，此时屏幕的显示如图 4-10。在这个示例中，已经定义了一个服务叫“ telnet”。缺省情况下，这个列表是空的。如果单击“编辑（Edit）”按钮，我们将看到如图4-11所示的屏幕显示。

在服务的端口号域中，内容是 23（这是Telnet服务器缺省的口），协议是TCP，名字域的内容是Portabeast，它是我们内部示例网络中一个机器的名字。

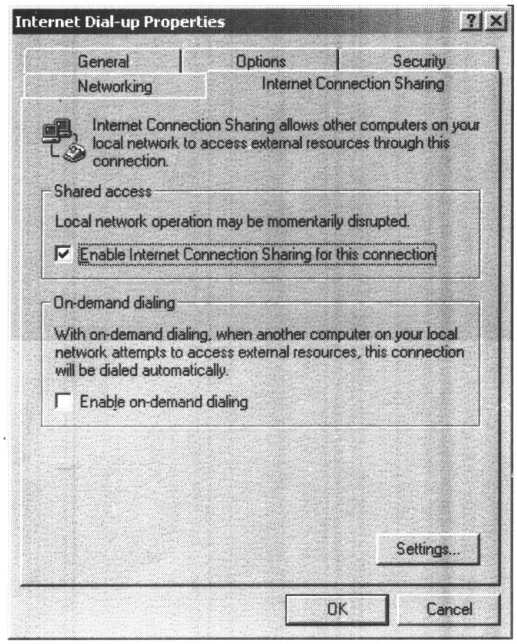

图4-9 拨号属性窗口中的ICS表

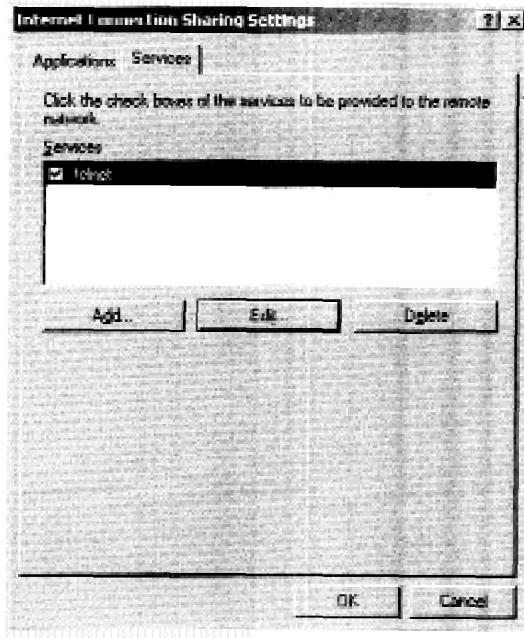

图4-10 ICS服务（Services）选项，选择了Telnet服务

由于ICS不是真正意义上的NAT，所以内部机器能够访问外部，但外部机器不能到达内部。这个服务特征将允许你明确定义从外部到达内部的服务。在这个例子中，我们允许从外部Telnet 到portabeast服务器。ICS也能够自动正确处理FTP的访问。

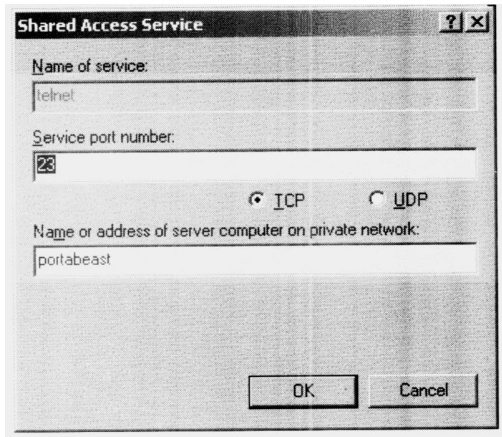

图4-11 Telnet服务的定义

2. Cisco的IOS

在我们将要讲述的三个操作系统中， Cisco的IOS**是最灵活的**NAT**软件。使用这个软件，我们将能够真正实现静态** NAT**的配置。这个示例来自于一个** 2621**路由器，它有两个快速以太网接口。在我们开始前，先看一看下面的相关配置：**

```
Using 827 out of 29688 bytes
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname NAT
!
enable secret 5 xxxxxxxxxxx
enable password 7 xxxxxxxxxxx
!
ip subnet-zero
!
!
interface FastEthernet0/0
ip adderss 192.168.0.1 255.255.255.0
no ip directed-broadcast
!
interface Serial0/0
no ip address
no ip directed-broadcast
!
interface FastEthernet0/1
ip address 130.214.99.254 255.255.255.0
no ip directed-broadcast
!
```

```
ip classless
ip route 0.0.0.0 0.0.0.0 130.214.99.1
no ip http server
!
!
line con 0
transport input none
line aux 0
line vty 0 4
password 7 xxxxxxxxxxx
login
!
no scheduler allocate
end
```

Interface FastEthernet 0/0是我们的内部网络接口，使用的是 192.168.0网络。130.214.99是我们的外部网络，在这个例子中代表通向 Internet的路径。

这里有个内部机器，地址为 192.168.0.2，它想实现与外部的连接。此时我们可以给它分配一个外部地址。

```
NAT(config) #interface fastethernet 0/0
NAT(config-if) #ip nat inside
NAT(config-if) #int fastetherent 0/1
NAT(config-if) #ip nat outside
NAT(config-if) #exit
NAT(config) #ip nat inside source static 192 . 168 . 0 . 2  130 . 214 . 99 . 250
```

第一步是用来标记内部和外部网络接口的，使用的是 ip nat inside和ip nat outside命令。下一步我们要告诉路由器实现一个 IP地址映射。这个命令尽管不是一个网络接口命令，但也使用 ip nat格式。我们将映射一个内部内址，并且改变源地址（使用 IOS，也可以实现目的地址转换）。这是一个静态映射，我们将地址 192.168.0.2转换成了130.214.99.250。

这是一个真正的静态映射。外部使用地址 130.214.99.250仅能到达一台内部机器。

象前面提到的那样， IOS同样也支持目的地址映射。如果需要的话，可在一个物理路由器上加上两个NAT。

3. Linux的IP Masquerade软件

我们的Linux系统（Red Hat 6.0）也有两个LAN接口。IP Masquerade是Red Hat 6.0中的标准软件，但也可以用在其他版本的 Linux上，这就需要你自己安装了。要想获得具体的使用说明，请查看一下本章后边的“索引和资源”部分。我们的示例开始时假设 LAN网络接口已经配置，并且能够正常工作。下面内容是 ifconfig命令的输出结果：

```
eth0    Link encap:Ethernet  Hwaddr  00:80:C8:68:C8:44
        inet addr:130.214.99.253  Bcast:130.214.99.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:547 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          interrupt:11 Base address:0xfc00
eth1      Link encap:Ethernet  Hwaddr  00:60:97:8A:9D:30
          inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          interrupt:3 Base address:0x300
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING MTU:3924  Metric:1
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

在这里的地址管理设置与前面路由器的地址管理设置非常接近。接口 eth1是我们的内部网络，地址是192.168.0；接口eth0是我们的外部接口。使用 IP Masquerade，内部被转换的地址是由信息被路由的方向来决定的，它将使用外部网络接口的 IP地址。下面是一张路由表，它是通过netstat -rn命令输出的。

```
Kernel IP routing table
Destination     Gateway     Genmask     Flag  MSS  Window  irtt  Iface
130.214.99.253      0.0.0.0         255.255.255.255  UH   0   0   0  eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     0     0     0  eth1
130.214.99.0    0.0.0.0         255.255.255.0   U     0     0     0  eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0     0     0  lo
0.0.0.0         130.214.99.1    0.0.0.0         UG    0     0     0  eth0
```

由于缺省的路由（0.0.0.0）指向130.214.99.1，而通过eth0接口能够到达这个地址，所以除目标地址为192.168.0网以外，所有的流量都从这个网络接口输出。因此， eth0接口的IP地址130.214.99.253将被用做被转换后的源地址。

由于IP Masquerade软件依赖OS进行路由，所以路由功能应被允许（缺省情况下是不允许）。为了打开路由功能，使用下面命令：

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

这条命令将打开转发功能，但要等到重新启动后才能生效（用相似的方法可以关闭此功能）。为了能够在Red Hat系统中永久打开此功能，可以编辑一下配置文件 /etc/sysconfig/network，按下面内容进行修改。

将行：FORWARD_IPV4=false，改成 FORWARD_IPV4=true。

这条命令负责实现转发（路由）过程。下一步就是要安装一个实施转换的规则。 IP Masquerade能够很好地控制 FTP协议。事实上，要处理 FTP还需要安装一个可装载的模块。使用下面命令进行安装

```
/sbin/modprobe ip_masq_ftp
```

从名字上来看，这个模块很明显是用于FTP的。对IP Masquerade软件来说，还有几个其他类似的模块，我们将在本章的后面进行详细的论述。下面，我们将设置一些超时值：

```
/sbin/ipchains  - M  - S 3600 60 180
```

第一个数字3600说明的是空闲TCP连接能够保持的秒数（这种情况下是1小时）；第二个数字60说明的是在FIN标记交换完成后连接要保持的秒数，最后一个数字180说明的是在没有任何流量的情况下，UDP连接保持的秒数。

最后，我们设置实际的IP Masquerade规则：

