第 5 章　防火墙、入侵检测与蜜罐技术

防火墙、入侵检测和蜜罐是防御攻击的重要屏障，各有其独到之处。本章介绍防火墙中的包过滤型、应用代理型、状态检测型防火墙的技术原理与应用，涉及 Linux 的防火墙工具 iptables、Windows 的自带防火墙。同时也介绍了入侵检测和蜜罐的技术特点与应用。

5.1　防火墙

5.1.1　防火墙定义

防火墙是一个位于不同网络或者网络安全域（如内部网络和外部网络、专用网络和公共网络）之间的软件或硬件，用来在不同网络之间构造屏障，阻止对信息资源的非法访问。防火墙通过隔离控制机制，对所有流经的网络通信进行检查，可以防止大部分攻击，避免其非法操作在目标计算机上被执行。同时，防火墙还可以关闭指定端口，禁止特定协议的通信，从而阻断部分木马的网络连接。

5.1.2　防火墙类型

防火墙技术的发展经历了包过滤、应用代理网关、状态检测 3 个阶段，因而可以将防火墙分为包过滤型防火墙、应用代理型防火墙和状态检测型防火墙。其中，包过滤型防火墙以以色列的 Check Point 防火墙和美国 Cisco 公司的 PIX 防火墙为代表，应用代理型防火墙以美国 NAI 公司的 Gauntlet 防火墙为代表。

防火墙在网络中的位置一般如图 5-1 所示。

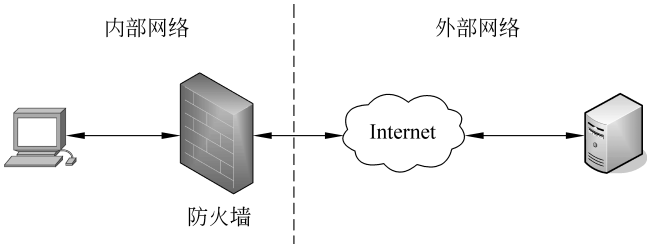

图 5-1　防火墙在网络中的位置

1. 包过滤型防火墙

包过滤型防火墙工作在 OSI 参考模型的网络层以及传输层，它能识别和控制数据包的源 IP 地址和目的 IP 地址，但是在传输层，只能实现判断数据包是 TCP 数据包还是 UDP 数据包及该数据包所用端口的相关信息，以决定是否允许这些包通过防火墙。防火墙没有任何关于活动连接的信息，所以每次收到包都要独立决定是否允许其通过。

由于包过滤型防火墙只分析 IP 地址、使用的 TCP/UDP 以及所用端口，因此这类防火墙具有较快的处理速度，并且容易配置。

包过滤型防火墙是防火墙技术中最基础的技术。包过滤防火墙主要存在以下不足：不能有效防范黑客的攻击；无法分析应用层协议；无法处理新出现的安全威胁。缓冲区溢出型木马和高端口监听木马能够轻易突破这种防火墙的限制。

2. 应用代理型防火墙

应用代理型防火墙工作在应用层，其实质是一个运行代理服务的网关。它能够彻底阻断内部网络和外部网络的直接通信，内部网络用户对外部网络的访问转变为防火墙访问外部网络，再由防火墙转发给内部网络用户。所有的通信都必须经过应用层代理软件的转发，任何时候访问者都不能与服务器直接建立 TCP 连接，并且应用层的协议会话过程必须符合代理的安全策略的要求。

应用代理型防火墙最大的优点是可以检查应用层、传输层和网络层的协议特征，对数据包的检测能力比较强，通常包含高级应用检测能力，允许防火墙检测应用层攻击，例如缓冲区溢出攻击和 SQL 注入攻击。所以它的安全性比包过滤型防火墙更高。其缺点主要表现为以下两点：应用代理型防火墙难以配置；处理速度非常慢。

