第 5 章　网络攻击与防范技术

从信息安全技术体系的角度来讲，网络攻击和评测的理论与实践是对信息系统安全性的考验。兵法说，知己知彼，百战不殆。只有对网络攻击技术和方法有深入、详细的了解，才能对系统提供更有效的保护。

5.1　网络攻击概述和分类

简单地说，“攻击”是指一切针对计算机的非授权行为。攻击的全过程应该是由攻击者发起的，攻击者应用一定的攻击方法和攻击策略，利用一些攻击技术或工具，对目标系统进行非法访问，达到一定的攻击效果，并实现攻击者的预定攻击目标。因此，凡是试图绕过系统的安全策略，或者对系统进行渗透，以获取信息、修改信息甚至破坏目标网络或系统功能为目的的行为都可以称为攻击。

5.1.1　网络安全漏洞

从技术上说，网络容易受到攻击的原因主要是网络软件不完善和网络协议本身存在安全漏洞。例如，使用最多、最著名的 TCP/IP 就存在大量的安全漏洞。这是因为 TCP/IP 在设计时，设计人员只考虑到如何实现粗犷的信息通信，而忽略了会有人破坏信息通信的安全性问题。下面举例说明 TCP/IP 的几个安全漏洞。

(1) 由于 TCP/IP 数据流采用的是明文传输，因此电子信息很容易被在线窃听、篡改和伪造。特别是在使用 FTP 和 Telnet 命令时，如果用户的账号、口令是明文传输的，攻击者就可以使用 Sniffer、WireShark 等软件截取用户的账号和口令。

(2) 由于 TCP/IP 是用 IP 作为网络节点的唯一标识，但是节点的 IP 地址却是不固定的，而且是一个公共数据，因此攻击者可以直接通过修改节点的 IP 地址来冒充某个可信节点的 IP 地址进行攻击，实现源地址欺骗或 IP 欺骗。所以，IP 地址不能作为一种可信的认证方法。

(3) TCP/IP 只能根据 IP 地址进行鉴别，而不能对节点上的用户进行有效的身份认证，因此服务器无法鉴别登录用户身份的有效性。目前主要依靠服务器软件平台提供的用户控制机制，如用户名、口令等进行身份认证。

TCP/IP 的安全漏洞还有很多，感兴趣的读者可以查阅有关网络安全方面的书籍，这里不赘述。

除了 TCP/IP 漏洞以外，软件系统本身的漏洞也是给网络攻击有机可乘的另外一个重要因素。

从操作系统的发展历史可以看到，早期的 Windows 3.1 操作系统大概有 300 万行代码，发展到后来的 Windows 95 约有 1500 万行代码，Windows 98 约有 1800 万行代码，Windows XP 约有 3500 万行代码，Windows 2000 约有 4000 万行代码，Windows Vista 系统

约有5000万行代码,发展到现在的Windows 7经过优化和精简大概有4000万行代码。可想而知,如此庞大规模的代码量,再加上人们的认知能力和实践能力的有限性,出现很多漏洞是一个大概率事件。图5.1所示为从国家漏洞库(CNNVD)统计得到的数据,可以看出,随着信息技术的发展,2006年以前漏洞的数量总体呈现一个快速上升的趋势,而在2006年以后则呈现小幅下降。在这些漏洞中,基础型漏洞(如系统内核漏洞)数量下降速度较快,但应用型漏洞数量却急剧增加,特别是Web漏洞数量增长极为明显。这一方面说明开发者的安全意识和防范技术都日渐提高以后,部分漏洞得到了适当的避免,但另一方面也说明有可能受到利益驱使,部分漏洞信息在地下传播,导致公开漏洞信息减少。从图5.1中可以看出,漏洞是无法避免的,安全的风险随时存在。

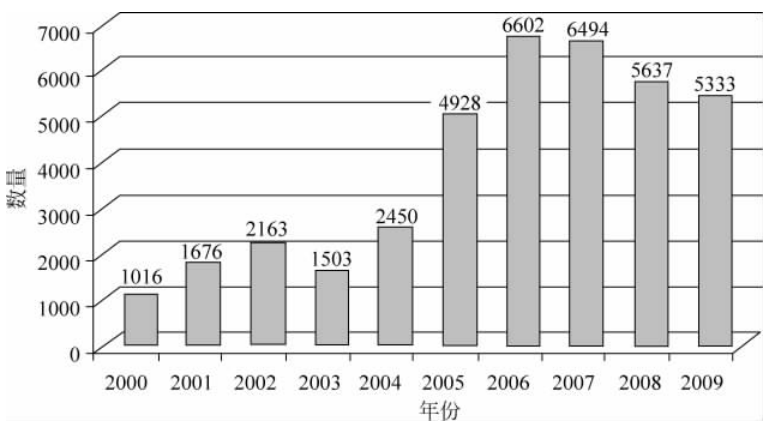

图5.1 近十多年安全漏洞发布趋势

5.1.2 网络攻击的基本概念

在介绍网络攻击概念之前,首先要清楚为什么会存在网络攻击,网络攻击的理由和目标又是什么。

其实,大多数网络攻击的理由都很简单,大体可以分为以下几个原因。

(1) 想要在别人面前炫耀自己的技术。例如,进入别人的计算机去修改一个文件或目录名。

(2) 恶作剧、练功。这是许多人进行入侵或破坏的主要原因,除了练习的效果外,还可得到网络探险的感觉。

(3) 窃取数据。可能是偷盗硬盘中的文件或各种账户和密码,然后从事某种商业应用。

(4) 报复心理。例如,对老板或公司制度不满,事先把报复程序或病毒程序写入所编的程序,并设定在将来某个时刻或某种条件下激活并发作,摧毁原公司的网络系统。

(5) 抗议或宣示。例如,2001年5月1日中美黑客大战,中美两国的黑客相互攻击对方网站,双方均有数以千计的网站遭到攻击,轻者被篡改主页面,严重的则整个系统遭受毁灭性打击。

总体来说,网络攻击可以从攻击的位置、攻击的层次进行分类。通常,攻击的位置有两种,即远程攻击和本地攻击。远程攻击是指外部攻击者通过各种手段,从该子网以外的地方向该子网或子网内的系统发动攻击,攻击发起者通常不会用自己的机器直接发动攻击,而是通过跳板的方式对目标进行迂回攻击,以迷惑系统管理员,避免暴露自己的真实身份。本地

攻击是指本单位的内部人员通过所在的局域网向本单位的其他系统发动攻击。

目前常见的网络攻击方法，大致可以分为以下几类。

(1) 窃听。攻击者通过非法手段对系统活动进行监视，从而获得一些安全关键信息。目前属于窃听技术的常用攻击方法有以下几种。

① 键击记录：进入操作系统内核的隐蔽软件通常为一个键盘设备驱动程序，能够把每次键击都记录下来，并存放到攻击者指定的本地隐藏文件中，如 Windows 32 平台下使用的 IKS 等。