```
/sbin/ipchains -P forward deny
/sbin/ipchains -A forward -s 192.168.0.2/32 -j MASQ
```

（192.168.0.2还是示例中内部机器的地址）。

此时，我们的内部机器就能够实现Internet连接了。如果你不希望在每次系统重新启动后输入这些命令，则可将这些命令放入到/etc/rc.d文件中的一个shell脚本中，这样当系统启动时，它们就能够自动运行了。

4.3.6 动态网络地址转换

静态NAT就是一对一的NAT。动态NAT就是多对多的NAT。注意，一对多的NAT是多对多NAT的特例，所以这里不把它当作一个独立的问题来讨论。如果能够实现多对多的NAT，那么也就能够实现一对多的NAT。

我们已经看到一对一的NAT工作过程，并且知道NAT能够减少对IP地址的需求量。动态NAT也是出于上述考虑才出现的。动态NAT的工作是实现将一组内部IP地址映射到一组外部IP地址，当然这组外部IP地址通常都比较小。在运行中，它根据需要动态建立一对一NAT映射。通过流量监控和定时器，它会根据需要清除一些映射，并为新的内部客户释放外部IP地址。此时，你也许已经发现了一些问题，但不要着急，记住这些问题，本章后面有关PAT一节将讨论这些内容。

这里是示例的背景：现在你得到了一个内部网络地址10.0.0.X，有50台机器在这个网上。现在你想实现与Internet连接，但你的ISP仅能为你提供16个地址，从192.138.149.0到192.138.149.15。由于标准的子网划分问题，0和15不能被使用。ISP的路由器使用1，你的路由器使用2，最后只留下3到14，共12个地址。一般来说，你肯定希望所有的内部机器都能访问Internet，这就是你进行Internet连接的目的。

图4-12给出了一个连接示意图。从前面的讨论我们已经知道，我们可以仅用一个外部IP地址和一个代理服务器来完成上述功能。对于这个示例，为了避免理论上设立一个新的专用服务器的花费，我们将使用动态NAT。

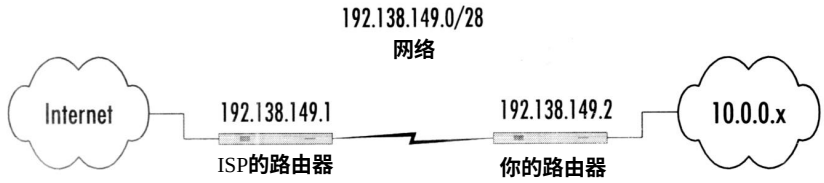

图4-12 通过ISP提供的16个地址连接到Internet

我们已经知道，可以使用的IP地址范围是从192.138.149.3开始到192.138.149.14结束。我们的路由器将使用这些地址并把它们看成一个外部地址池，同时也把10.0.0.X看成是一个内部地址池。单词“pool（池）”可以被简单地理解为一个IP地址系列。为了了解动态NAT所做的工作，路由器将需要知道NAT要负责哪些IP地址。因为NAT使用的外部地址非常明确，所以路由器能很容易获得外部地址信息。对于内部地址池来说，使用那些地址并不是很直观。为什么NAT仅转换部分内部地址呢？这里有两个原因。首先，你也许只希望让一部分内部地址映射到一个外部地址池，而让其他部分映射到不同的外部地址池中。其次，你也需要对某些机器使用静态NAT，如一个mail服务器，因为你不希望这个机器的地址经常被动态转换。

4.3.7 动态NAT是如何工作的

为了实现动态NAT，路由器要完成哪些工作？前面我们已经简单地讨论过一个路由器实现动态NAT，所要使用的所有信息。它需要有一个状态表，记录连接的开始和结束时间以及定时器的值。

我们已经知道静态NAT的工作过程。为了讨论动态NAT，我们先假设一个正在工作的静态NAT有一个状态表和协议说明。现在需要对它进行扩展。第一个主要变化就是静态NAT映射结构不再是很难处理的（以前由管理员手工配置），路由器能够按需求改变部分表的内容。当我们开始时，表为空的，没有一对一的映射。除非有一个内部机器连接到Internet，否则这个表将维持不变。

此时我们可以看到，在安全问题上动态NAT比静态NAT略有提高。对于静态NAT，任何在Internet上的机器都可以在任何时刻连接到静态NAT中的外部IP地址，并能够进入内部。如果使用动态NAT，则缺省时没有任何对外部IP地址的映射。当映射表为空时，任何与外部IP地址的连接都是不能完成的，同时外部也就不能够连接到内部机器上了。如果出于安全性的考虑，这些内容还是远远不够的，但它毕竟还是一个提高。

当一个内部机器试图连接Internet时，路由器将参考这张表，并选择一个可能的、没有被使用的IP地址。在我们的示例中，由于当前表是空的，它将选择第一个可用地址。此时将在映射表中建立一个入口，同时建立内部机器的IP地址与所选择的外部IP地址之间的一个（临时）静态映射。要注意的是，从路由器观点来看，到Internet的连接过程是非常简单的：当路由器得到从内部到外部的分组，它将会建立一个映射，同时路由器也开始启动一个定时器。

当内部机器与外部机器之间有信息交流时，这个映射将一直保持。每当一个分组经过一次映射时，定时器都将被重新设置。

有两种情况会将这个映射删除。第一种情况是连接正常终止。例如： FTP会话已经完成，并且客户已经退出。要完成这项工作，路由器要知道连接结束的标志。对于 TCP连接，这相对比较容易，因为它有一个特定的标志来说明连接的结束。当然，如果路由器要想检测到连接结束标记，它将从开始就要一直检查。本章有关 PAT一节将详细讲述它的工作过程。第二种映射被删除的情况是在定时器设定时间内，没有任何信息传递。当定时器的定时完成时，就说明此时的通信已经结束，该映射将被删除。

以此类推，当一个内部机器与 Internet进行通信时，其他的机器也可以以同样方式开始，并得它们自己相对独立的映射。

4.3.8 动态NAT的问题

到目前为止，动态 NAT问题已经变得很明显。假设我们使用最简单的方式。当一些分组想要从一个内部机器到达 Internet时，路由器应建立了一个映射。当定时器到时时，连接将被释放。此时，这映射还要维持一段时间。如果我们有 50台机器，但仅有 14个外部IP地址，这将导致在一天中的某些时间段上产生连接拥塞问题，如早上或午餐时间，此时每个人都希望访问 Web。

如何解决这个问题呢？一种缓解这个问题的方法是提供更多的 IP地址。在我们的示例中，这种方法很不实际，因为我们只能从 ISP那里获得这些地址。另外，要想使 50台内部机器在一天中的某个时间同时访问 Internet，那么我们应有 50个外部地址。从这个观点来看，我们又回到了静态NAT，并且也没有节省地址的开支。

另外一种方法就是试图减少映射所持续的时间长度，这会在高峰时间里，给向外连接的内部机器更多的机会。由于我们减少了定时器的值，所以增加了连接被删除的机会。如在一个连接上，一台内部机器正在等待一台在 Internet上的慢速服务器的应答，如果这个连接被中断，将会导致分组到达错误的内部客户。

其他减少时间的方法就是提高路由器识别连接是否完成的能力。然而，这个问题比较复杂，一般来说，一个客户在连接 Internet时，同一时刻可能会同时打开多个连接。如 Web冲浪，至少在HTTP 1.0版本以下时，索引一个 Web页上的多个不同元素时，都需要不同的连接。如果连接的Web页中有 10张图片，这将会引起至少 11个连接 —— 1个是HTML页面，10个是针对图片的。所以路由器不能只看到部分连接的结束，而要看到所有连接的结束。这就要求路由器知道此时有多少个连接发生。为了计算这些连接个数，路由器不得不从连接一开始就对其进行监测。

这些内容由另外一张表来控制。当一个连接开始时，在表中建立一个入口，表中的每一个入口都有它自己的定时器，而不是使用整个内部机器的全局时间。对于面向连接的协议，如 TCP协议，由于在连接中它们有清楚的开始和结束，对于这些协议它能工作得很好，但是对于

一些无连接的协议如UDP和ICMP，它就不能够进行处理了，此时应使用定时器。

总之，这里所描述的动态NAT并不能够胜任所有工作。在我们的示例中已很清楚地表明，在同一时刻，内部仅能有14个人连接到Internet，其他人则不能实现对Internet的连接。

很明显，系统应该有某种机制来保证任何一台内部机器都能够公平地访问Internet。动态NAT不能实现这一点。本章后面PAT一节将详细讨论这个问题。

4.3.9 配置示例

很不幸，多对多动态NAT的配置示例非常少。事实上，在前面所讲的三个示例中，仅有Cisco的IOS支持多对多的NAT。

我们将会看到一个使用IOS来实现的多对多示例。对于这个示例，我们先返回到我们看到的第一个配置（不是NAT配置）。具体命令如下：

```
NAT(config) # interface fastethernet 0/0
NAT(config-if)# ip nat inside
NAT(config-if)# int fastethernet 0/1
NAT(config-if)# ip nat outside
NAT(config-if)# exit
NAT(config)# nat pool dynpool 130.214.99.200 130.214.99.250 mask
255.255.255.0
NAT(config)#ip nat inside source list 1 pool dynpool overload
NAT(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

前五行与前面内容一样，第六行定义了一个缓冲池，叫dynpool。它的有效的地址范围是从130.214.99.200开始到130.214.99.250结束。当路由器使用这些地址时，子网掩码应为255.255.255.0。

下一条命令是NAT命令，它的开头为ip nat inside source。在这种情况下，我们将配置一个存取列表，以便取出源地址，被转换的地址来自于dynpool地址池。过载（overload）这个词意味着内部地址个数多于外部地址池中的地址数目，此时，路由器将使用特殊方法（见下一节PAT）来处理这种情况。最后，我们定义列表1，在前面的命令中我们可以看到它。列表1仅简单地定义了内部IP地址范围。

使用这种配置，当内部机器想访问外部机器时，路由器将动态地从地址池中选择一个地址分配给它。当要测试这个配置时，分配的IP地址应该为.200。

4.3.10 端口地址转换

有一种方法能够解决静态和动态NAT所引起的多个内部机器共享一个外部IP地址的问题。这种方法被叫做端口地址转换（PAT）。像我们所看到的那样，要想使动态NAT正常工作，PAT是必须的，所以许多人就把PAT当成了动态NAT，销售商也经常把PAT简称为NAT，所以你需要仔细了解产品的性能，以便确定它是哪种类型。Checkpoint公司有一个非常通用的防火墙产

品Firewall-1，在此产品的说明中将 PAT认为是“ 隐藏的 NAT ”，即许多内部 IP地址都隐藏在一个IP地址下。

这种命名上的混淆有两种原因：首先， NAT是销售商的市场部给其产品的定义，所以非常容易产生混淆。其次， PAT是现在NAT所使用的主要结构（尽管静态 NAT有时也是安全结构中的一个重要部分）。所以许多带PAT能力的产品的销售商由于过于简化，则把这些特征的集合叫做NAT。象其他产品的一样，如果你正在考虑购买一个产品，则要仔细查看产品的技术文档，以使明确知道它所具有的能力。

两个内部机器共享一个外部 IP地址会带来什么问题呢？如果你已经学习过以太网知识，在以太网环境中没有冲突，但在端口号和 IP地址是存在冲突的。让我们看一看共享一个外部地址的情况。如果内部的两台机器向 Internet发出请求，当它们的应答到达时，目的地址是一个外部IP地址。此时，路由器将如何决定将这些分组送到哪个内部 IP地址上呢？

再让我们看一看一个较复杂的 NAT路由器。它尝试使用一个外部 IP地址为多个内部机器提供服务。在动态 NAT那一节，我们已经讨论过，当连接被建立和转换时，路由器能够记录这个独立的连接。加入这个功能好象能够解决路由器不知道将分组送到哪一个内部 IP地址的问题。这只需要扫描路由器的内部连接表，寻找与分组相匹配的连接。当找到对应的连接时，它将查看这个连接的内部IP地址，在完成相应的转换后，将分组传送给相应的内部机器。

工作完成了吗？还没有。重新回到发生冲突这个问题。现假设两台共享同一个外部 IP地址的机器向ISP的DNS服务发送了一个请求，由于 DNS是由ISP维护的，从客户的角度来看，它应该在Internet上，至少不在NAT路由器的客户那一端，此时就会产生转换问题。先看一看在我们谈到的连接表中有哪种类型的信息：肯定会有 Internet IP地址（服务器）、内部IP地址（内部机器的实际地址）以及外部 IP地址（内部地址将被转换到的），另外一些要记录的内容还有用于这些连接的TCP和UDP的源端口号和目的端口号。在我们的示例中，假设要记录所有内容。

再回来看一看客户与 DNS服务器通信的问题：它们将会把分组送到同一个服务器 IP地址、同一个端口号上（客户对 DNS请示的UDP端口号为53）。我们已经知道它们共享一个外部 IP地址，所以在连接表中存在两个独立的“ 连接 ”（由于UDP是无连接的，所以这里使用了引号）。其中这两个连接的 Internet地址相同、外部IP地址相同、目的端口地址相同。不相同的内容是内部IP地址和源端口号。这个请求将会被正常发送出去。

此时这两个来自于内部不同机器的请求非常相似，仅源端口号和分组的数据部分不同。

当应答信息返回到外部 IP地址时，分组中的不同内容仅为原来的源端口号（因为路由器并不知道这个应答是对应于哪个 IP的请求；这正是我们要重点指出的）。要特别注意一下目的端口（在应答分组中，源端口和目的端口的位置与请求分组相反），它能通过比较内部机器的源端口来决定将应答信息送到哪个机器上。

此时，就可能会引起冲突。大部分操作系统都从 1 025开始分配源端口号。按顺序向上增加，出于巧合，正好两台机器同时使用了一个源端口号。如果它们要与在 Internet上的同一个IP地址

通信的话，返回信息将有相同的目的端口号。除了内部 IP地址外，每一项内容看起来都一样。这就会造成当分组到达路由器上的外部 IP地址时，两个应答信息的内容不能被区分开的。