3. 状态检测型防火墙

包过滤型防火墙和应用代理型防火墙是通过对数据包的 IP 地址、端口等参数进行检查来实现的，而忽略了数据包在传输过程中的连接状态的变化。网络通信都必须遵循 TCP/IP，根据 TCP，每个可靠连接的建立需要经过三次握手。这说明数据包在传输过程中前后的状态变化有着非常密切的联系，状态检测技术就是针对数据包传输过程中的状态变化而提出的。在实现中把进出网络的数据当作一个个会话，为每个会话建立状态表，在该表中记录会话的状态变化。这类防火墙不仅根据规则对数据包完成检查，而且还要考虑被检查数据包是否符合其对应会话所处的状态。当有一个新的包到达防火墙时，过滤机制首先检查这个包是否是当前活动连接（前面已经授权过的）的一部分。只有当这个包没有出现在当前的活动连接列表里时，防火墙才会以它的过滤规则评估这个包。由此可见，在传输层上状态检测型防火墙能提供较为有效的控制能力。

状态检测型防火墙的优点是效率与性价比较高，广泛适用于保护网络的边界。

5.1.3 包过滤技术

包过滤的原理是使用者建立的安全模式和规则，过滤从防火墙经过的被认为是不安全的数据包。对数据包的安全选择的依据是系统内部事先设置的过滤规则（又称安全规则库）。一个包过滤防火墙通常是通过对数据包的 IP 包头、TCP 包头或 UDP 包头的检查实现的，主要信息如下。

（1）源 IP 地址和目的 IP 地址。通过对 IP 地址的过滤，可以阻止与特定网络（或主机）的不安全连接。

（2）源 TCP/UDP 端口和目的 TCP/UDP 端口。通过对端口的过滤，可以阻止与特定应用程序的连接。

（3）ICMP 消息类型。

（4）TCP 包头中的 ACK 位。

（5）TCP 链路状态。

TCP 链路状态是指地址、端口等因素的组合。为了实现其可靠性，每个 TCP 连接都要

先经过一个握手过程交换连接参数。每个发送出去的包在后续的其他包被发送出去之前必须获得一个确认响应。但并不是对每个 TCP 包都非要采用专门的 ACK 包响应，实际上仅仅在 TCP 包头上设置一个专门的位就可以完成这个功能。因而只要产生了响应包就要设置 ACK 位。连接会话的第一个包不用于确认，也就没有设置 ACK 位；后续会话交换的 TCP 包就要设置 ACK 位。

包过滤是在网络中适当的位置对数据包实施有选择的放行。为系统内设置的过滤规则通常称为访问控制表(Access Control List，ACL)，只有满足过滤规则的数据包才被转发至相应的网络接口，否则将被丢弃。

包过滤型防火墙在本地端接收数据包时，一般不保留上下文，只根据目前数据包的内容决定。根据不同的防火墙类型，包过滤可能在进入、输出时或这两个时刻都进行。可以拟定一个要接受的设备和服务的清单以及一个拒绝的设备和服务的清单，组成访问控制表。在主机或网络级容易用包过滤技术接受或拒绝访问。例如，可以允许主机 A 和主机 B 之间的任何 IP 访问，或者拒绝除 A 外的任何设备对 B 的访问。

包过滤实际上通过建立一个可靠的、简单的规则集创建一个被防火墙所隔离的更安全的网络环境。创建时尽量保持规则集精简。规则越多，就越可能犯错误；规则越少，理解和维护就越容易。规则少意味着只分析少数的规则，防火墙的 CPU 周期就短，其效率就可以提高。当要从很多规则入手时，就要认真检查整个安全体系结构，而不仅仅是防火墙。每个防火墙规则都有一个默认的策略和一组对特定消息类型响应的动作集，每个包依次在访问控制表中对每条规则进行检查，直到找到匹配的规则。

防火墙有两种基本的安全策略：没有被列为允许访问的服务都是被禁止的；没有被列为禁止访问的服务都是被允许的。包过滤的设置必须遵循如下规则。

(1) 必须明确什么是应该被允许的和不应该被允许的，即必须制定一个安全策略。

(2) 必须正式规定允许的包类型、包字段的逻辑表达式。

(3) 必须用防火墙支持的语法重写逻辑表达式。

为了说明这个问题，以一个简单的按源 IP 地址数据包过滤方式为例。假设网络 202.101.x.0 是一个危险的网络，可以用源 IP 地址过滤禁止内部主机和该网络进行通信。当数据包经过防火墙时，防火墙检查数据包，根据规则决定是否允许该包通过。就源 IP 地址过滤而言，防火墙只要检查目的 IP 地址和源 IP 地址就可以了。表 5-1 是根据上面的要求制定的规则。