② 网络监听：攻击者一旦在目标网络上获得一个立足点之后，刺探网络情报的最有效方法就是网络窃听。其通过设置网卡的混杂模式获得网络上所有的数据包，并从中抽取关键信息，如明文方式传输的口令等。网络监听工具有 Windows 平台下的 Sniffer 和 UNIX 平台下的 Libpcap 等。

③ 非法访问数据：攻击者或内部人员违反安全策略对其访问权限之外的数据进行非法访问。

④ 获取密码：进行口令破解，获取特权用户或其他用户的口令。

(2) 欺骗。攻击者冒充正常用户以获取对攻击目标的访问权或获取关键信息。属于此类的攻击方法有以下几种。

① 获取口令：通过默认口令、口令猜测和口令破解 3 种途径。针对一些弱口令进行猜测，也可以使用专门的口令猜测工具进行破解，如遍历字典或高频密码列表，从而找到正确的口令。

② 恶意代码：包括特洛伊木马应用程序、邮件病毒、网页病毒等，通常冒充成有用的软件工具，诱导用户下载运行，或者利用邮件客户机和浏览器的自动运行机制，在启动后悄悄安装恶意程序，通常为攻击者给出能够完全控制该主机的远程连接。

③ 网络欺骗：攻击者通过向攻击目标发送冒充其信任主机的网络数据包，达到获取访问权限或执行命令的目的，具体有 IP 欺骗、会话劫持、ARP 重定向和 RIP 路由欺骗等。

(3) 拒绝服务。拒绝服务是指造成终端完全拒绝对合法用户、网络、系统和其他资源的服务的攻击方法，其意图就是彻底破坏系统。这也是比较容易实现的攻击方法。特别是分布式拒绝服务攻击对目前的 Internet 构成了严重威胁。

(4) 数据驱动攻击。通过向某个程序发送数据，以产生非预期结果的攻击，通常为攻击者给出访问目标系统的权限。大致可分为以下几种。

① 缓冲区溢出：通过向程序的缓冲区中写入超出其边界的内容造成缓冲区的溢出，使得程序转而执行攻击者指定的代码，通常是为攻击者打开远程连接的 ShellCode，以达到攻击的目的。

② 格式化字符串攻击：主要利用由于格式化输出函数的微妙程序设计错误造成的安全漏洞，通过传递精心编制的含有格式化指令的文本字符串，以使目标程序执行任意命令。

③ 信任漏洞攻击：利用程序滥设的信任关系获取访问权限的一种方法。

5.1.3　网络攻击的步骤概览

如图 5.2 所示，网络攻击的一般流程大致如下。

(1) 目标探测。攻击者在攻击之前的首要任务，就是明确攻击目标是单个主机还是整

个网段,并了解目标的具体网络信息等。

(2) 端口扫描。通过端口扫描可以搜集到目标主机的各种有用信息,包括端口是否开放,能否匿名登录,等等。

(3) 网络监听。黑客可以借助网络监听技术对其他用户进行攻击,同时也可以截获用户名、口令等有用信息。

(4) 实施攻击。采用有效的方式对目标主机进行攻击,如缓冲区溢出、DoS等。

(5) 撤退。留下后门,消除攻击的痕迹。

图5.2 网络攻击的一般流程

5.2 目标探测

攻击者在攻击以前的首要任务,就是要明确攻击对象是单个主机还是整个网段。目标探测是通过自动或人工查询的方法获得与目标网络相关的物理和逻辑参数。目标探测是黑客攻击的第一步。

5.2.1 目标探测的内容

目标探测所包含的内容基本上有以下两类。

(1) 外网信息。外网信息包括域名、管理员信息、域名注册机构、DNS主机、网络地址范围、网络位置、网络地址分配机构信息、系统提供的各种服务和网络安全配置等。

(2) 内网信息。内网信息包括内部网络协议、拓扑结构、系统体系结构和安全配置等。

一次攻击的成功与前期的目标探测关系很大。通常,目标探测方法可以分为以下三类。

(1) 使用各种扫描工具对攻击目标进行大规模扫描,得到系统信息和运行时的服务信息。这涉及一些扫描工具的使用,将在后面的章节中介绍。

(2) 利用第三方资源(如常用的搜索引擎谷歌、百度等)对目标进行信息收集。其实,Google Hacking在国外已经流行很久了。攻击者利用谷歌强大的搜索功能来搜索某些关键词,找到有系统漏洞和Web漏洞的服务器,并将其打造成自己的“肉鸡”。

(3) 利用各种查询手段得到与被攻击者相关的一些信息。通过这种方式得到的信息会被社会工程学这种入侵手法用到。社会工程学(Social Engineering)通常是利用大众疏于防范的心理,让受害者掉入陷阱。该技术通常采用交谈、欺骗、假冒或口语用字等方式,从合法用户中套取敏感的信息,如用户名单、用户密码及网络结构等,即使很小心的人,也有可能被高明的社会工程学手段侵害。网络安全是一个整体,对某个目标在久攻不下的情况下,黑客会把矛头指向目标的系统管理员,因为人在这个整体中往往是最不安全的因素。黑客通过搜索引擎对系统管理员的一些个人信息进行搜索,如电子邮件地址、MSN、QQ等关键词,分析出这些系统管理员的个人爱好,常去的网站、论坛等,然后利用掌握的信息与系统管理员拉关系套近乎,骗取对方的信任,使其一步步落入黑客设计好的圈套,最终系统被入侵。这也就是常说的“没有绝对的安全,只有相对的安全;只有时刻保持警惕,才能换来网络的安宁”。

5.2.2　目标探测的方法

目标探测的方法和手段多种多样，除了必要的技术之外，还要有丰富的经验和相应的技巧。

1. 确定目标范围

入侵一个目标，首先要确定该目标的网络地址分布和网络分布范围及位置，通过开放的资源进行搜索是获得该信息最有效的方法，因为在因特网上的一些规模巨大的数据库可以方便、自由和实时地提供目标网络的信息。

例如，目标网络中有一个域名 www. sina. com. cn，通过该域名可以查看提供该 Web 服务的一台服务器的地址（一个大型网站通常有很多台服务器提供同一个网站的服务）。通过 Ping 命令就可以获取其中一台服务器的 IP 地址。

```
C:\> ping www.sina.com.cn
    Pinging newstietong.sina.com.cn [211.98.132.93] with 32 bytes of data:
Reply from 211.98.132.93: bytes = 32 time = 53ms TTL = 55
```