这个问题的关键是两个请求中的头部是完全相同的，但数据部分不同。此时 NAT不得不要确定哪个分组要到哪个机器。

4.3.11 PAT是如何工作的

从统计学观点来看，与直接使用动态 NAT相比，使用PAT发生冲突的机会会更小，有时甚至可以忽略这种冲突。这就是 PAT的出发点。也许你能从这个名字看出什么，但要知道 PAT是用于转换端口号和IP地址的。当发送方向向外时，在转换源地址的同时，也转换源端口号。

为了避免冲突，路由器往往要选择一个新的源端口号。这种解决办法至少对 TCP和UDP来说能工作得很好，并能够避免冲突。对于没有端口号的 ICMP协议，还要使用一些其他的技巧。

现在，通过分组信息与连接表进行匹配，路由器能够找到一个唯一的端口号。 PAT能够实现大量内部机器共享一个外部 IP地址，它能够支持多少台机器呢？由于与使用方式相关，所以给出一个精确的数字是很困难的。现在让我们做一个假设。假设要限制一个 Internet IP地址在同一时刻同时通信的内部机器台数。最坏的情况发生在 UDP协议上，此时我们不得不使用定时器来仿真连接。假设定时器被设置成 2分钟，如果2分钟之后仍然没有分组传送的话，连接将被中止。端口号的范围从 0开始到65 536结束，从原理上来讲，同时连接的最大数量应为 65 536。这需要它们在某一时刻同时发生，产生这种情况的原因有两个：一个原因是它们在同一时刻开始，并且已等待了两分钟；另一个原因是这些连接已经被激活很长时间，并达到了最大数量。这里针对的是一个外部 IP地址。如果使用动态 IP地址的话，连接数量应为带有 PAT功能的动态NAT所使用的IP地址数量乘以1个外部IP所能支持的连接数量。

记住：这里所讲到的内容只针对于所有客户同 Internet上的同一机器进行的通信。如果考虑在Internet上有多个机器，冲突机会几乎下降到零。这种情况在实际应用中看起来很好。但是，当它达到理论上的极限时，你的 NAT设备中的内存将会被全部消耗掉。

PAT的安全情况怎么样？情况还比较好。一个外部地址不再对应单个的内部 IP地址，而是依靠连接来实现。这也就是说，如果有一个与外部地址相连的新连接，则在连接表中不会有与它相同的内容出现，所以也就不存在要连接的内部 IP地址了。当一台内部的计算机试图连接一个外部地址时，这种情况是最常出现的。理论上讲，可以设计出一个 PAT，能够使一个特定的外部地址匹配特定的内部地址（将静态 NAT和PAT合在一起）。但为了安全起见，你可能不希望地这样做。另外一个要注意的问题是外部的 IP地址并不是 NAT设备接口的那个 IP地址。例如，有些路由器在实现 PAT时使用路由器自己的外部 IP地址。在这种情况下，试图与外部 IP地址进行的连接将会连到路由器，这是我们不希望的。

许多PAT的实现仅允许一个特定的内部地址缓冲池与一个外部 IP地址匹配。产生这种情况的原因可能是系统仅允许一部分内部网络匹配一个外部地址。

让我们看一看已经讨论过的这些连接表中的内容。它们包括：内部源 IP地址、外部源IP地址、目的Internet IP地址、原始的源端口号、被转换的源端口号、目的端口号、传输协议、 FIN标志和定时器。 FIN标志有两个，分别描述两个方向的 FIN交换是否完成。对于 TCP的连接，如果正常关闭的话，则每个方向都被关闭，所以我们需要记录每个方向。当两个标志都被设置时，则整个连接已经完成。如果出现的是 RST，则不需要这种标志，连接将会马上终止。

图4-13包含一个连接图。我们将使用它作为示例。在这个图中，内部机器地址是 10.0.0.2，路由器的外部 IP地址是 192.138.149.1。我们要连接的、在 Internet上的服务器地址为 207.244.115.178，在Web服务器与路由器之间的连线表示它们之间有 Internet连接。

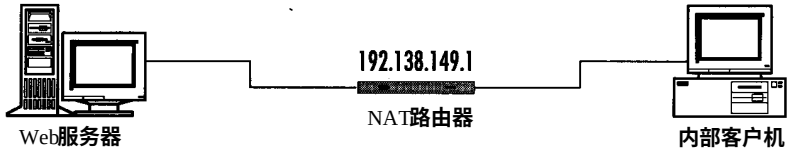

图4-13 PAT排列示例，路由器使用它自己的外部IP地址

内部机器向Web服务器的80号端口发送一个 SYN分组，使用的源端口号为 1030。下面连接是表中的一条内容。

源地址	目的地址	转换地址	源端口号	目的端口号	转换端口	协议	源FIN	目的FIN	定时器值
10.0.0.2	207.244.115.178	192.138.149.1	1030	80	5309	TCP	Off	OFF	2:00

所有描述方向的标签都来自于第一个分组， SYN分组，它的方向是从内部到外部。在反方向的分组中，许多内容都要被颠倒过来。所以路由器要注意记录到达接口的分组。

下面是一个比较简单的 SYN分组的头部，每个分组刚离开内部机器。

目的地址	源地址	目的端口号	源端口号	标志
207.244.115.178	10.0.0.2	80	1030	SYN

在通过路由器后，分组的头部变化如下：

目的地址	源地址	目的端口号	源端口号	标志
207.244.115.178	192.138.149.1	80	5309	SYN

注意，源地址和源端口号都已经被转换。下面是一个来自于 Web服务器的应答分组的头部。

目的地址	源地址	目的端口号	源端口号	标志
192.138.149.1	207.244.115.178	5309	80	SYN-ACK

源地址和目的地址已经被颠倒过来，标志位被设置为 SYN-ACK。这个分组将会到达路由器的外端，路由器要根据这个分组中主要域中的内容进行选择。路由器所要做的就是要将最左

边的四个域与连接表相匹配，如果发现了相同内容，它将路由这个分组，并恢复原来的源地址和源端口号（即现在的目的地址和目的端口号）。

目的地址	源地址	目的端口号	源端口号	标志
10.0.0.2	207.244.115.178	1030	80	SYN-ACK

路由器要转换回来的地址和端口号可通过简单地查询连接表来获得。如果下面的三个条件一个都没有满足前，则连接表中的入口将保持不变。

- 两组FIN都已经到达。
- 一端送来一个RST分组。
- 定时器超时。

定时器将被定期检查，以便确定时间是否超时。另外，每次在这个连接上分组被路由时，定时器都要被重新设置成为两分钟或所需要的其他值。

除了不使用FIN或RST分组来描述连接的结束外， UDP的工作过程完全相同。它要依靠定时器来断定UDP连接是否结束。

4.3.12 PAT带来的问题

使用PAT会带来什么问题呢？ PAT不仅有着静态NAT所引起的所有问题（例如，不得不转换在分组的数据部分中的地址），而且还会引起两个新问题。我们有关 PAT的讨论是基于一种全功能的静态NAT的设想，任何在分组中数据部分传输 IP地址的协议，如 FTP协议，我们都应该能够很好地处理。共享一个外部 IP地址能够达到类似防火墙的效果，它不允许在 Internet上的机器连接到内部。

FTP服务器就是这个问题的很好示例，我们首先假设分组中的数据部分（ FTP PORT命令）已经被正确修改。此时，当 FTP服务器试图使用所给定的端口连接一个外部 IP地址时，会发生什么情况？由于在连接表中没有对应的条目，则操作失败。

解决方案也很明显。当 NAT软件在修改PORT命令时（与其他连接一样，它也要改变端口的值），在连接表中建立一个入口。

在图4-9的示例中，我们用 FTP协议替换HTTP协议。在开始的连接被建立后，连接表内容如下：

源地址	目的地址	转换地址	源端口号	目的端口号	转换端口号	协议	源FIN	目的	定时器
10.0.0.2	207.244.115.178	192.138.149.1	1042	21	6123	TCP	FIN Off	Off	2:00

在连接期间， FTP客户将发出一个 PORT命令。在我们这个示例中，使用 PORT 10，0，0，2，4，19命令。其中端口号部分 4，19被转换成的十进制数为 1043，这是操作系统要发送出去

的端口号。此时路由器将不得不对这个 PORT命令进行转换，如果我们假设路由器要转换到的下一个端口号6177，则PORT命令将变成PORT 192，138，149，1，24，33（PORT中的字节计算为：24×256+33=6177）。另外，路由器必须将这个新端口加入到连接表中。现在连接表的具体内容如下：

源地址	目的地址	转换地址	源端口号	目的端口号	转换端口号	协议	源FIN	目的	定时器
10.0.0.2	207.244.115.178	192.138.149.1	1042	21	6123	TCP	Off	Off	2:00
10.0.0.2	207.244.115.178	192.138.149.1	1043	20	6177	TCP	Off	Off	2:00

现在，有了上述内容， PAT就能够正确地处理 FTP。数据的连接将被当作一个独立的连接来处理，同其他的 TCP连接一样，在某些环境下，连接将被删除，最后，我们达到了节省地址的目标。这也正是我们为什么使用 NAT的最重要的原因。

注意 当传送的数据返回到客户时，FTP服务器使用的源端口号为20。

使用了上述内容后， PAT能工作的很好。有时偶尔还会出现一些小问题，尽管这些问题有时很难解释，但在 Internet上的一些服务器在被连接时，它们特别注意使用的源端口，这些服务器大部分出现在 DNS上。从传统来说，当两个 DNS服务器使用 UDP进行通信时，它们将使用53号端口做为目的端口和源端口，这只是一种习惯，不是一种硬性的规则。如果此时我们正在转换源地址，就会产生麻烦。在 Internet上有一些站点在配置它们的 DNS时，定义的连接仅能来自于53号端口。

过去的apple.com和intel.com就是这样，当然还有其他公司。要实现让对方修改以满足你的要求是件很困难的事情，如果你发现在使用的 DNS服务器有些问题，则可以将内部 DNS改成静态，这样源端口号 53将在向外发送时不发生变化。这种情况仅用于你是在运行自己的内部 DNS服务器。如果你使用的是 ISP提供的DNS服务器（它在外边），则它不会产生这些问题。

4.3.13 配置示例

总之，几乎所有的配置示例（除 Cisco的静态NAT示例）都是PAT示例。尽管有时仅进行一个地址的转换，从它们内涵来看， ICS和IP Masquerade都是PAT产品。IOS能够根据你的配置来决定是否使用 PAT。现在我们有机会学习得更深一些，并且有更多的示例。

实际上，没有 PAT的NAT是不能够工作的，这就是要讲述下面内容的原因。我们所讨论的所有问题都说明单纯的 NAT是没有用的。

1. Windows NT 2000

有关第一个示例中的 ICS，我们没有什么更多的要说。它是一个 PAT产品，所有的内部 IP地址都被定义到192.168.0网段上；向外被端口转换的所有内部地址都使用单一的拨号地址。然而，这里还有我们以前没有看到过的其他选项。通过使用“ 设置 ”按钮，一张表将会出现在窗口中，

如图4-14所示。

象Services屏幕一样，在这里可以定义特殊的应用控制。这其中就包括 FTP，需要对它实现反向连接。与以前我们看到的 FTP处理器相比，它缺少灵活性。使用 FTP处理器时，为实现连接，要长时间打开一个必要的端口。在这种情况下，一旦服务启动，内部将有一系列端口被打开。由于外部只有一个口被打开，这样会引起多对一 NAT的所有问题，会带来更多的冲突。使用它只能让一个应用很好地工作，其他应用都不可以。尽管不理想，但有选择比没有选择好。