表 5-1　源 IP 地址数据包过滤的规则

规　则	方　向	源 IP 地址	目的 IP 地址	动　作
A	出	内部网络	202.101.x.0	拒绝
B	入	202.101.x.0	内部网络	拒绝

源 IP 地址数据包过滤方式没有利用数据包的全部信息，难以满足防火墙的需求。一种更好的过滤方式是按服务过滤。

假设安全策略是禁止外部主机访问内部的 E-mail 服务器(属于 SMTP，端口号为 25)，允许内部主机访问外部主机，实现这种过滤的访问控制规则与表 5-2 类似。

表 5-2　按服务过滤的规则

规则	方向	动作	源 IP 地址	源端口	目的 IP 地址	目的端口	注释
A	入	拒绝	*	*	E-mail	25	不信任
B	出	允许	*	*	*	*	允许连接
C	双向	拒绝	*	*	*	*	默认状态

规则按从前到后的顺序匹配，* 代表任意值，没有被过滤规则明确允许的包将被拒绝。也就是说，每个规则集都跟随一条隐含的规则，例如表 5-2 中的规则 C 就是这样。这与一般原则是一致的，即没有明确允许就被禁止。

任何一种协议都是建立在双方的基础上的，信息流也是双向的，所以在考虑允许内部用户访问 Internet 时，必须允许数据包不但可以出站而且可以入站。同理，若禁止一种服务，也必须从出站和入站两方面制定规则，规则总是成对出现的。

当防火墙进行包过滤时，首先，假设处于一个 C 类网 116.101.y.0 中，认为网站 202.101.x.3 上有不安全的 BBS，希望阻止网络中的用户访问该网站的 BBS。再假设这个网站的 BBS 服务是通过 Telnet 方式提供的，因而需要阻止往该网站的出站 Telnet 服务。允许内部网用户通过 Telnet 方式访问 Internet 的其他网站，但不允许其他网站以 Telnet 方式访问网络。此外，为了收发电子邮件，允许 SMTP 出/入站服务，邮件服务器的 IP 地址为 116.101.y.1。最后，对于 WWW 服务，允许内部网用户访问 Internet 上任何网络和网站，但只允许一个公司的网络访问内部 WWW 服务器，内部 WWW 服务器的 IP 地址为 116.101.y.5，该公司的网络为 98.120.z.0。根据上面的策略安排，可以得到如表 5-3 所示的规则。

表 5-3　过滤规则示例

规则	方向	源 IP 地址	目的 IP 地址	协议	源端口	目的端口	ACK 位设置	动作
A	出	116.101.y.0	202.101.x.3	TCP	＞1023	23	任意	拒绝
B	入	202.108.x.6	116.101.y.0	TCP	23	＞1023	置位	任意
C	出	116.101.y.0	任意	TCP	＞1023	23	任意	允许
D	入	任意	116.101.y.0	TCP	23	＞1023	置位	允许
E	出	116.101.y.1	任意	TCP	＞1023	25	任意	允许
F	入	任意	116.101.y.1	TCP	25	＞1023	置位	允许
G	入	任意	116.101.y.1	TCP	＞1023	25	任意	允许
H	出	116.101.y.1	任意	TCP	25	＞1023	任意	允许
I	出	116.101.y.0	任意	TCP	＞1023	80	任意	允许
J	入	任意	116.101.y.0	TCP	80	＞1023	置位	允许
K	入	98.120.7.0	116.101.y.5	TCP	＞1023	80	任意	允许
L	出	116.101.y.5	98.120.z.0	TCP	80	＞1023	任意	允许
M	双向	任意	任意	任意			任意	任意