屏幕上所显示的 211. 98. 132. 93 就是提供 www. sina. com. cn 服务的一台服务器地址。但这种方法可以被防火墙所屏蔽。

另外，还可以利用 Whois 查询得到目标主机的 IP 地址分配、机构地址位置和接入服务商等重要信息。

Whois 查询就是查询域名和 IP 地址的注册信息。国际域名由设在美国的 Internet 信息管理中心（InterNIC）和它设在世界各地的认证注册商管理，国内域名由中国互联网络信息中心（CNNIC）管理。通过 http：//www. allwhois. com 就可以查询到目标主机的相关信息。

随着 Internet 的迅猛发展，各种信息呈现爆炸式的增长，用户要在信息海洋里查找信息就像大海捞针一样。每个上网用户都面临着信息过载的问题，无法准确找到所需要的信息。搜索引擎正是为了解决这个问题而出现的技术。现在通过谷歌、百度等搜索引擎可以获得大多数需要的信息。也就是说，通过搜索引擎，同样可以获得大多数目标主机的相关信息。

当然，借助于一些软件工具，也可以获得目标网络的相关信息，如 Netscan、VisualRoute 和 Traceroute 等。这些软件的主要功能是快速分析和辨别 Internet 连接的来源，标示某个 IP 地址的地理位置，目标网络的 Whois 查询，提供可视化的显示。图 5. 3 所示为 Visual Route 的主界面。

2. 分析目标网络路由

虽然每次数据包从某个出发点到达同一目的地所走的路径可能不一样，但大部分时间是相同的，因此了解信息从一台计算机到达另一台计算机的传播路径是非常重要的。如果某段网络不通或网速很慢，可以利用路由跟踪找出故障点，方便维护人员的维护工作。对于攻击者来说，这是个很有用的功能，它可以大概分析出目标所在网络的状况。

要检测数据包的传播路径有很多种工具，目前最常用的检测工具是 Traceroute。该工具在 UNIX 系统环境中的命令为 Traceroute，在 Windows 中的命令为 Tracert。在 Windows 中有最新的 3d Traceroute，如图 5. 4 所示，可以通过图形界面的形式给出跟踪的结果。通过

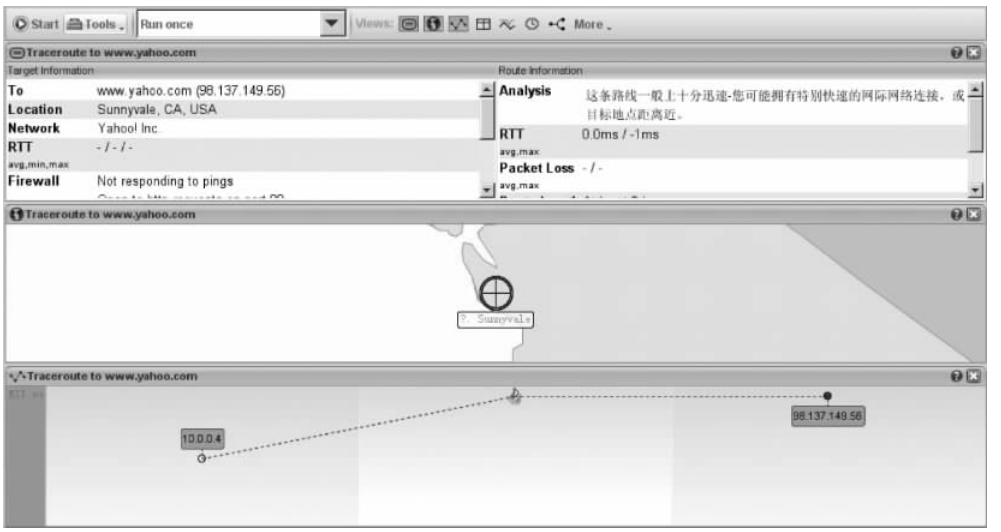

图 5.3 Visual Route 的主界面

Traceroute 可以知道信息从本地计算机到 Internet 另一端的主机走的是什么路径，通过发送小的数据包到目标设备再返回来测量其需要多长时间。一条路径上的每个设备要测试 3 次，输出结果中包括每次测试的时间和设备的名称及其 IP 地址。在这里通过 Windows 中的 Tracert 来介绍路由跟踪技术。

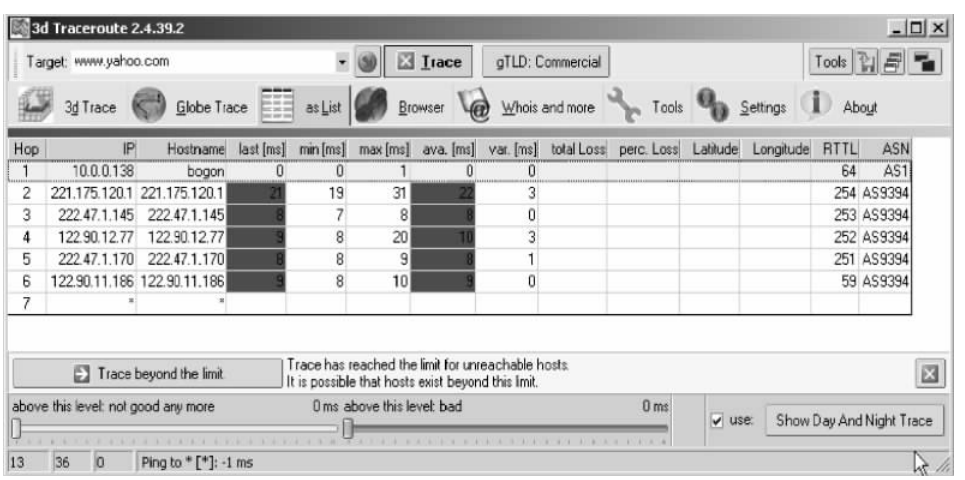

图 5.4 Traceroute 的主界面

1) Traceroute 工作原理

Traceroute 程序的设计是利用 ICMP 及 IP header 的 TTL 字段。首先，Traceroute 送出一个 TTL 是 1 的 IP 报文，当路径上的第一个路由器收到这个数据包时，它将 TTL 减 1。此时，TTL 变为了 0，所以该路由器会将此数据包丢掉，并送回一个“ICMP time exceeded”消息(包括发 IP 包的源地址、IP 包的所有内容及路由器的 IP 地址)，Traceroute 收到这个消息后，便知道这个路由器存在于这个路径上，接着 Traceroute 再送出另一个 TTL 是 2 的

数据包，发现第二个路由器……Traceroute 通过每次将送出的数据包的 TTL 加 1 来发现另一个路由器，这个重复的动作一直持续到某个数据包抵达目的地。当数据包到达目的地后，该主机并不会送回 ICMP time exceeded 消息，因为它已是目的地了。那么，Traceroute 是如何得知 UDP 数据包已到达目的地了呢？

Traceroute 在送出 UDP 数据包到目的地时，它所选择送达的端口号是一般应用程序都不会用的一个号码（30 000 以上），所以当此 UDP 数据包到达目的地后，该主机会送回一个“ICMP port unreachable”的消息，而当 Traceroute 收到这个消息时，便知道目的地已经到达了。所以，Traceroute 在 Server 端也是没有所谓的 Daemon 程序。