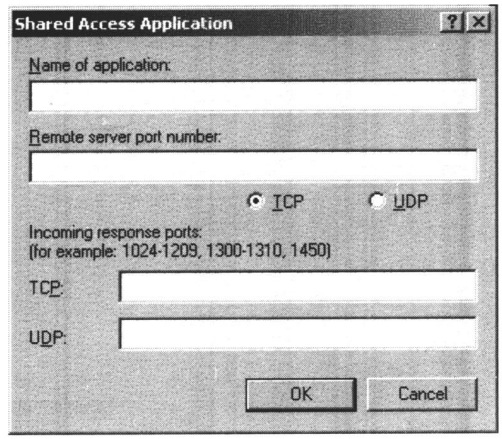

图4-14 ICS反向连接设置

由于这个软件还处在 beta版阶段，所以文档资料也比较少。我曾试过，不需要什么特殊配置，FTP就能够正常工作。一些其他协议的处理可能还需要一些特殊的方法，但 Microsoft没有说明这一点。

ICS存在的最大问题是它仅能工作在拨号状态下，并且必须使用 DHCP。这也就是说，它不能用于Cable调制解调器、 DSL线路以及通过 LAN网络接口进行连接的任何技术。 Microsoft还销售一个高端产品Microsoft Proxy Server(MSP)(微软代理服务器)，它非常灵活，但价格较高，为1000美元。

还有一些其他的解决方案，价格一般在 0到1000美元之间。本章后面的“索引与资源”这一节将给出一些基于 NT的商用NAT产品列表。对我个人而言，我很庆幸我使用了 Sygate软件，它最贵的版本（没有内部用户限制）仅需 300美元。

2. Linux的IP Masquerade软件

当仅使用一个内部 IP地址时， IP Masquerade 软件也能实现PAT。它能够很简单地实现将静态NAT转换成多对一PAT。具体的操作如下：

将此行 /sbin/ipchains -A forward -s 192.168.0.2/32 -j MASQ

改写成 /sbin/ipchains -A forward -s 192.168.0.0/24 -j MASQ

它将负责整个的内部子网。

目前有许多关于如何使用 IP Masquerade的文档。在“索引和资源”这一节中将会找到许多

有关这些文档的连接。如果你计划在产品中使用 IP Masquerade，你首先要读一下这些文档。在阅读上述文档的同时，你也应该阅读 IP Chains文档（注意：我们使用的 ipchains命令主要用于配置IP Masquerade）。在Linux 2.2.x内核中，IP chains是一个内置防火墙。如果要使系统更加安全，仅使用IP chains是远远不够的。

下面再让我们看一看有关 IP Masquerade软件的其他方面内容。我们已经知道，这里有一个专门处理 FTP的模块。软件中还有其他的模块吗？回忆一下，安装 FTP处理程序的命令是 modprobe。使用命令modprobe -l可列出所有可安装的模块。具体的列表内容如下：

/lib/modules/2.2.5-15/ipv4/ip_masq_vdolive.o

/lib/modules/2.2.5-15/ipv4/ip_masq_user.o

/lib/modules/2.2.5-15/ipv4/ip_masq_raudio.o

/lib/modules/2.2.5-15/ipv4/ip_masq_quake.o

/lib/modules/2.2.5-15/ipv4/ip_masq_portfw.o

/lib/modules/2.2.5-15/ipv4/ip_masq_mfw.o

/lib/modules/2.2.5-15/ipv4/ip_masq_irc.o

/lib/modules/2.2.5-15/ipv4/ip_masq_ftp.o

/lib/modules/2.2.5-15/ipv4/ip_masq_cuseeme.o

/lib/modules/2.2.5-15/ipv4/ip_masq_autofw.o

我们的FTP模块也在列表中。一般来说可通过名字来判断这些 IP Masquerade模块。一些模块对我们来说很熟悉。当它们同防火墙或 NAT一起使用时，还会引起一些问题。这些模块包括的内容有FTP、Real Audio、Quake、IRC(特别是DDC发送)、CUSeeMe和VDOLive。

有一个位置可以获得IP Masquerade处理程序，如果不存在，则可发出请求。具体细节请查看本章4.9一节“ 索引和资源 ”。

3. Cisco IOS

我们已经看到Cisco PAT-这是一个“ 过载 ”的配置。与一般路由器的不同就是它能够通过使用路由器自己的IP地址使内部所有机器向外访问。

```
NAT(config)#ip nat inside source list 1 interface fastethernet 0/1 overload
NAT(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

上述命令告诉路由器使用存取列表 1（匹配所有的192.168.0地址），并使用路由器自己的快速以太网O/1口的IP进行转换，把它做为源地址。

下面是一个完全的工作配置：

```
!
version 12.0
service timestamps debug uptime
```

```
service timestamps log uptime
service password-encryption
!
hostname NAT
!
enable secret 5 xxxxxxxxxxx
enable password 7 xxxxxxxxxxx
!
ip subnet-aero
!
!
interface FastEthernet0/0
ip adderss 192.168.0.1 255.255.255.0
no ip directed-broadcast
ip nat inside
!
interface Serial0/0
no ip address
no ip directed-broadcast
!
interface FastEthernet0/1
ip address 130.214.99.254 255.255.255.0
no ip directed-broadcast
ip nat outside
!
ip nat inside source list 1 interface fastethernet 0/1 overload
ip classless
ip route 0.0.0.0 0.0.0.0 130.214.99.1
no ip http server
!
access-list 1 permit 192.168.0.0 0.0.0.255
!
```

```
line con 0
   transport input none
      line aux 0
      line vty 0 4
      password 7 xxxxxxxxxxxx
      login
      !
no scheduler allocate
end
```

一般来说，如果你想使用这个配置，你将不得不要修正 IP地址和接口名称。在这个例子中，保密字已经被去掉，可手工对此进行添加。当你让别人看你的路由器配置文件前，你最好要预先整理一下这个文件。

当仅连接一个 ISP时，这种类型的配置是很有用的。它可实现所有内部机器地址到一个外部IP地址的转换。

Cisco还有另外一个我们没有看到过的、令人感兴趣的特征。 IOS允许你检查连接表。以前我们曾看到过一些理论上的示例，现在再看一些实际示例。

下面的示例来自于IOS的静态NAT。

```
NAT#sho ip nat trans
Pro Inside global          Inside local       Outside local      Outside global
Tcp 130.214.99.250:1055    192.168.0.2:1055   130.214.250.9:23   130.214.250.9:23
```

Cisco不能用FIN标志和定时器。注意，由于 IOS在一个盒子内部使用了双 NAT，所以这里有4个地址：端口成对出现。

在这种情况下，内部机器 192.168.0.2已经telnet(端口为23)到IP地址130.214.250.9。源地址转换成130.214.99.250。在左边，你所看到传输协议是 TCP。

下面是有关动态NAT的配置示例（使用一个外部地址池）。