规则 A、B 用来阻止内部主机以 Telnet 服务形式连接到网站 202.101.x.6，规则 C、D 允许内部主机以 Telnet 方式访问 Internet 上的任何主机。在设置规则时，规则的次序非常关键。防火墙实施规则的特点是，当防火墙找到匹配的规则后就不再向下应用其他的规则，所以当内部网主机访问网站 202.101.x.6，并试图通过 Telnet 建立连接时，这个连接请求会被规则 A 阻塞，因为规则 A 正好与之相匹配。规则 B 用来限制网站 202.101.x.6 的 Telnet 服务的返回包。事实上，内部主机试图建立 Telnet 连接时就会被阻塞，一般不会存在返回包。但用户如果通过其他办法使连接成功，则规则 B 将起作用。当用户以 Telnet 方式访问除 202.108.x.6 之外的其他网站时，规则 A、B 不匹配，所以应用规则 C、D，内部主机被允许建立连接，返回包也被允许入站。

规则 E、F 用于允许出站的 SMTP 服务，规则 G、H 用于允许入站的 SMTP 服务。表 5-3 中目的端口一栏中的 25 是 SMTP 的服务端口。

规则 I、J 用于允许出站的 WWW 服务，规则 K、L 用于允许网络 98.120.z.0 的主机访问本网络 WWW 服务器。

规则 M 是默认项，它实现的准则是没有明确允许就被禁止。

防火墙对网络通信的访问控制一般都是通过访问控制表实现的，其形式是类似如下的一些规则。

(1) accept from 源 IP 地址，源端口 to 目的 IP 地址，目的端口(动作)。

(2) deny from 源 IP 地址，源端口 to 目的 IP 地址，目的端口(动作)。

(3) nat from 源 IP 地址，源端口 to 目的 IP 地址，目的端口(动作)。

规则(1)表示防火墙允许指定的源 IP 地址和源端口到目的 IP 地址和目的端口的网络通信；规则(2)则相反，拒绝指定的源 IP 地址和源端口到目的 IP 地址和目的端口的网络通信；规则(3)表示允许地址转换。防火墙在网络层(包括其下的数据链路层)接收到数据包后，就在以上的访问控制表中进行逐条匹配，如果符合，就执行相应动作(例如丢弃数据包)。

5.1.4 应用代理技术

代理技术是最常用的防火墙技术之一，通过对防火墙代理服务的配置可以使局域网内部的主机通过一台代理服务器访问外网，也可以使外网对内网的访问受到防火墙的限制。安装了代理软件的主机即为应用代理型防火墙主机。

代理服务器是介于浏览器和 Web 服务器之间的服务器。有了代理服务器之后，浏览器发出的信息会先送到代理服务器，由代理服务器取回网页内容并传送给客户的浏览器。对企业网络而言，代理服务器可以起到控制网络访问、屏蔽不安全信息以及网络加速的目的。

Squid 是 Linux 下缓存 Internet 数据的代理服务器软件，是一个应用层代理服务器，能和 iptables 配合建立透明代理服务器。

Squid 接收用户的下载申请并自动处理下载的数据。当用户要下载一个主页时，向 Squid 发出一个申请，让 Squid 代替其进行下载。然后 Squid 连接相应的网站并请求该主页，接着把该主页传给用户，同时保留一个备份。当别的用户申请同样的页面时，Squid 把保存的备份传给用户，使用户觉得速度相当快。代理服务流程如图 5-2 所示。

代理服务具体过程如下。

(1) 客户端向代理服务器发送 Web 访问请求。

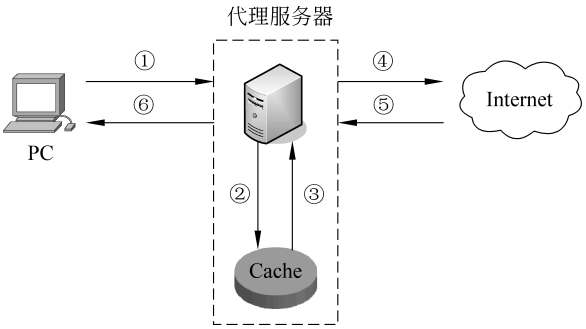

图 5-2　代理服务流程

(2) 代理服务器接收到请求后,首先判断是否满足访问控制列表的规则。如果满足,则在缓存中查找是否有客户端所需要的信息。

(3) 如果缓存中有客户端需要的信息,则将信息传送给代理服务器。

(4) 如果没有,代理服务器就代替客户端向 Internet 请求指定的信息。

(5) Internet 上的主机将代理服务器的请求信息发送给代理服务器,同时代理服务器会将信息存入缓存中。

(6) 代理服务器将 Internet 上的回应信息传送给客户端。

Squid 可以代理 HTTP、FTP、Gopher、SSL 和 WAIS 等协议,并且 Squid 可以自动地进行处理,通过访问控制特性灵活地控制用户访问时间、站点等限制,根据需要过滤数据包。