Traceroute 提取发 ICMP TTL 到期消息设备的 IP 地址并做域名解析。每次，Traceroute 都打印出一系列数据，包括所经过的路由设备的域名及 IP 地址，3 个包每次来回所需要的时间。

Traceroute 有一个固定的时间等待响应（ICMP TTL 到期消息）。如果这个时间过了，它将打印出一系列的 * 号，以表明在这个路径上该设备不能在给定的时间内发出 ICMP TTL 到期消息的响应。然后，Traceroute 给 TTL 计数器加 1，继续进行。

在大多数情况下，作为网络工程技术人员或系统管理员会在 UNIX 主机系统下直接执行命令行：

```
Traceroute hostname
```

而在 Windows 系统下是执行 Tracert 命令：

```
Tracert hostname
```

如果要使用 Windows NT 系统中的 Tracert 命令，用户可通过选择“开始”→“运行”命令，在打开的对话框中输入 cmd 调出命令窗口，然后使用此命令。

```
C:\> tracert www.yahoo.com
Tracing route to www.yahoo.com [204.71.200.75] over a maximum of 30 hops:
  1 161 ms 150 ms 160 ms 202.99.38.67
  2 151 ms 160 ms 160 ms 202.99.38.65
  3 151 ms 160 ms 150 ms 202.97.16.170
  4 151 ms 150 ms 150 ms 202.97.17.90
  5 151 ms 150 ms 150 ms 202.97.10.5
  6 151 ms 150 ms 150 ms 202.97.9.9
  7 761 ms 761 ms 752 ms border7 - serial3 - 0 - 0.Sacramento.cw.net [204.70.122.69]
  8 751 ms 751 ms * core2 - fddi - 0.Sacramento.cw.net [204.70.164.49]
  9 762 ms 771 ms 751 ms border8 - fddi - 0.Sacramento.cw.net [204.70.164.67]
  10 721 ms * 741 ms globalcenter.Sacramento.cw.net [204.70.123.6]
  11 * 761 ms 751 ms pos4 - 2 - 155M.cr2.SNV.globalcenter.net [206.132.150.237]
  12 771 ms * 771 ms pos1 - 0 - 2488M.hr8.SNV.globalcenter.net [206.132.254.41]
  13 731 ms 741 ms 751 ms bas1r - ge3 - 0 - hr8.snv.yahoo.com [208.178.103.62]
  14 781 ms 771 ms 781 ms www10.yahoo.com [204.71.200.75]
Trace complete.
```

2）用 Traceroute 解决问题

Traceroute 最早是由 Van Jacobson 在 1988 年编写的小程序。当时主要是为解决他自己碰到的一些网络问题。Traceroute 是一个正确理解 IP 网络并了解路由原理的重要工具。它对负责网络工程技术与系统管理的 Webmaster 来说是一个十分方便的程序。

可以使用 Traceroute 确定数据包在网络上的停止位置。下例中,默认网关确定 192.168.10.99 主机没有有效路径,这可能是路由器配置的问题,或者是 192.168.10.0 网络不存在(错误的 IP 地址)。

```
C:> Tracert 192.168.10.99
Tracing route to 192.168.10.99 over a maximum of 30 hops
1 10.0.0.1 reports: Destination net unreachable.
Trace complete.
```

Tracert 实用程序对于解决大网络问题非常有用,可以采取几条路径到达同一个点。

5.3 扫描的概念和原理

扫描就是对计算机系统或其他网络设备进行安全相关的检测,以找出安全隐患和可能被黑客利用的漏洞。例如,可以通过扫描发现远程服务器各种 TCP 端口的分配情况、提供的服务和它们的软件版本,从而间接或直观地了解远程主机所存在的安全问题。通过扫描,能对扫描对象的脆弱性和漏洞进行深入了解,从而为扫描时发现的问题提供一个良好的解决方案。对于黑客来说,扫描是信息获取的重要步骤,通过网络扫描可以进一步定位目标或区域目标系统相关的信息,同时为下一步的攻击提供充分的资料,从而大大提高攻击的成功率。

扫描技术可以分为三类:主机扫描、端口扫描和漏洞扫描。其中,主机扫描能够发现系统的存活情况,确定在目标网络上的主机是否可达,同时尽可能多地映射目标网络的拓扑结构,其主要利用 ICMP 数据包来实现;端口扫描用于发现远程主机开放的端口,也就是发现哪些服务在运行;漏洞扫描能够发现和了解网络上潜在的脆弱性,以便采取措施,避免遭受不必要的攻击。

5.3.1 主机扫描

主机扫描分为简单主机扫描和复杂主机扫描。传统的主机扫描是利用 ICMP 的请求/应答报文,主要有以下 3 种。

(1) 通过发送一个 ICMP Echo Request 数据包到目标主机,如果接收到 ICMP Echo Reply 数据包,说明主机是存活状态;如果没有收到,就可以初步判断主机没有在线或使用了某些过滤设备过滤了该消息。

(2) 使用 ICMP Echo Request 轮询多个主机称为 Ping 扫描。对于中型网络,使用这种方法来探测主机是一种比较好的方式,但对大型网络,这种方法会比较慢,因为 Ping 在处理下一个命令之前会等待正在探测主机的回应。

(3) 广播 ICMP 扫描,即通过发送 ICMP Echo Request 到广播地址或目标网络地址可以简单地反映目标网络中活动的主机,这样的请求会广播到目标网络中的所有主机,所有活动的主机都会发送 ICMP Echo Reply 到攻击者的 IP 地址。

这 3 种方法的缺点是会在目标主机的 DNS 服务器中留下攻击者的日志记录。

利用被探测主机产生的 ICMP 错误报文可以进行复杂的主机扫描,主要有以下几种方式。

(1) 异常的 IP 包头。向目标主机发送包头错误的 IP 包,目标主机或过滤设备会反馈 ICMP Parameter Problem Error 信息。常见的伪造错误字段为 Header Length 和 IP Options。

(2) IP 头中设置无效的字段值。向目标主机发送的 IP 包中填充错误的字段值,如协议项填一个没有使用的超大值,目标主机或过滤设备会反馈 ICMP Destination Unreachable 信息。

(3) 错误的数据分片。当目标主机接收到错误的数据分片,并且在规定的时间间隔内得不到更正时,将丢弃这些错误数据包,并向发送主机反馈 ICMP Fragment Reassembly Time Exceeded 错误报文。

(4) 反向映射探测。用于探测被过滤设备或防火墙保护的网络和主机。构造可能的内部 IP 地址列表,并向这些地址发送数据包。当对方路由器接收到这些数据包时,会进行 IP 识别并路由,对不在其服务范围的 IP 包发送 ICMP Host Unreachable 或 ICMP Time Exceed 错误报文,没有接收到相应错误报文的 IP 地址被认为在该网络中。