```
Pro Inside global          Inside local       Outside local      Outside global
udp 130.214.99.200:1063    192.168.0.2:1063   130.214.250.43:53  130.214.250.43:53
udp 130.214.99.200:1068    192.168.0.2:1068   130.214.250.9:23   130.214.250.9:23
udp 130.214.99.200:1066    192.168.0.2:1066   130.214.250.9:23   130.214.250.9:23
udp 130.214.99.200:1067    192.168.0.2:1067   130.214.250.43:53  130.214.250.43:53
udp 130.214.99.200:1064    192.168.0.2:1064   130.214.250.9:23   130.214.250.9:23
udp 130.214.99.200:1065    192.168.0.2:1065   130.214.250.43:53  130.214.250.43:53
```

地址池是从 130.214.99.200开始的。同一个机器所有连接所需要的地址都来自于这个地址

池。在这里，我们会看到更多的 Telnet连接和一些DNS连接（UDP端口为53）。

在我们的示例中，当所有的内部机器通过路由器的 IP地址访问外部时，静态路由表的内容如下：

Pro	Inside global	Inside local	Outside local	Outside global
icmp	130.214.99.254:256	192.168.0.2:256	130.214.250.9:256	130.214.250.9:255
udp	130.214.99.254:1069	192.168.0.2:1069	130.214.250.43:53	130.214.250.43:53
tcp	130.214.99.254:1070	192.168.0.2:1070	130.214.250.9:23	130.214.250.9:23

在这里，我们能看到 TCP、UDP和ICMP三个协议。注意，一个 ICMP连接的后面将会出现一个端口号。为了能够辨别这个端口号，一些 NAT设备将使用 ICMP的状态信息。目前还不清楚将来会发生什么。但是，路由器可能会使用 256或其他表示来取代 Ping数据流中部分，这样便于跟踪。

使用PAT配置命令后，下面所显示的列表与一个 FTP会话过程非常相似。

NAT#sho ip nat trans

Pro	Inside global	Inside local	Outside local	Outside global
tcp	130.214.99.254:1080	192.168.0.2:1080	192.138.151.73:21	192.138.151.73:21
tcp	130.214.99.254:1081	192.168.0.2:1081	192.138.151.73:20	192.138.151.73:20

NAT#sho ip nat trans

Pro	Inside global	Inside local	Outside local	Outside global
tcp	130.214.99.254:1082	192.168.0.2:1082	192.138.151.73:20	192.138.151.73:20
tcp	130.214.99.254:1080	192.168.0.2:1080	192.138.151.73:21	192.138.151.73:21

第一个列表来自于 FTP客户发送一条 ls命令后的情况。从这里我们可以看到向外连接的端口号为21，而反向连接的端口号为 20。第二个列表反映的是另一个 ls命令发送后的情况。注意，前面反向连接入口已经被去掉。如果有必要，可以使用下面命令手工将这些转换表设置为空。

NAT # clear ip nat trans

NAT # ohow ip nat drans

IETF WORK

4.4 使用NAT的优点

如果你已经读完本章前面各节的内容，你就可以知道使用 NAT所带来的一些好处。主要的好处是它能够使用相对较少的公共 IP地址来实现大量内部机器对 Internet的连接。当你要连接到其他网络时，它具有很好的灵活性。

管理人员参考　你实际需要多少 IP地址？

与几年前相比，现在有多种 Internet连接选择。这些连接包括：调制解调器、 ISDN、传统

的租用电话线、 DSL、Cable、无线网等。从价格、性能、可靠性、可能性上来看，它们之间存在着很大的不同，但它们之间也有一个共同的特征：需要的 IP地址越多，花费的代价就越高。从经济观点来看，地址使用的越少越好。 NAT能够有效地减少所需IP地址的数量。如果仅有一个IP地址的话， PAT在大多数情况下都能够让你所有的内部机器连接到 Internet。这个问题非常重要，因为一些存取技术仅允许使用一个外部 IP地址，如拨号存取（调制解调器、 ISDN）。现在如果你计划使用多台机器来提供多种公共存取服务的话，例如一个 Web服务器或DNS服务器，此时你可能需要多个IP地址。尽管拨号存取不适合于这种多台公共服务器的情况，但也不会引起很大的问题。你还可使用一个 PAT，通过一个IP地址进行内部Internet存取，并把其他的地址分配给你想运行的服务器。不要设想 NAT是一个完全安全的解决方案，它不是。你必须自己实现一个安全的解决方案，例如加上一个防火墙。

如果你有大量的内部机器，则你仅需在你的 NAT设备上使用少量IP地址就能够实现向外的存取，这就是使用 NAT的诱人之处。这个目标在实际应用中可通过一种特定类型的 NAT，即PAT来实现。 PAT允许多台内部机器使用少量 IP地址（也可能少到 1个）来实现对 Internet的连接。

在控制转换或冲突时， NAT会给你带来更大的灵活性。有时，一台机器不能正常运行或被移走，甚至客户机都已经被重新配置，此时你只需要改变一下路由器上的地址，让它指向一个新服务器或指向一个有新地址的已有服务器就可以了。这对于临时解决地址冲突问题是非常有用的。

4.5 性能问题

所有这些NAT特征会花费多少性能代价呢？花费的代价并不是很高。对于 NT的ICS，对性能的影响是很难评估的，因为它使用的是一个拨号接口。在进行对外拨号连接上， ICS将尽可能以最快速度进行连接。对于 IP Masquerade软件，有一些非常有意义的测试，但我没有能够看到这些性能测试。另外， Linux是一个“移动”产品，变化的非常快。当然在变化过程中，其性能也会被提高。 Linux能够运行于多种平台。如果在使用 IP Masquerade时产生了性能瓶颈，你最好要升级一下你的硬件。 Cisco在下面网址提供了一些参考数据：

http://www.cisco.com/warp/public/458/41.htmo#Q6

这些数据来自于三个路由器平台： 4500、4700和4500。不论分组的大小如何， 4500在10Mb以太网上速率能够达到 7.5-8.0Mbps；4700能够达到完全的 10Mbps。在快速以太网，对于64字节分组，速率可达到 24Mbps；对于1500字节分组，速率可达到 96Mbps。

当然，对于我们已经看到三个 NAT软件包，性能主要依靠所使用的平台。如果 NT ICS服务器运行在大量使用 CPU的游戏状态下，性能将会被大大降低。如果 Cisco路由器对经过的流量进行加密工作，则其性能也会降低。

与使用一般的路由过程相比，执行 NAT后的路由过程将会引起一定的延迟。如果站在较高

的水平，则路由的功能将会相对地简单。

1) 接收分组。

2) 检查校验和。

3) 查看路由表。

4) 降低TTL字段中的值。

5) 重新计算校验和。

6) 信息传送。

再看一看加入NAT后的路由功能。

1) 接收分组。

2) 检查校验和。

3) 如果要到外部接口，检查是否能够匹配连接表的入口。

4) 查看路由表。

5) 检查外端的接口是否被标记为 NAT。

6) 确定分组中被修改的部分。

7) 如果是新连接中的第一个分组，则建立一个表入口。

8) 如果是一条PORT命令或其他类似命令，则要重写数据部分的内容，并且要建立一个新的表入口。

9) 如果是一个FIN分组，删除它在这个表中的入口。

10) 按照需要修改分组。

11) 重新计算校验和。

12) 信息传送。

尽管CPU速度足够快，但随着这些步骤对内存读写量的增加，系统的延时也将会有少量的增加。这里有一个好消息要告诉大家，在大部分环境下，性能并不是一个问题。仅当路由器的负载很重时，NAT才会带来性能降低的问题。

IT专业人员参考　你将要选择哪种产品？

选择的因素主要依靠你所认为的最好的操作系统以及你所拥有的设备。如果你喜欢使用UNIX系统，则你的首选应该为IP Masquerade或类似的产品；如果你常使用NT，则应该选择一些基于NT的某种产品。如果从实际应用上来看， ICS是不可取的，因为它仅适用于一个家庭。对此你不要感到奇怪，因为它就是针对这个目的来设计的。也许你可以通过 Cable调制解调器进行网络连接，所以在某些情况下， ICS对于家庭来说也是不合适。如果你是一个网络专业人员，或者你已经使用了Cisco路由器，则你也许希望在路由器上实现 NAT。Cisco路由器不是唯一能够实现NAT的路由器，还有其他不同的品牌。在进行产品选择时，不仅要考虑 NAT的结构，而且还要考虑安全结构，所以选择一个能够运行在你要求平台上的解决方案是至关重要的。不

管你愿意还是不愿意，一旦你连接到 Internet，你将不得不要考虑安全性问题。这就要求你对运行平台的配置要尽可能地安全，这个系统应该是你所知道的最好的操作系统。

4.6 代理和防火墙的能力

现在，我们已经比较深入地学习了什么是 NAT以及它的工作过程。现在让我们讨论一下它的安全性的问题。在我们讨论 NAT时，我们已经间接地谈到了防火墙。在这里，我们首先了解一些有关防火墙的基本定义，然后再了解防火墙与 NAT软件有哪些相同点和不同点。

什么是防火墙？这有点象一个宗教问题。这也就是说，不同的人对防火墙的理解也有所不同。防火墙的原始含义就是一个护拦屏障。从结构上来看，主要用于着火时阻止大火的通过。例如，一幢楼的一部分墙或墙的一部分是防火墙，为了避免伤害，防火墙应被设计能将火焰分开一段时间，以减少伤害。有些人把防火墙当作护拦上的安全报警器。它不仅能够检测到在一段时间里的入侵者，而且能够对网络部分进行分割。如果网络中的一部分出现问题，则其他部分不会受到影响。

其他人认为，一个防火墙应具有 X特征、Y特征和Z特征。这些特征都是他们所希望的。一些人认为防火墙是网络安全结构的一部分，能够阻止流量的通过；而其他人则认为防火墙应允许某种类型的流量通过。

参与这些讨论的人们都是防火墙方面的专家。这些讨论时常会出现在有关防火墙的 mail列表中。现在，许多人对这种定义的混乱感到不安，这其中有许多是防火墙的发明人，他们并不同意这些术语。

实际上，防火墙是由卖防火墙的公司来定义的。由于几乎所有的这些产品都有一些共有的特征，所以情况并不象我们所看到的那样坏。现在我们首先要讨论一下防火墙的一些特征。

4.6.1 分组过滤器

人们会很自然想到，网络主要用于更多、更快地传递信息。除了被破坏的分组外，原始的路由器并不关心阻塞问题。它通过匹配校验和来决定分组是否被破坏。在 Internet早期，人们对安全问题关心的并不是很多。

在过去，我曾听说过有关这方面的事情。由于产生错误，人们试图过滤掉这些错误流量。当某人在某个位置上进行了一个错误配置后，流量就可能开始失控，此时会给在其他地方的一些人带来烦恼。鉴于这种情况，就出现了分组过滤器。

分组过滤器就是用来过滤分组的设备。它们大部分都是路由器，但也可能是用于此目的的主机，如Windows NT或Linux。早期的分组过滤器能够根据分组内部包含的 IP地址阻止一些分组的通过，随后，系统也能根据分组中的端口号对分组进行阻塞。现在的分组过滤器可根据多种标准进行过滤，其中的内容包括： IP地址、端口号、传递类型、 TCP头中的标志等。

这种分组过滤器早已成为传统的代理结构和用于屏蔽的路由器防火墙结构中的一部分（下

面将会讲到代理）。典型的应用就是阻止具有某种类型的流量通过。当它们检测到攻击信息时，也能阻止它们通过（如阻止来自于某个特定地址范围的所有信息）。