在此过程中,合理使用访问控制是非常重要的。使用访问控制特性,可以控制其在访问时根据特定的时间间隔进行缓存、访问特定站点或一组站点等。Squid 访问控制有两个要素:ACL 元素和访问控制列表。

ACL 元素是 Squid 访问控制的基础,其基本语法如下:

```
acl name type value1 …
acl name type "file"…
```

其中,name 为 ACL 元素的名字,在书写访问控制列表时需要引用它们;type 可以是在 ACL 中定义的任意类型;每个 ACL 元素可以有多个值,当进行匹配检测的时候,多个值由逻辑或运算连接,即,任一 ACL 元素的任一值被匹配,则这个 ACL 元素即被匹配。

例如:

```
acl http_ports port 80 8000 8080
```

可以匹配 80、8000、8080 这 3 个端口中的任意一个。

```
acl clients src 192. 168. 0. 0/24 10. 0. 1. 0/24
```

使用了子网 192. 168. 0. 0 和 10. 0. 1. 0。

```
acl guests src "/etc/squid/guest"
```

允许文件/etc/squid/guest 列出的客户机访问代理服务器。其中,文件/etc/squid/guest 中的内容可以如下:

```
172.18.18.3/24
```

```
222.111.24.8/16
10.0.1.24/25
```

不同类型的 ACL 元素写在不同行中。当一个 ACL 元素的值较多，不方便全部列出的时候，可以使用文件为 ACL 元素指定值，该文件的格式为每行包含一个条目。

ACL 的类型较多，有 src 类型、dstdomain 类型、port 类型、time 类型等。

ACL 元素是建立访问控制的第一步。第二步是访问控制列表，用来允许或拒绝某些动作。

访问控制列表的语法如下：

```
access_list allow|deny [!] aclname …
```

例如：

```
http_access allow MyClients
http_access deny !Safe_Ports
```

Squid 有大量访问控制列表，其中 http_access 是最重要也最常用的访问控制列表。它决定哪些用户的 HTTP 请求被允许或被拒绝。当读取配置文件时，Squid 只扫描一遍访问控制行，因此访问控制列表规则的顺序也非常重要，而且规则总是遵循由上而下的顺序。根据访问控制列表允许或禁止某一类用户访问，如果最后一条为允许，则默认就是禁止。通常应该把最后的条目设为 deny all 或 allow all 以避免安全隐患。

访问控制列表的规则按照它们的顺序进行匹配检测，一旦检测到匹配的规则，匹配检测就立即结束。访问控制列表可以由多条规则组成，如果没有任何规则与访问请求匹配，默认动作将与列表中最后一条规则对应。一个访问条目中的所有元素将用逻辑与运算连接，多个 http_access 声明间用或运算连接。

例如：

```
http_access Action 声明 1 AND 声明 2 OR http_access Action 声明 3
```

下面给出使用这些访问控制方法的实例。

(1) 允许列表中的机器访问 Internet。

假设规则：只允许 IP 地址为 192.168.0.10、192.168.0.20 及 192.168.0.30 的客户机访问 Internet，除此之外的客户机将拒绝访问本地代理服务器。规则如下：

```
acl allowed_clients src 192.168.0.10 192.168.0.20 192.168.0.30
http_access allow allowed_clients
http_access deny !allowed_clients
```

(2) 限制访问时段。

假设规则：允许子网 192.168.0.1 中的所有客户机在周一到周五的上午 10:00 到下午 4:00 访问 Internet。规则如下：

```
acl allowed_clients src 192.168.0.1/255.255.255.0
acl regular_days time MTWHF 10:00-16:00
http_access allow allowed_clients regular_days
http_access deny !allowed_clients
```