对主机扫描的工具非常多,如著名的 Nmap、Netcat 和 Superscan 等。

主机扫描大多使用 ICMP 数据包,因此使用可以检测并记录 ICMP 扫描的工具,使用入侵检测系统,在防火墙或路由器中设置允许进出自己网络的 ICMP 分组类型等方法都可以有效地防止主机扫描的发生。

5.3.2　端口扫描

端口扫描的直接结果就是可以得到目标主机开放和关闭的端口列表。这些开放的端口往往与某些服务相对应。通过这些开放的端口,黑客就能了解主机运行的服务类型,从而进一步整理和分析这些服务可能存在的漏洞,为后续的攻击提供依据。端口扫描是建立在 TCP/IP 基础之上的。在 TCP/IP 的实现中,一般遵循以下原则。

(1) 当一个 SYN 或 FIN 数据包到达一个关闭的端口时,TCP 丢弃数据包,同时发送一个 RST 数据包。

(2) 当一个 SYN 数据包到达一个监听端口时,正常的三阶段握手继续,回答一个 SYN|ACK 数据包。

(3) 当一个 SYN|ACK 或 FIN 数据包到达一个监听端口时,数据包被丢弃。

(4) 当一个 SYN|ACK 或 FIN 数据包到达一个关闭端口时,数据包被丢弃,并返回一个 RST 数据包。

(5) 当一个包含 ACK 的数据包到达一个监听或关闭的端口时,数据包被丢弃,同时发送一个 RST 数据包。

(6) 当一个 SYN 位关闭的数据包到达一个监听端口时,数据包被丢弃。

基于上述的 TCP/IP,常用的端口扫描方法主要有 TCP Connect 扫描、TCP SYN 扫描、TCP FIN 扫描和 TCP NULL 扫描等。

1. 常用的端口扫描技术

1) TCP Connect 扫描

TCP Connect 扫描是最为简单的端口扫描方式,本地主机通过调用 Connect 函数连接目标主机的特定端口,如果成功建立连接,则说明这个端口是打开的;否则,说明该端口是关闭的。因为该扫描需要建立一个完整的端口连接,所以该扫描也称为全连接扫描。

该方法最大的优点是不需要任何权限,系统中的任何用户都可以使用这个调用。该方法的另一个优点是速度比较快。该方法最大的缺点是容易被察觉,因为它会在目标计算机的日志文件中留下一串连接的消息。

2) TCP SYN 扫描

扫描器向目标主机的选择端口发送 SYN 置 1 的数据包,如果应答是 RST 置 1 的数据包,则说明端口是关闭的,如图 5.5 所示。如果应答是 SYN 和 ACK 置 1 的数据包,则说明目标处于监听状态,再传送一个 RST 包给目标机,停止建立连接,如图 5.6 所示。由于在 TCP SYN 扫描时全连接尚未建立,因此这种技术通常称为半打开扫描。

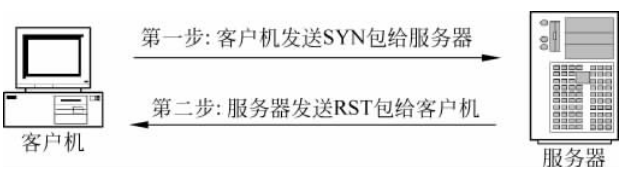

图 5.5 目标端口关闭时 TCP SYN 扫描的步骤

图 5.6 目标端口打开时 TCP SYN 扫描的步骤

TCP SYN 扫描的优点是隐蔽性比全连接扫描好,因为很少有系统会记录这样的行为。另外,它的扫描结果也是相当准确的,并能达到很快的速度。该方法的缺点是通常构造 SYN 数据包需要超级用户或授权用户访问专门的系统调用。SYN 洪泛是一种常见的拒绝服务攻击方法,许多防火墙和入侵检测系统对 SYN 包都建立了报警和过滤机制,因此 SYN 扫描的隐蔽性逐渐下降。

3) TCP FIN 扫描

TCP FIN 扫描是利用操作系统协议栈实现上的不同来达到扫描的目的。客户机向目标端口发送一个带 FIN 标志的数据包,如果目标端口是开放的,它就会忽略这个数据包;如果目标端口关闭了,目标主机会向本地主机回应一个 RST 数据包,如图 5.7 所示。利用这点差异就可以判断目标主机是否开放了某个端口。FIN 扫描只对 UNIX/Linux 系统有效。FIN 扫描的优点是比 TCP SYN 扫描更为隐蔽,能够通过只检测 SYN 包的防火墙或入侵检测系统。它的缺点是因其是反向确定结果,如果网络的传输收不到返回包就会导致错误判

图 5.7 目标端口关闭时 TCP FIN 扫描的步骤

断，扫描结果不是很可靠。

4）TCP Xmas 扫描

根据 RFC793 规定，当主机收到一个带 FIN、URG 和 PSH 标志的 TCP 数据包时，如果其对应的端口开放，则会忽略这个数据包；如果其对应的端口关闭，主机会返回一个 RST 包作为响应。利用这种差异就可以判断目标端口是否开放，如图 5.8 所示。

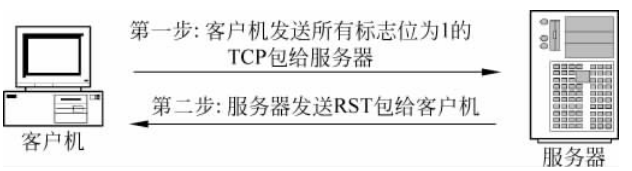

图 5.8　目标端口关闭时 TCP Xmas 扫描的步骤

这种扫描技术的优点是扫描活动比较隐蔽；不足之处是效率不高，需要等待超时，而且这里涉及数据包的构造与发送，所以需要管理员权限才能操作。

5）TCP NULL 扫描

与 TCP Xmas 扫描相反，TCP NULL 扫描将 TCP 包中的所有标志位都置 0。当这个数据包被发送到主机时，如果目标端口是开放的，则不会返回任何数据包；如果目标端口是关闭的，被扫描主机将发回一个 RST 包，如图 5.9 所示。不同的操作系统会有不同的响应方式。

图 5.9　目标端口关闭时 TCP NULL 扫描的步骤

这种扫描技术的优点也是比较隐蔽；不足之处与前一种扫描技术一样，需要等待超时，所以效率不高。此外，不同操作系统的扫描有差别，不能适用于所有的操作系统，而且仍然需要管理员权限才能操作。

6）UDP 扫描

UDP 扫描利用 UDP 向目标端口发送一个 UDP 包，开放的 UDP 端口并不需要送回 ACK 包，而关闭的端口会送回一个 ICMP_PROT_UNREACH 的包，用于说明端口关闭。