传统的分组过滤器（PF）有这样一种特性：它们不仅不改变分组的内容，而且也没有状态信息。换名话说，PF仅用来完成传送或不传送一个分组的任务，并且仅根据当前分组中的内容做出决定。另外，PF采用的是静态配置，也就是说它们不能根据流量来改变过滤器的规则。

许多分组过程器能够根据“已建立”标志进行过滤。这也就是说，它应该能够记录在整个过程中的对话。事实上，对于PF来说，“已建立”标志可简单理解为TCP头中ACK位被设置。

做为防火墙，PF还有许多严格的限制。让我们返回来再看一看对FTP的处理过程。假设你有一台内部机器，并允许FTP存取外部。有关控制通道的连接是非常容易。过滤器规则认为，内部IP仅能够到达一些外部IP地址的21端口。下一步，你要建立规则，允许已建立的分组从任何外部IP到达内部IP。此时连接控制将开始工作，并且你已经受到了保护。现在的问题是如何处理反向连接。返回的第一个分组仅ACK位被设置，这样已建立的规则将不能对此进行处理。它并不知道内部IP的哪个端口在等待信息，仅能知道端口号要大于1023。

对于PF，你所做的所有事情就是加入一些规则。规则允许将分组从任何IP的TCP端口20传送到任何TCP端口大于1023的IP地址。当然这样做这会引起一个很大的安全性漏洞。任何人只要能够知道端口号大于1023的所有地址，并且源端口号也使用20，他就能够攻击你。对于一些聪明的攻击者，这种防火墙将不会起到什么作用。

FTP就是一个既简单又熟悉的例子。如果你看一看过有关IP Masquerade的处理程序，你就会知道许多协议的示例都是采用这种方法来处理的。

然而，如果你有一台特殊的机器，没有运行端口号>1024的这种容易受到攻击的服务，并且加入许多安全措施。此时使用这种方法来配置PF进行流量控制是可以接受的，但要考虑本地的安全原则。这台机器通常被叫做堡垒主机。但问题是这种机器对于每天的用户来说是很少被使用的，所以它们不能被放在每个人的桌面上，充当主要的工作机器。但这种机器能够做什么呢？它可以充当代理。

4.6.2 代理

在本章的开始，已经讨论了一些代理问题。一个代理就是一台机器，有时也被叫做堡垒主机。它一般被配置成用于处理其他机器提出的请求，通常这些机器都是内部机器。我们将会马上看到代理的实际工作过程。

假设我们已经配置了PF，它仅允许流量从Internet传送到代理。由于我们配置的很好，所以有关Internet是否能够到达大于1023端口的问题已不再是我们所关心的内容。另外，在代理和内部机器之间，PF也是非常有用的，它能够阻止恶意的内部用户攻击代理。具体连接结构如图4-15所示。

一定要特别注意，图4-15只是一张逻辑图而不是一张物理图。尽管我们已经实现了图中的

所有相关内容，并且达到了所希望的效果，但是，这张图的有些内容也许不是必须的。例如，图中说明代理有两个接口，但在通常的使用中也可以不使用两个。如果能够很好地管理在过滤路由器上的地址，信息可进出同一个出口，并且不会带来任何困难。如果充当 PF的路由器足够灵活的话，这个设计仅需要一个带有三个接口的路由器就可以了，而没有必要使用两个带有两个接口的路由器，但通过这张图能够很容易地看到数据的流动。

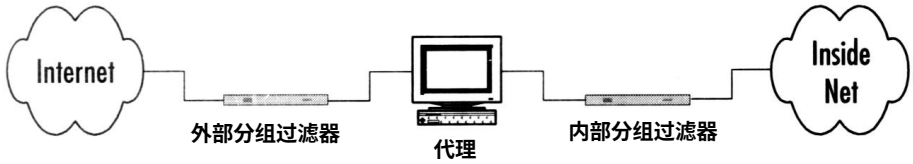

图4-15 被保护的代理服务器

内部的PF除了防止内部用户攻击路由器外，还有其他的功能。如果代理在某些方面设置的较好，它将能保护内部机器不受代理的入侵，这个概念非常重要，通常被叫做 DMZ（Demilitariaed Zone：非军事化地带）。对于防火墙方面的专家来说，术语 DMZ有两个不同的含义。一些学者认为 DMZ是一个防火墙外部接口以外的网络（在我们的例子，应在 PF的外部接口以外）。而我们所使用的定义是：“这是一个既不信任内部，也不信任外部的网络段，同时它也不被内部信任。”单词“信任”在这种情况下表示可自由地进行网络存取。例如， Internet在很大程度上信任任何人，每个人都可以对它进行存取。内部网络不信任任何人，所以没有人能够存取内部网络。从实际上来说，许多人认为 DMZ是在防火墙上的第三个接口上（第一个和第二个接口分别是内部接口和外部接口）。

一个代理是如何工作的呢？我们首先从传统的代理开始。一般来说，对内部机器来说，代理充当的是一个服务器；对 Internet来说，代理充当的是一个客户。内部机器既可使用被修改过的软件，也可通过操作程序上的变化来使用代理。传统的代理不是路由器。事实上，去掉堡垒主机上的路由器，如果你将一个分组传送到代理，而这个分组的目的 IP地址又不是代理的IP地址，此时代理就会将这个分组丢掉。在所有的 NAT示例中，分组的目的地址总保持着最终的目的地址（双NAT的示例除外），它们一般是Internet上的主机地址。而代理的工作方式有所不同，客户不得不按要求做一些相应的变化。

第一个要求就是目的IP地址必须是代理服务器的IP地址，不能是用户想通信的，在 Internet上的服务器地址，让我们看一个简单的示例： Telnet。

如果使用NAT类型的解决方案，则你只需要简单地 Telnet到你想连接的机器名或地址就可以了。让我们设计一个想象中的代理来处理 Telnet。首先，我们要写一个程序来监听网络连接，并且要在代理上选择一个运行端口号。一般来说，端口号为 23，这样以取代代理服务器上的Telnet机制。当然也可以运行在它自己的端口上。例如，我们选择的端口号是 2000，我们的程序将接收TCP的连接，然后提示输入一个机器名或 IP地址。一旦它得到了这个机器名，它将尝试连接那个来自于23号端口的机器名。一旦连接成功，则来自于外部机器上的 23号端口的输出

将被送到相应的内部机器上。同时，来自于内部机器上的的输出结果（如用户的键盘输入）将被送到外部机器上。

现在一个想telnet到外部的用户必须首先telnet到代理的2000号端口，并且要输入你真正想telnet的机器名。如果连接成功，它们将能看到来自于目的主机的输出，此时也可以输入进一步的内容。

当然，在实际应用中，Telnet协议并不是那么简单。尽管我们这个示例并不能说明它的全部内容，然而它却说明了一种基本的设计思路：让内部客户通知代理它想要完成的任务。代理要根据客户的要求进行连接，检索一些数据，并将内容返回给客户，然后再将来自于客户的输入送给服务器。

现在让我们看FTP是如何工作的。在这里使用FTP时，它所带来的问题与以前讨论的问题是一样，要颠倒它的连接顺序。代理的工作方式与PAT的工作方式非常相似，但在具体方法上略有不同。控制通道连接（到21号端口）的过程除了PORT命令外，与上面的Telnet代理示例非常相似。为了能够在数据流中识别PORT命令，代理将使用与PAT设备相同的方法来进行处理，并替换它的地址。此时代理将OS请求一个可能的端口号，并监听这个端口号，同时从这个端口号发送数据。此时代理还应保存着原来的PORT命令备份，以便以后使用。当外部服务器回连到代理时，代理将打开一个到内部机器的连接，并发送数据。这个内部机器在PORT命令中。

如果一个内部用户想使用FTP，这两种方式有什么不同呢？它们的不同仅体现在表示方法上。在我们的Telnet示例中，非常容易看到如何从用户获得一些输入信息。而FTP所面对的问题是它有多种类型的FTP客户程序。它们也许是使用命令行的客户，它们有很多方法进行输入工作；它们也可能是GUI FTP客户，每一件工作只需要点击鼠标就能够完成了。

在实现上的一种策略就是让内部用户使用一个特殊的用户名。例如，使用anonymous@ftp.example.com命令代替以前输入的anonymous。这条信息将告诉代理使用用户名anonymous连接到FTP服务器ftp.example.com上。保密字将保持不变。

对于许多使用给出用户名和保密字的FTP客户来说，它们都能够使用这种方式进行工作。但问题是当Web浏览器运行一个FTP连接时，它们能自动使用anonymous以及你写入到浏览器中的e-mail地址。他们将不会停下来给出提示。

总之，Web浏览器存在着一些问题。浏览器要如何才能够连接到代理，又如何将实际站点的URL送给代理呢？这里有一些技巧可以试一试，例如，使用一个特殊的URL，并且把代理看成是一个Web服务器。尽管有些问题，但从理论上说它还能正常工作。这些机制是非常不切合实际的，当用户很快地完成测试后，他们就会发出抱怨。

在使用代理进行存取上，还有另外一种相对独立的策略：使用特殊的客户端软件。这也就是说，客户端软件为了存取一个代理服务器将不得不被修改。此时，用户所做的事情与用户直接进行Internet连接是完全一样的。软件负责使用代理服务器的过程。当用户运行一个特殊的Telnet程序时，它将负责连接到代理，并且通知代理你要连接的服务器，这些工作对用户来说

都是透明的。用户所做的所有工作就是使用特殊的客户软件 Telnet到你想要连接的服务器。从原理上来说，许多客户端程序都能够完成这项工作，用户不必关心其中的细节。

现在的问题是市场上现在有太多的客户端程序，并且大部分没有公开的源代码，这样很难对其进行修改。如果都使用自己的 proxy软件，就会存在过多的代理协议。很显然，为它们制定一些标准是非常有用的。

目前所使用的代理协议标准有 SOCKS代理和CERN代理。在这里我们并不打算讲述它们的细节。CERN代理协议来自于 CERN HTT服务中的一个代理特性，由于它能够支持早期的 Web浏览器协议，所以它是很重要的。 SOCKS代理不仅能支持早期的浏览器，而且还能够代理任意的端口号。当然，你的 SOCKS代理服务器必须能够处理与端口号匹配的协议。

SOCKS协议来自于一些已经被重写的客户端程序，如 rtelnet和rftp，这些都是“ SOCKS化 ”的Telnet和FTP程序。它们都有 UNIX源码，你能够在大部分 UNIX平台上编译这些版本。在随后的发展中，第三方的 Windows应用程序也开始支持 Socks协议。现在，如果一个客户程序支持使用代理，则它通常能够支持 SOCKS协议。有关 SOCKS协议的进一步信息请参考“ 索引和资源 ”这一节。

有关SOCKS能够支持任意端口的想法会带来这样一个问题：是否存在一个通用的代理呢？这个代理的确是存在的。假设连接只是一个方向，没有反向连接，则它就能够代理一个数据流，这就好象是一个 Telnet连接。

这个代理通常被叫做“ 电路级 ”代理或叫插件网关（这个词来自于 Gauntlnt防火墙的play-gw特征。这个防火墙是一个通用的、基于代理的商业防火墙）。SOCKS代理一般都能支持这种方式。

实现客户请求发送到代理的另一种方法就是修改在客户端的 IP堆栈。这个软件一般被叫做“ 垫片 ”。Microsoft的代理服务器就是这样工作的，它为 Microsoft Windows客户提供了一个垫片。对于非Windows客户，MSP也能支持SOCKS协议。在这种方式下，如果协议简单，或者处理程序都已经被设计好，则 MSP能够支持在Windows平台上的任何客户端程序。

最后，在我们离开有关代理的话题前，一些代理还有一个透明选项，它能够改变以前代理的工作方式。像我们以前所讨论的那样，传统的代理需要客户使用多种不同的工作方式；而透明代理可实现路由器和代理的自动连接，非常像一个 PAT设备。这些代理有一个非常重要的优点，即在客户端不需要一些特殊的软件或配置。那么 PAT和透明代理又有什么不同呢？我们将在本章的后面详细讨论这个问题。这个内容的标题为“ 为什么代理服务器不是 NAT？ ”。

4.6.3 状态分组过滤器

随着代理的不断发展，它又被叫做 PFs。当在PFs中加入保持状态信息的能力后，它就变成了一个状态分组过滤器（ SPF）。例如，一个 SPF能够检测到一个 PORT命令的通过，并且仅允许返回到上面提到的端口，而不是让所有端口号都大于 1023的信息通过。它并不允许所有 ACK

位被设置的每个TCP分组通过，而仅允许与向外的分组相对应的信息通过。在向简单 PF增加这些强大的功能前，它仅能够记录很少的附加信息。

由于已经向SPFS增加了许多其他的功能，所以纯粹的 SPF是不存在的。我们将会很快地讨论到这个问题。现在我们使用一台 Cisco路由器来描述一个SPF的示例，它使用反向的存取列表。根据与其他存取列表行的匹配情况，这些存取列表有能力修改它们自身的内容。

4.6.4 带有重写功能的状态分组过滤器

前面对SPF的定义并不为广泛接受。尽管 SPF在中间使用了单词过滤器，但当人们讨论 SPF时，他们还想让这个设备能够修改通过它的分组。从理论上来讲，给 SPF增加这个能力能够对分组进行完全控制。

分组重写特征能够被放入到一个特定 NAT中的SPF引擎中。回忆一下 NAT的需求：重写分组的能力和记录信息的能力。用于实现 PAT的连接表与用于实现 SPF 的连接表非常相似。如果你能够实现SPF，则对PAT的实现就非常容易了，反过来也是一样。