(3) 屏蔽含有某些特定字词的网站。

假设规则：使用正则表达式，拒绝客户机通过代理服务器访问包含诸如 sexy 等关键字的网站。规则如下：

```
acl deny_url url_regex -i sexy
http_access deny deny_url
```

(4) 禁止网段 172.16.1.10～172.16.1.50 上网。规则如下：

```
acl client src 172.16.1.10-172.16.1.50/32
http_access deny client
```

Squid 服务器的主配置文件 squid.conf 保存在/etc/squid 目录中，其提供代理服务的默认端口是 3128。

Squid 代理服务器访问控制策略功能丰富，ACL 类型和访问控制列表众多，在实际应用中可根据不同需求灵活进行配置。

实验 5-1　应用代理型防火墙应用实验

【实验目的】

(1) 理解应用代理型防火墙的技术原理。

(2) 熟练掌握 Squid 的安装和配置、ACL 命令及规则。

【实验原理】

应用代理型防火墙技术是在网关计算机上运行应用代理程序，运行时由两部分连接构成：一部分是应用网关同内网用户计算机建立的连接，另一部分是代替原来的客户端程序与服务器建立的连接。通过代理服务，内网用户可以通过应用网关安全地使用 Internet 服务，而对于非法用户的请求将予拒绝。代理服务技术与包过滤技术的不同之处在于内网和外网之间不存在直接连接，同时提供审计和日志服务。

在 Linux 环境下，一般采用 netfilter/iptables 构筑防火墙，代理采用 Squid。Squid 是一款在 Linux 系统下使用的优秀的代理服务器软件。Squid 是一个缓存 Internet 数据的软件，它接收用户的下载申请，并自动处理所下载的数据。也就是说，当一个用户要下载一个主页时，它向 Squid 发出一个申请，要 Squid 替它下载；然后 Squid 连接相应的网站并请求该主页，接着把该主页传给用户，同时保留一个备份，当别的用户申请同样的页面时，Squid 把保存的备份立即传给用户，因而速度较快。

对于 Web 用户来说，Squid 是一个高性能的代理缓存服务器，可以加快内网浏览 Internet 的速度，提高客户机的访问命中率。

Squid 控制用户的访问权限等功能是使用 Squid 的访问控制特性实现的。Squid 访问控制有两个要素：ACL 元素和访问控制列表。访问控制列表可以允许或拒绝某些用户对特定服务的访问。每个 ACL 元素由列表值组成。当进行匹配检测时，多个值由逻辑或运算连接，即任一 ACL 元素的值被匹配，则这个 ACL 元素即被匹配。可以使用许多不同的 ACL 元素，不同的 ACL 元素写在不同行中，Squid 将把它们组合在一个访问控制列表中。