UDP 扫描并不可靠，主要原因有以下几点。

① 目标主机可以禁止任何 UDP 包通过。

② UDP 本身不是可靠的传输协议，数据传输的完整性不能得到保证。

③ 系统在协议栈的实现上有差异，对一个关闭的 UDP 端口，可能不会返回任何信息，而只是简单地丢弃。

7）FTP 返回扫描

FTP 返回扫描是利用 FTP 支持代理 FTP 连接这个特点来实现的。本地主机首先与 FTP 服务器建立连接，然后通过 PORT 命令向 FTP 服务器传输目标主机的地址和端口，最

后发送 LIST 命令。如果目标主机相应的端口已打开,就会返回成功的消息;如果目标端口关闭,则返回连接失败的消息。FTP 返回扫描示意图如图 5.10 所示。

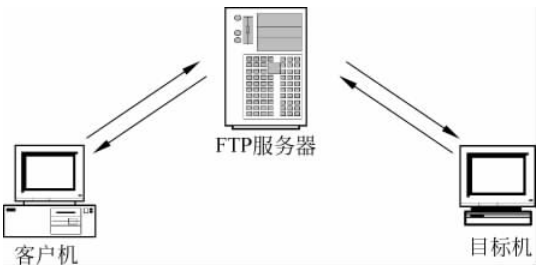

图 5.10 FTP 返回扫描示意图

这种扫描的优点很明显,很难跟踪,而且能有效穿透防火墙。缺点是速度较慢,而且需要一台 FTP 服务器做代理,现在提供这种功能的服务器较少。

2. 防止端口扫描

对抗端口扫描的对策主要有以下两种方法。

1) 关闭闲置和有潜在危险的端口

除正常使用的计算机端口外(如 HTTP 的 80 端口,FTP 的 21 端口,QQ 的 4000 端口等),将所有其他端口都关闭。因为对黑客来说,所有端口都可能成为攻击的目标。

在 Windows NT 为核心的操作系统中,要关闭闲置端口还是比较方便的,可以采用"定向关闭指定服务的端口"和"只开放允许端口的方式"。计算机的一些网络服务会由系统分配默认的端口,将一些限制的服务关闭掉,其对应的端口也关闭了。打开"控制面板"→"管理工具"→"服务"选项,关闭一些没有使用的服务,它们对应的端口也就关闭了。至于"只开放允许端口的方式",可以利用系统的 TCP/IP 筛选功能实现,设置的时候只允许系统中一些基本网络通信需要的端口即可。在 UNIX/Linux 系统中,在/etc/inetd.conf 中注释掉不必要的服务,并在系统启动脚本中禁止其他不必要的服务。

2) 利用网络防火墙软件

对抗端口扫描最好的方法是使用防火墙软件,当攻击者进行端口扫描时,攻击者会不断与目标计算机尝试建立连接,可以通过防火墙自带的拦截规则进行判断,当发现有端口扫描症状时,通过防火墙可以立即屏蔽该端口,即通过设置防火墙的过滤规则,可以有效阻止对端口的扫描。例如,可以设置检测 SYN 扫描而忽略 FIN 扫描。另外,借助入侵检测系统,禁止所有不必要的服务,把自己的机器暴露程度降到最低也是一种很好的方法。

5.3.3 漏洞扫描

漏洞扫描是对目标网络或目标主机进行安全漏洞检测与分析,发现可能被攻击者利用的漏洞。当前的漏洞扫描技术主要是基于特征匹配原理,漏洞扫描器通过检测目标主机不同端口开放的服务,记录其应答,然后与漏洞库进行比较,如果满足匹配条件,则认为存在安全漏洞。漏洞扫描技术中,漏洞库的定义精确与否直接影响最后的扫描结果。

目前,漏洞扫描器主要分为两类,即通用漏洞扫描器和专用漏洞扫描器。它们各自的侧重点不同。通用漏洞扫描器侧重扫描主机的整体安全,适用于攻击及本机防护;专用漏洞

扫描器侧重主机的某一特定漏洞，主要用于漏洞攻击。

通用漏洞扫描器一般由控制台模块、扫描活动处理模块、扫描引擎模块、结果处理模块和漏洞库组成。而专用漏洞扫描器相对于通用漏洞扫描器来说要简单一些，可以说是一种简化了的通用漏洞扫描器。专用漏洞扫描器不用考虑多个漏洞，只需检测某个特定的漏洞，并发线程减少，检测效率提高了很多。

目前常用的漏洞扫描工具主要有 Nmap、X-Scan、SuperScan、Shadow Security Scanner 和 MS06040Scanner 等。感兴趣的读者可以通过网络信息进一步了解漏洞扫描工具。

5.4　网 络 监 听

网络监听技术是提供给网络安全管理人员进行网络管理的工具，可以用来监视网络状态、数据流动情况及网络上传输的信息等。当信息以明文的形式在网络上传输时，只要将网卡设置成混杂模式，便可以源源不断地截获网络上传输的信息。然而，黑客也会利用网络监听技术对其他用户进行攻击，黑客可以利用网络监听截取口令，当黑客控制一台主机后，如果想通过这台主机控制其所在的整个局域网，网络监听往往是他们的最佳选择。

5.4.1　网络监听原理

在因特网上有很多使用以太网(Ethernet)协议的局域网，许多主机通过电缆、集线器连在一起。在协议的高层或从用户的角度来看，当同一网络中的两台主机通信时，源主机将写有目标主机地址的数据包发向目标主机，或者当网络中的一台主机同外部的主机通信时，源主机将写有目标主机 IP 地址的数据包发向网关。

但这种数据包并不能在协议栈的高层直接发出去，要发送的数据必须从 TCP/IP 的 IP 层交给网络接口，而网络接口是不会识别 IP 地址的，因此网络接口中由 IP 层传来的带有 IP 地址的数据包又增加了一部分以太帧帧头的信息。在帧头中，有两个域分别为只有网络接口才能识别的源主机和目标主机的物理地址。这是一个与 IP 地址对应的 48 位地址。

下面用一个常见的 UNIX 系统命令 ifconfig 来看一看作者本人的一台正常工作的计算机的网卡：

```
[yiming@server/root]# ifconfig -a
    hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.35 netmask ffffffe0
        ether 8:0:20:c8:fe:15
```

从这个命令的输出中可以看到上面讲到的这些概念，如第二行的 192.168.1.35 是 IP 地址，第三行的 8：0：20：c8：fe：15 是 MAC 地址。请注意第一行的 BROADCAST 和 MULTICAST，这是什么意思？一般而言，网卡有几种接收数据帧的状态，如 Unicast、Broadcast、Multicast 和 Promiscuous 等。Unicast 是指网卡在工作时的接收目的地址是本机硬件地址的数据帧；Broadcast 是指接收所有类型为广播报文的数据帧；Multicast 是指接收特定的组播报文；Promiscuous 则是通常所说的混杂模式，是指对报文中的目的硬件地址不加任何检查而全部接收的工作模式。对照这几个概念，看看上面的命令输出，可以看