目前市场上有许多基于 SPF的商业防火墙示例。为了支持这种技术，它们使用了许多不同的术语。市场上的领先产品 CheckPoint公司的防火墙Firewall-1就是基于SPF的。它在这里被叫做带状态的多层探测（ SMLZ）。另外一个通用示例是 Cisco公司的PIX防火墙。

我们不想更深入探讨有关 SPF的工作细节-如果你已经理解 PAT内部的细节，那么你将会理解SPF的工作过程。在执行一个 SPF功能时，它所要维护的表与 PAT要维护的表完全一致。一个SPF防火墙至少要完成 PAT所做的工作。理想情况下，在它上面还要加入较好的数据验证和内容过滤功能。

4.6.5 代理服务器与NAT的不同

此时，讨论代理与 NAT的不同是比较合适的。为了达到讨论的目的，假设所有的 NAT、DF和SPF的特点都城是相同的，透明代理有些不同，但在讨论时，这个代理也被当做传统的代理。

从高层来看，代理和 NAT在外表上看是相同的，它们都将许多内部机器隐藏到一个 IP地址后面，由于需要地址转换，它们两个都能修改通过的数据流。为了正确地处理复杂的协议，它们都保留状态信息。从结果上来看，它们是相同的，但实现的手段有很大的差别，从低层来看，设备（一个代理或 NAT设备）的内部处理分组的方法完全不同，最根本的不同是：对于 NAT，工作的基本单位是分组，对于代理，所有的工作都是基于数据流的。现在让我们从代理开始讨论它们的含义。

当一个服务器接到一个分组时，服务器首先要确定分组的目的地址是否为此服务器（例如，目的地址是这些地址之一）。传统的代理就是这样工作的。然后分组被传送到服务器的 IP堆栈中，如果分组属于一个已经存在的连接，分组的数据部分将被截取出来，放入一个缓冲区，以便代理程序能够读到它，如果是一个新连接，则建立一个新缓冲区，并且代理程序应被告知，

这是一个新连接服务。它们两个的处理是相同的。

当代理需要发送一些信息时，则要进行相反的处理。代理要发送的信息将被放到一个输出缓冲区中，在服务器上的TCP/IP软件将从缓冲区中取出信息，放入一个分组，然后将它发送出去。

使用IP协议，分组的大小将有所不同，由于底层处理的帧有最长限制，为了通过网络，大的分组将会被划分为几个分段，例如，一个以太网段的 MTU为1500字节，如果一个 2000字节的分组通过这个网络，则它致少要被划分成两部分，在一个代理服务器上， IP堆栈将这些小分段组合在一起，然后将这些数据放入缓冲区，理想情况下，这种分段不会发生。如果发生，主机不会传递没有被分段的分组。一个主机不应该总是判断通过网络的分组是否被分段，主机最好不传递比本地网络分组要大的分组。

讨论分段的目的就是要说明，进入代理服务器的分组数量不一定等于输出的分组数量，这里有一个最简单的示例，一个代理服务器接到包含有“ Hello World ”的一个分组。然而，当它被传送出去时，可能出现两个分组“ Hello ”和“ World ”，反向也一样。事实上，代理输入的仅为字符串，然后将它们输出到不同的缓冲区中。在这个过程中，内容可能会发生变化，代理并不关心分组是如何被分段的，当它检测到一个 FTP PORT命令时，它将这条命令读入，然后决定如何对它进行修改，最后输出修改后的内容。它并不关心命令的长短。

这与一个NAT设备恰恰相反，当一个 NAT设备的IP堆栈得到一个地址不是到达此设备的分组时，它将路由这个分组。这种情况是经常发生的。在路由过程中， NAT设备就有机会对分组进行操作，除了分段和两个特殊情况外， NAT保持分组的一进一出，并且大小相同。当一个PORT命令通过时， NAT设备将使分组尽可能简洁。这也就意味着要有一些特殊的代码扩充或缩减分组，以适合过长或过短的地址。当分段到达时，典型的 NAT设备将对这些分段进行装配。尽管分段也可能就是分组，但它们通常也是一个较长分组的一部分。虽然分段做为一个整体考虑，但还要对分组的进和出进行计数。

两种方法的安全性能怎么样呢？各有优缺点。受攻击的类型与一个分组的结构有关，对于代理来说，由于分组要被拆开，所以这种类型的攻击很难影响内部机器。但是，由于它也要处理到达它本身的分组，所以代理比内部机器更容易受到攻击。一个 NAT设备不容易受到上述同一类型的攻击，但这种攻击可能会被传送到内部主机上。值得庆幸的是，这种类型的攻击几乎都是拒绝式服务（Denial of Service:DOS）攻击。这些攻击将会摧毁一些情，但不会引起信息集成的冲突。在某些情况下，防火墙被摧毁，而另一些情况是防火墙还能执行正常的功能，但内部主机都已经不能工作，它们两个中没有一个是绝对好的，可根据防火墙管理员的爱好进行选择，没有人希望他们的防火墙不能工作。但是另一方面，它的工作就是要保护内部机器。

NAT和代理的另一个最大的不同是数据的验证和修改，有许多代理软件包有限地考虑了一些安全情况，这也就是说，代理中的协议是设计者认为有效的协议。它们有一些允许值，被高等的代理将负责查看这些值，如果需要，还要进行修改。在某些情况下，如果一个协议有一些不一致的问题，则代理不会为它们工作，这就使你使用此协议的用户感到很沮丧。

许多NAT软件包采用了一个不同的策略，它们对协议的通过要求很少，并且让尽可能多的协议通过。缺省时，NAT更加开放。这也就是说，如果想从内部到外部生成一个连接，并且不知道使用什么协议，NAT也能够将它转发出去。

从现在来看，这是一个不平的比较，我比较的是最好的代理与最坏的 NAT的实现。一般来说，产品来自于两个阵营，并满足折中原则。一个好的防火墙管理员能够使 NAT/SPF更加安全，一个不好的防火墙管理员经常会错误地配置一个好的代理，目前的趋势是：如果想让协议工作的话（象FTP示例）NAT设备仅能向上到第4层，而代理部是工作在第4层之上。最简单的代理（电路线代理）应工作在第5层。一般来说有这么一种假设：堆栈走得越高系统就越安全。

上述的讨论带有宗教色彩，但你购买的是一个实际产品，而不是一个概念上的防火墙，要根据产品本身的特点来对它进行评价。

引起讨论的另一个因素是SPF和代理之间的界限非常模糊。大部分商业防火墙的最新版本中，不管来自于什么背景，它们包括来自于代理世界的特征，也包括来自于 SPF世界的特征。例如：Firewall-1防火墙。当一个NAT类型的分组通过时，大量的“安全服务器”可有选择地被激活，它们还包括一些特殊的功能，如牢记、剥离不希望要的内容（例如 Java或Active X），通过名子或UKL来阻塞特点的站点，许多代理防火墙都具不透明性，代理为了实现这种透明，它不得不在某种程序上改变它的行为。最简单的解释是当分组的目的地址不是代理时，代理不得不要执行一个SPF功能，将分组传送给代理软件。

4.6.6 SPF的缺点

尽管有关SPF缺点的讨论很多，但我们这里仅讨论安全性的内容，从性能上看，所有的产品都能工伯的很好。尽管它们之间存在着性能上的不同和管理上的不同，但是如果要个产品声明能支持某个特定协议，它一般都能正常工作。

由于信息在通过代理时，代理要做更多的工作，所以它的运行速度一般比较慢。它要做的工作包括要去掉分组的头，对它进行处理、分配端口以及大量的数据缓冲和拷贝。而 SPF却跳过了这些工作。它对通过的协议不做任何工作，这是它的一个优点。如果对协议处理的过细，这可能就会变成一个缺点，这与如下观点相一致，代理类型的软件比 NAT类型的软件更加容易处理复杂的协议。这也就是说要根据分组的内容来进行选择处理过程，至少 TIP是这样。从这两个软件包中各取优点将使它们间的界限更加模糊，防火墙的设计者能够为协议选择最好的工具。

代理中的透明选项是一个非常好的特征。这就不需要修改所有内部机器上的软件，并且也不要对这些改变的支持，这上一个非常好的优点，使用这种选择，就不能获得详细的信息了。

对于传统的代理，特别是每个协议都有一个程序的结构，对这些协议的操作永远不会有问题。例如，如果用户连接到Telnet服务器，则应使用Telnet协议，如果连接到HTTP代理，则应使用HTTP协议，如果你已经花费许多时间在 Web上进行浏览，你可能会注意到一些 URL上会说明一些不同的端口号，例如，不输入下面内容：

http://www.example.com

而输入以下内容：

http://www.example.com:8080

在这种情况下，我们明确地说明要通过端口 8080连接到Web服务器，而不是通过缺省HTTP协议的80号端口连接到服务器。对于传统的代理来讲，这并不是一个问题，它知道你想使用HTTP；它知道你想通过8080端口来存取。

此时代理要求客户明确地说明协议和端口地址，再让我们看一看透明代理的情况，客户没有做任何特殊的配置，用户甚至没有意识到防火墙的存在。现在，客户没有向代理说明使用哪个协议，因为它并不知道代理在什么地方。这样，当它连接在端口 80时，代理先进行一个假设-假设这就是HTTP协议，然后对它进行处理。如果浏览器上说明的端口是 8080，这又该如何处理呢？代理不得不又进行另一个假设。端口 8080一般被用做非标准的HTTP端口，但并不总是这样，代理必须选择HTTP代理或线路级代理。这是最常用的配置。

下面的情况将如何处理呢：

http://www.example.com:21

一些在Internet上爱开玩笑的人有时将Web服务器端口定义到21，用户不得不使用这个端口进行连接，代理又不得不做出一个假设 -假设这是一个FTP协议。此时，这种连接将不能正常地工作。

使用透明方式会使我们失去一些信息，此时我们不得不要让透明代理根据端口号对协议进行假设。这种方式大部分时间都能工作的很好，但也有些例外。 SPF也同样会遇到这种问题。

一些人认为，对于防火墙管理员来说， SPF类型的结构太容易受到攻击，但是， SPF却很灵活，并且很诱人。这种想法是不实际的，因为大部分代理也有相似的性能。

大部分防火墙都使用GUI界面来配置哪些规则允许，哪些规则不允许。这对于维护大量的规则集合是非常方便的。一些有经验的防火墙管理员经常报怨，这种方式会阻碍管理员对防火墙功能的理解。他们认为在复杂的产品上使用简单的操作，这将给没有经验的管理员一种错觉，即他们能够理解防火墙的所有内容，这也就是说，在安全问题上给人一种错觉。

4.7 小结

当分组通过一个网络地址转换（NAT）设备时，NAT将改变分组中的第三层地址。其他协议，如IPX也能被转换，但大多数商业NAT的实现都是针对IP地址的。一般来说，仅简单地修改第三层协议是不够的，通常在高层上的信息也要被修改。 NAT和安全通常被一起使用。

NAT的内部构想来自于早期的基于代理的防火墙解决方案。代理服务器允许管理员根据内容对流量进行过滤，然后过滤后的信息将到达外部网络。到达外部的每一项内容都来自于一个IP地址。

代理管理员通常要配置一个过滤路由器（例如，一个分组过滤器）以防止从内部到外部以

及从外部到内部的直接存取。这种配置仅允许内部机器直接与代理进行通信。如果内部机器想存取外部网络，则内部机器必须使用代理。所有流量通过网络中（至少通向 Internet）的那个点通常被叫做阻塞点。为了尽可能安全，要仔细配置代理服务器。

一个代理防火墙另一方面的效果就是从外部只能看到一个 IP地址。这将把所需公共的、可路由的IP地址减少到一个。RFC1918认识到了这一点，并且给出了一系列地址范围供代理服务器或NAT防火墙内部使用。

NAT的类型有多种。第一种类型就是静态 NAT，它能够实现两个 IP地址之间的一对一的映射。在一个方向上，源地址或目地地址要被转换；在另一个方向，过程相似，内容相反。一般来说，源地址都要被转换，但有时也对目的地址进行转换。一种转换目的地址的示例就是在不对客户机器进行重新配置的情况下，将客户机器重定向到不同的服务器。

为了知道什么时间转换以及是否要转换源或目的地址， NAT必须要区别两个网络接口。通常情况下它们被叫做“内部”和“外部”。由于是一对一的映射，所以静态 NAT没有节省地址空间。

一种静态NAT的变化被叫做双 NAT，它将同时转换源地址和目的地址。它在连接具有两个相同地址的网络时是非常有用的。

一个静态NAT在实现时，仅简单地转换第三层地址，而根本不改变数据流内容。这种实现对于某些协议会产生一些问题。一个典型的协议就是 FTP协议，它的IP地址存在数据流中，为了使FTP也能正常工作，一个静态 NAT必须在FTP PORT命令经过时对它进行修改。如果 PORT命令被划分成几个分组，它也应该能够正常工作。

另外一类NAT是动态NAT。动态NAT虽然与静态NAT相似，但动态NAT能够实现多对多或多对一的映射。在从一个地址池中取地址时，可实现静态映射。如果内部地址多于外部地址，那么就会引起地址等用问题。为了解决这个问题， NAT设备将试图检测映射结束的时间。具体的实现策略将包括使用定时器以及检测代表连接结束的分组。

为了记录这些内容，动态 NAT必须维护一个连接表。它记录了 IP地址、端口号、 FIN位和计时器。尽管使用了这些机制，但动态 NAT还会引起资源的竞争，并且内部机器还不能到达外部，这就需要有一个更好的方法。