【实验拓扑】

实验拓扑如图 5-3 所示。

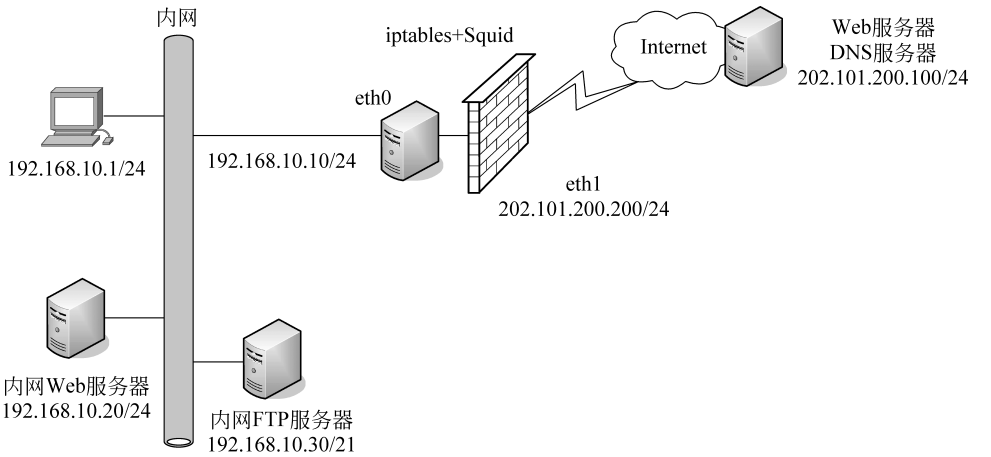

图 5-3　实验拓扑

【实验要求】

在图 5-3 所示的实验拓扑中，要求使用 Linux 构建安全、可靠的防火墙。具体要求如下：

(1) 只允许在防火墙主机上进行操作管理。除管理员外，禁止任何人访问防火墙。

(2) 内网 Web 服务器要求通过地址映射发布出去，只允许外网用户访问 Web 服务器的 80 号端口，而且需通过有效的 DNS 注册。

(3) 内网用户必须通过防火墙才能访问内网 Web 服务器，不允许直接访问该服务器。

(4) 内网 FTP 服务器只对内网用户提供服务，且只允许内网用户访问 FTP 服务器的 21 号和 20 号端口，不允许外网用户访问该服务器。

(5) 内网用户要求通过透明代理上网(不需要在客户机浏览器上做任何设置就可以上网)。

(6) 内网用户所有的 IP 地址必须通过 NAT 转换之后才能够访问外网。

【实验过程】

1) 实验准备

(1) 实验需要安装 Apache 和 vsftpd：

```
sudo apt-get install vsftpd
sudo apt-get install Apache2
```

软件安装后，根据实际情况决定是否对配置文件 apache2.conf、vsftpd.conf 和 allowed_users.conf 进行修改。FTP 服务器还要新建用户、设置密码。

(2) 验证能否正常访问 Web 服务器和 FTP 服务器，如不能访问，需要予以解决。

(3) 策略分析。实验重点在防火墙策略的考量上。通常先将防火墙的策略设置为最严格监控，然后根据需要逐步放宽管理。

2) Linux 下 iptables 的具体设置

(1) 清空 filter 表和 nat 表中的配置策略。

清空所选的系统表 filter 中的默认链：

```
iptables -F
```

清空所选的系统表 nat 中的默认链：

```
iptables -F -t nat
```

删除表中的自定义规则链：

```
iptables -X -t
```

指定链的所有计数器清零：

```
iptables -Z -t
```

(2) 放行 filter 表的默认 OUTPUT 链，阻止 FORWARD 链和 INPUT 链。全部放行 nat 表中的 PREROUTING 链、POSTROUTING 链以及 OUTPUT 链。设置完成之后，目前的状态是只允许数据从内网出去，不允许外网任何数据进来。

设置默认策略规则：

```
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -P INTPUT DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P OUTPUT ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
```

测试：当前外网用户是否可以通过 80 号端口访问防火墙外网接口的 IP 地址？

(3) 将内网 Web 服务器的 IP 地址映射到防火墙外网接口的 IP 地址上，命令如下：

```
iptables -t nat -A PREROUTING -p tcp -d 202.101.200.200 --dport 80 -j DNAT --to-destination 192.168.10.20
```

设置完成之后，外网用户就可以通过 80 号端口访问防火墙外网接口的 IP 地址。并能够将目的 IP 地址转换为内网 Web 服务器的 IP 地址，但是还不能够访问内网 Web 服务器。

测试：IP 地址转换情况如何？能否访问内网 Web 服务器？

(4) 放行转发访问内网 Web 服务器的数据包，命令如下：

```
iptables -A FORWARD -p tcp -d 192.168.10.20 --dport 80 -j ACCEPT
```

设置完成之后，外网用户除可以通过 80 号端口访问防火墙外网接口的 IP 地址，还能够将外网 IP 地址映射为内网 Web 服务器的 IP 地址，进而访问内网 Web 服务器。

测试：贴出外网用户访问内网 Web 服务器截图。内网用户能否访问内网 Web 服务器？

(5) 将内网 Web 服务器的 IP 地址映射到防火墙内网接口的 IP 地址上，命令如下：

```
iptables -t nat -A POSTROUTING -p tcp -d 192.168.10.20 --dport 80 -j SNAT --to-source 192.168.10.10
```

设置完成之后，内网用户就可以通过访问防火墙内网接口的 IP 地址访问内部 Web 服务器。之所以不直接让内网用户通过内网 IP 地址直接访问，是为了增强内网 Web 服务器的安全性。

测试：贴出内网用户访问内网 Web 服务器截图。内网用户能否访问 Internet？