到,正常的网卡应该只接收发往自身的数据报文、广播和组播报文。

对网络使用者来说,浏览网页、收发邮件等都是很平常、很简单的工作,其实在后台这些工作是依靠TCP/IP协议簇实现的。下面从TCP/IP模型的角度来看数据包在局域网内发送的过程:当数据从应用层自上而下传递时,在网络层形成IP数据包,再向下到达数据链路层,由数据链路层将IP数据包分割为数据帧,增加以太网包头,再向下一层发送。需要注意的是,以太网的包头中包含着本机和目标设备的MAC地址,即链路层的数据帧发送时,是依靠48位的以太网地址而非IP地址来确认的,以太网的网卡设备驱动程序不会关心IP数据包中的目的IP地址,它所需要的仅仅是MAC地址。

目标IP的MAC地址又是如何获得的呢?发送端主机会向以太网上的每一台主机发送一份包含目的地的IP地址的以太网数据帧(称为ARP数据包),并期望目的主机回复,从而得到目的主机对应的MAC地址,并将这个MAC地址存入自己的一个ARP缓存中。

当局域网内的主机都通过集线器等方式连接时,一般称为共享式的连接。这种共享式的连接有一个很明显的特点:集线器会将接收到的所有数据向集线器上的每个端口转发,也就是说,当主机根据MAC地址进行数据包发送时,尽管发送端主机告知了目标主机的地址,但这并不意味着在一个网络内的其他主机听不到发送端和接收端之间的通信,只是在正常状况下其他主机会忽略这些通信报文而已。如果这些主机不愿意忽略这些报文,网卡被设置为Promiscuous状态,那么对于这台主机的网络接口而言,任何在这个局域网内传输的信息都是可以被监听的。

5.4.2 网络监听检测与防范

一般来说,网络监听是很难被发现的,因为运行网络监听的主机只是被动地接收在局域网上传输的信息,不会主动与其他主机交换信息,这就导致检测与防范网络监听是比较困难的。

1. 网络监听检测

1) 反应时间

向怀疑有网络监听行为的网络发送大量垃圾数据包,根据各个主机回应的情况进行判断,正常的系统回应的时间应该没有太明显的变化,而处于混杂模式的系统由于对大量的垃圾信息照单全收,因此很有可能回应时间会发生较大的变化。

2) 观测DNS

许多的网络监听软件都会尝试进行地址反向解析,在怀疑有网络监听发生时可以在DNS系统上观测有没有明显增多的解析请求。

3) 利用ping模式进行监测

当一台主机进入混杂模式时,以太网的网卡会将所有不属于它的数据照单全收。按照这个思路,可以这样来操作:假设所怀疑的主机的硬件地址是00:30:6E:00:9B:B9,其IP地址是192.168.1.1;伪造出这样的一种icmp数据包,即硬件地址是不与局域网内任何一台主机相同的00:30:6E:00:9B:B9,而目的地址是192.168.1.1不变。可以设想,这种数据包在局域网内传输会发生以下现象:任何正常的主机会检查这个数据包,比较数据包的硬件地址,如果地址与自己的不同,就不会理会这个数据包;而处于网络监听模式的主

机，则由于其网卡现在是在混杂模式，因此它不会去对比这个数据包的硬件地址，而是将这个数据包直接传到上层，上层检查数据包的 IP 地址，如果符合自己的 IP，就会对这个 ping 的包作出回应。这样，一台处于网络监听模式的主机即被发现。

4）利用 ARP 数据包进行监测

除了使用 ping 进行检测外，目前比较成熟的是利用 ARP 数据包进行检测。这种模式是上述 ping 方式的一种变体。它使用 ARP 数据包替代上述的 icmp 数据包，向局域网内的主机发送非广播方式的 ARP 包，如果局域网内的某个主机响应了这个 ARP 请求，那么就可以判断它很可能就是处于网络监听模式了。这是目前相对而言比较好的检测模式。

值得注意的是，现在因特网上流传着一些基于上面这两种技术的脚本和程序，它们宣称能准确捕捉到局域网内所有进行网络监听的主机。目前来讲，这种说法基本上是不可靠的，因为上述技术在实现中，除了要考虑网卡的硬件过滤外，还需要考虑不同操作系统可能产生的软件过滤。虽然理论上网卡处于混杂模式的系统应该接收所有的数据包，但实际上不同的操作系统甚至相同的操作系统的不同版本在 TCP/IP 的实现上都有自己的一些特点，有可能不会接收这些理论上应该接收的数据包。

相对而言，对发生在本机的网络监听是可以利用一些工具软件来发现的，有兴趣的读者可以参考相关的网站。

2. 网络监听的防范方法

首先，采用加密手段进行信息传输是一个很好的办法，如果监听到的数据都是以密文形式传输的，那么对入侵者来说，即使抓取到了传输的数据信息，意义也不大。这是目前相对而言使用较多的手段之一，在实际应用中往往是指替换掉不安全的采用明文传输数据的服务，如在服务器端用 SSH OpenSSH 等替换 UNIX 系统自带的 Telnet、FTP、RSH，在 Client 端使用 Securecrt、SSHtransfer 替代 Telnet、FTP 等。

目前，除了加密外，使用交换机也是一个应用比较多的方式。不同于工作在第一层的集线器，交换机是工作在第二层，也就是数据链路层。以 Cisco 的交换机为例，交换机在工作时维护着一张 ARP 数据库，其中记录着交换机每个端口所绑定的 MAC 地址，当有数据包发送到交换机上时，交换机会将数据包的目的 MAC 地址与自己维护的数据库内的端口进行对照，然后将数据包发送到“相应的”端口上。注意，不同于集线器的报文广播方式，交换机转发的报文是一一对应的。对二层设备而言，仅有两种情况会发送广播报文，一是数据包的目的 MAC 地址不在交换机维护的数据库中，此时报文向所有端口转发；二是报文本身就是广播报文。由此可以看到，这在很大程度上解决了网络监听的困扰。随着 Dsniff、Ettercap 等软件的出现，交换机的安全性已经面临着严峻的考验。

此外，对安全性要求比较高的公司可以考虑 Kerberos。Kerberos 是一种为网络通信提供可信第三方服务的面向开放系统的认证机制。它提供了一种强加密机制，使 Client 端和 Server 端即使在非安全的网络连接环境中也能确认彼此的身份，而且在双方通过身份认证后，后续的所有通信也是被加密的。在实现中，通过可信的第三方服务器保留与之通信的系统的密钥数据库，仅 Kerberos 和与之通信的系统本身拥有私钥（Private Key），然后通过私钥及认证时创建的 Session Key 来实现可信的网络通信连接。