端口地址转换（ PAT）也是NAT的一种，它允许多台内部机器同时共享一个外部 IP地址。与转换IP地址相类似，它是通过转换端口地址来实现的。当一台内部机器要连接到外部时，它的源口号和地址都要被转换。 NAT路由器将记住使用了哪个源端口，这样当要选择一个新的源端口时就不会产生冲突。 PAT最终能够实现地址的节省以及达到某种程度的安全性。

PAT使用的连接表与动态 NAT使用的连接表非常相似。 PAT能够按需要动态打开端口，以便能够处理反向连接上的协议，如 FTP。大部分存在的 NAT解决方案都是基于 PAT的，并且能够按要求实现静态NAT。

NAT的主要特征就是节约地址空间。另外，当某种类型的网络出现问题时，它还能暂时地

工作。NAT对性能的影响不是很大。除了网络负载很重以外，这种影响都被忽略。

尽管NAT、代理、防火墙经常在一起使用，但代理和防火墙要完成的使命与 NAT有所不同。防火墙主要针对安全 -在这里安全指的是控制网络的连接。从整个发展过程来看，防火墙有如下几种类型：代理、分组过滤器（ PF）以及静态分组过滤器（ DPF)。

通过让客户连接到代理，而不直接连接到最终的服务器上，这样代理才能工作。代理将索引出希望使用的内容，并且将它们送到内部客户。与 NAT一样，为了能够正确地处理协议，代理必须要了解所通过的协议。

PF通常被用在与代理的连接上，以达到保护的目的。建立一个抑制点，以便让所有的流量都经过代理。分组过滤器并不维护状态，并且能够提供很大的端口范围，以适应多种协议，如FTP。像NAT一样，PF通常也可被叫做路由器。

SPF是带有状态的PF。另外，几乎所有的SPF都能够需要重写分组。如果一个 SPF能够重写分组的话，理论上讲它也能对分组进行任何操作，这包括实现 NAT。PAT所需要的连接表与SPF所需要的连接表是一样的。

NAT（和SPF）与代理的根本不同是在它们的实现上。对于 NAT，基本的工作单位是整个分组；对于一个代理，它针对的是一个数据流。在实际操作中，代理可能要把一个分组分成几部分，使其变成多个分组；而一个 NAT设备永远保持相同数量的分组进和分组出。

目前，市场上的大部分防火墙都是代理技术和 SPF技术的混合产品，这些产品的主要优点是它们具有很好的透明性，在内部客户机组上不需要特殊的软件或配置。

4.8 常见问题解答

问题：为什么有些程序不能工作？

解答：如果你正在负责管理一个防火墙、 NAT设备、代理或类似的设备，你不可避免地总要问下面的问题：“我刚刚下载一个新的媒体流协议的 beta版本，叫“ Foo ”，但它不能正常运行，如何修复它呢？”不论是什么原因，媒体流协议总要使用反向连接。为了能让它正常工作，试一试下面的内容。

- 访问一下销售这个程序的销售商 Web站点。他们通常会维护一个 FAQ，这样就可以知道如何让它们的协议同防火墙一起工作。在某些情况下，你只需要在客户程序上做一些简单的设置就可以了。而在有些情况下，将会有一个使用说明来指导你如何了解你的防火墙。
- 询问一下防火墙的销售商，看一看是否有升级的版本能够处理这个协议。大部分防火墙销售商都有一个可进行内容查询的 Web站点。你能够查到有关这个协议的问题。
- 检查一下防火墙的日志文件，看一看是否有正在返回的并且被拒绝的反向连接。一般情况下，你可以使用一个协议分析器，来确定哪些协议能够正常工作。
- 不要忘记考虑你也许不希望使用这个协议。如果你对安全问题很敏感，则你应充分认识

到一些新程序可能会存在某种缺陷，这些缺陷可能会威胁到你的网络。目前，许多客户端软件都存在着很大漏洞。

问题：为什么我不能连接到任何设备？

解答：当你第一次设置你的NAT防火墙/代理时，经常会发生这种问题。不能够实现连接的原因很多，任何一个原因都会造成这种情况。下面列出一些要特别注意的事项。

- 首先确保你的路由是正确的，为了验证分组是否能够通过，如果可能的话，可临时关闭NAT或一些安全特征。如果不是这方面的问题，那么也许是路由器问题；如果是这方面的问题，则需要你检查安全配置。
- 确保你要发送的信息能够通过。这听起来很明显，但问题还是会经常发生。判断这个问题的最简单的方法是查看LOG文件。如果显示已经结束，则说明你要发送的信息没被允许通过。
- 确保正确的一些ARP设置。有些方案需要虚拟的IP地址，此时你不得不手动发布ARP地址，一种快速的检查方法是查看路由器上的ARP表。
- 确保你的客户配置是正确的。如果你正在使用代理的话，则这项内容是非常有用的。首先将客户程序设置成使用代理，然后查看容易忽略的内容或错误配置。
- 如果上述的工作全部失败，则不得不使用一个协议分析器来查看线路上发生了什么。为了获取全部情况（防火墙的内部、外部等），你不得不在许多地方使用这个分析器。

问题：我如何验证我们地址已经被正确转换？

解答：这个问题非常容易解决。最简单的方法是连接到这样一种设备，它能够告诉你所使用的IP。如果这是你自己的网络，则你的在NAT外端的路由器能够显示的这些地址。例如，如果你连接到一个Cisco路由器，且输入了“show users”命令，此时路由器将会告诉你正在连接的DNS的名子或IP地址。

如果你是一个终端用户，并且怀疑地址正在被转换，要想知道此时的详细情况就会有些难度。如果你已经从Internet上的路由器或UNIX上获得了一个帐号，你通常能够找到详细的情况。另外一种选择是使用一个Web页，它会通知你IP来自于何方。请看下面这个例子。

http://www.anonymizer/3.0/snoop.cgi

问题：什么样的防火墙结构最好？

解答：这又是一个类似于宗教上的问题，很难解答。但这里也有一些被人们普遍接受的实践经验。现在我们拿一个中型公司做一个示例。假设它们要进行全天后的Internet连接，并且有自已的Web服务器和E-Mail服务器。假设以前他们没有防火墙，现在想安装一个。

外部的Internet应该能够到达公司的Web服务器和E-Mail服务器-这也是他们的目的，同时内部机器也应能够到达这两个服务器。这台服务器要尽量安全，以防止从Internet上的入侵。典型的设备是使用一个带DMZ的防火墙，这个防火墙有时也被叫做3路或3口的防火墙。具体的连接见图4-16

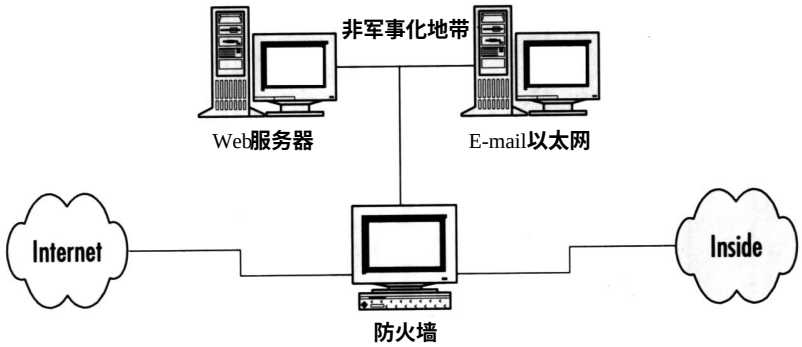

图4-16 带有DMZ的透明防火墙

在这个示例中，防火墙带有路由功能。它可能是一个 SPF防火墙，也可能是一个透明代理。当一个内部用户希望访问外部时，信息必须穿过防火墙。不论是内部，还是外部，只要想连接到公共服务器，那么都必须穿过防火墙。尽管在上面的图中没有说明这一点，但这种类型防火墙上的规则会禁止 Internet直接连接到内部网络中。一般来说，内部地址应使用 RFC1918文档中所说明的地址。对于内部机器而言，防火墙将使用 PAT技术。

出于管理的目的，在对防火墙进行规则设置时，一般至少要允许一些内容机器在较高的水平上能够访问公共的服务器。

采用这种结构类型的结果会造成内部机器不能够完全信任 DMZ。这也就是说，DMZ机器上的信息不能返回到内部机器上，至少不能到达所有的端口。这也说明如果 DMZ机器设置的比较恰当的话，内部机器将会得到很好的保护。

4.9 参考信息

由于本章不可能覆盖 NAT、代理、防火墙的全部细节，所以我们在这里提供了一些资源，以便读者进行索引。一些内容是普通的，如 RFC文档；一些内容是很特殊的，如 Cisco公司有关NAT软件的Web页面。一般情况下，希望你扫描一下这个列表，然后查询你感兴趣的话题。如果你已经计划实现在这里所提到的某种技术，你就需要阅读相关的文档。

1. RFC文档

有关RFC1918文档的网址是：

http://www.cis.ohio-state.edu/htbin/rfc/rfc1918.html

RFC1918文档覆盖了私有地址空间问题以及 NAT技术。有关私有地址范围（ 10.x.x.x,176.16.xx-172.31.x.x,192.168.x.x）的官方文档也在这个 RFC中。文档的顶部有一些连接，内容包括相关文档以及一些旧的 RFC。

下面的连接是一个相关的 RFC文档，但它不在RFC1918的索引中。

http://www.cis.ohio-state.edu/htbin/rfc/rfc1631.html

使用它的主要对象是开发人员和实现人员。

2. IP Masquerade/Linux

http://ipmasq.cjb.net/

这是查看IP Masquerade文档的主要网址。在这个网页上，你将会发现一个变化日志，以及一个到HOWTO的连接。

http://member.home.net/ipmasq/ipmasq-HOWTO.html

这个页面将说明如何加入 IP Masquerade邮件列表以及获得 IP Masquerade处理器软件的位置。在写这本书时，到本页的连接不知什么原因被中断（在你查看时，也许已经被恢复），但下面这些网址还能正常工作：

http://www.tsmservices.com/masq/

http://www.rustcorp.com/linux/ipchains/

这是有关IPChains软件的信息。它需要同 IP Masquerade软件一起工作。

3. Cisco

Cisco有一些文档是用来说明它们的路由器是如何实现 NAT的。如果你想使用 NAT，你自己至少要熟悉它。

http://www.cisco.com/warp/public/458/41.htmp

这个网址是Cisco NAT常见问题解答

http://www.cisco.com/warp/public/701/60.html

这个网址是 Cisco NAT的技术提示。这里会说明 Cisco文档包含哪些协议，不包含哪些协议。

http://www.cisco.com/warp/public/cc/sol/mkt/ent/ndsgn/natl-wp.htm

这是Cisco NAT的白皮书。它与我们本章讨论的技术水平相类似，但明显偏向于 Cisco产品。其中包括配置示例以及这里没有讲到的一些特征，如 TCP装载平衡等。

4. Windows

http://www.uq.net.au/~zzdmacka/the-nat-page/nat-windows.html

这里一个优秀的、基于 Windows的NAT产品列表。事实上，还有一些讲述 NAT网络的站点值得去查看，但这里没有列出。

我喜欢的、价位较低的 Windows NAT产品是来自于 Syberge Networks公司的 SyGate。它很便宜，并且容易进行设置，你可以得到它的测试版进行评估，看下列网址：

http://www.sygate.com/

Microsoft代理服务器已经被提到了几次，有关它的信息请看下面网址：

http://www.microsoft.com/proxy/default.asp

如果你想真的运行它，你应该查看一下 MSProxy FAQ:

http://proxyfag.networkgotls.com/

5. NAT白皮书

这里是两个独立的NAT白皮书/资源

http://www.alumni.caltech.edu/~dank/peer-nat.html

它主要针对端到端网络和NAT

http://www.kfu.com/~dwh/nat-wp.html

它主要强调有关RFC1918没有谈到的一些驱动问题。@work维护着一个NAT FAQ。这些内容主要针对它们的用户，但也包含有一些有用的信息和定义。

http://work.home.net/whitepapers/natfaq.html

6. 防火墙

下面是几个有关防火墙的FAQ

http://www.clark.net/pub/mjr/pubs/fwfaq/

这个网址非常好，它很完整，并且讲得也很基础。

http://www.waterw.com/~manowar/vendor.html

这是一个很好的有关防火墙信息的集合。它们以比较表格方式出现。

fpt://ftp.gratcirde.com/pub/firewalls/welcome.html

这个网址包含有防火墙的邮件列表以及相应文档。

http://www.nfr.net/firewall-wizards/

这里有一些关于Firewall-1防火墙的FAQ

http://www.phoneboy.com/fwl/

phoneboy公司的FAQ。讲述有关Firewall-1的内容。

http://www.dreamwvr.com/bastions/Fwl_faq.html

这是第二个FW-1的FAQ。你也许更喜欢它们的公司。

http://www2.checkpoint.com/-joe