5.5 缓冲区溢出攻击

5.5.1 缓冲区溢出原理

缓冲区(Buffer)是程序运行期间在内存中分配的连续空间,用于保存包括字符数组在内的各种数据类型。溢出是所填充的数据超出了原有缓冲区的边界,并非法占据了另一端内存区域。缓冲区溢出是指由于填充数据越界而导致程序运行流程的改变,黑客借此精心构造填充数据,让程序转而执行特殊的代码,最终获得系统的控制权。

通过向程序的缓冲区写超出其长度的内容,造成缓冲区的溢出,从而破坏程序的堆栈,使程序转而执行其他指令,以达到攻击的目的。造成缓冲区溢出的原因是程序中没有仔细检查用户输入的参数。例如下面的程序:

```
void function(char * str)
{
    char buffer[16];
    strcpy(buffer,str);
}
```

上面的 strcpy()直接把 str 中的内容复制到缓冲区中。这样,只要 str 的长度大于 16,就会造成缓冲区的溢出,使程序运行出错。存在像 strcpy 这样问题的标准函数还有 strcat()、sprintf()、vsprintf()、gets()和 scanf()等。

当缓冲区溢出时,为什么会导致程序不能正常工作呢? 因为一个程序在内存中是按代码区、数据区和堆栈区顺序存放的。其中,代码区存放程序的机器码和只读数据;数据区存放程序中的静态数据和全局数据;堆栈区存放程序运行时申请的内存空间,用来存放动态数据。图 5.11(a)所示为程序在内存中的分配情况,图 5.11(b)所示为栈中的数据排列顺序。

图 5.11 程序运行时内存分配和堆栈排列

当然,随便向缓冲区中填数据也可造成程序溢出,这时一般只会出现“分段错误”(Segmentation Fault),而不能达到攻击的目的。为了说明该攻击的有效性,下面通过例子来说明溢出攻击的基本原理。

通常 C 语言对边界不进行检查,当输入的数据超出缓冲区的大小时,接下来的数据就会将 EBP(基址寄存器)、RET(返回地址)等覆盖掉,导致程序无法正常执行。以下是另外一种缓冲区溢出。

```
#include <iostream.h>
#include <string.h>
void function(int a)
{
    char buffer[5];
    char *ret;
    ret = buffer + 12;
    *ret += 8;
}
void main()
{
    int x;
    x = 10;
    function(7);
    x = 1;
    cout << x << endl;
}
```

如果不仔细分析这段程序，很可能认为它的执行结果是 1，而不是 10。实际上，这段程序的运行结果是 10，而不是 1。通常函数调用的执行过程大致如下。

(1) 为该函数的形式参数分配内存，并将实际参数的值赋给形式参数。

(2) 将函数返回地址压栈。

(3) 执行被调用函数。

(4) 被调用函数执行结束以后，跳到 RET 指向的指令继续执行。

这段代码的执行过程是：首先为形式参数 a、RET 和 EBP 各分配 4 字节的空间，最后为语句“char buffer[5];”分配内存时，因为对齐的问题需要分配 8 字节的空间。执行“ret＝buffer＋12;”这条语句后，ret 恰好指向 RET，而 RET 的值恰好是函数 function(7)的返回地址，即“x＝1;”这条语句的首地址。但执行“*ret＋＝8;”语句后，就将 RET 的值加上了 8，而“x＝1;”这条语句恰好占用 8 字节。由于 RET 存放函数 function(7)的返回地址，因此 function(7)执行结束后将跳过“x＝1;”这条语句，直接执行“cout << x << endl;”语句。

缓冲区溢出攻击之所以成为一种常见的安全攻击手段，其原因在于缓冲区溢出漏洞太普遍了，并且易于实现，而且缓冲区溢出漏洞给予了攻击者想要的一切：植入并且执行攻击代码。被植入的攻击代码以一定的权限运行有缓冲区溢出漏洞的程序，从而得到被攻击主机的控制权。

5.5.2　缓冲区溢出攻击方法

缓冲区溢出攻击的目的在于扰乱具有某些特权运行程序的功能，从而使得攻击者取得程序的控制权。如果该程序具有足够的权限，那么整个主机就被控制了。为了达到这个目的，攻击者必须达到以下两个目标。

(1) 在程序的地址空间里安排适当的代码。

(2) 通过适当的初始化寄存器和内存，让程序跳转到入侵者安排的地址空间执行。

根据这两个目标可以对缓冲区溢出攻击进行分类，缓冲区溢出攻击分为代码安排和控制程序执行流程两种方法。

1. 在程序地址空间里安排适当代码的方法

(1) 植入法。攻击者向被攻击的程序输入一个字符串，程序会把这个字符串放到缓冲

区里。这个字符串包含的资料是可以在这个被攻击的硬件平台上运行的指令序列。在这里,攻击者用被攻击程序的缓冲区来存放攻击代码。缓冲区可以设在任何地方:栈(stack,自动变量)、堆(heap,动态分配的内存区)和静态资料区。

(2) 利用已经存在的代码。有时攻击者想要的代码已经在被攻击的程序中了,攻击者所要做的只是对代码传递一些参数。例如,攻击代码要求执行 exec ("/bin/sh"),而在 libc 库中的代码执行 exec (arg),其中 arg 是一个指向一个字符串的指针参数,那么攻击者只要把传入的参数指针改为指向/bin/sh 即可。

2. 控制程序转移到攻击代码的方法

所有的这些方法都是在寻求改变程序的执行流程,使之跳转到攻击代码。最基本的就是溢出一个没有边界检查或其他弱点的缓冲区,这样就扰乱了程序的正常执行顺序。通过溢出一个缓冲区,攻击者可以用暴力的方法改写相邻的程序空间,而直接跳过系统的检查。

分类的基准是攻击者所寻求的缓冲区溢出的程序空间类型。原则上可以是任意的空间。实际上,许多的缓冲区溢出是用暴力的方法来寻求改变程序指针的。这类程序的不同之处就是程序空间的突破和内存空间的定位不同。主要有以下 3 种。

(1) 激活记录(Activation Records)。每当一个函数调用发生时,调用者会在堆栈中留下一个活动记录,它包含了函数结束时返回的地址。攻击者通过溢出堆栈中的自动变量,使返回地址指向攻击代码。通过改变程序的返回地址,当函数调用结束时,程序就跳转到攻击者设定的地址,而不是原先的地址。这类缓冲区溢出称为堆栈溢出攻击(Stack Smashing Attack),是目前最常用的缓冲区溢出攻击方式。

(2) 函数指针(Function Pointers)。函数指针可以用来定位任何地址空间。例如,"void (* foo)()"声明了一个返回值为 void 的函数指针变量 foo。所以,攻击者只需在任何空间内的函数指针附近找到一个能够溢出的缓冲区,然后溢出这个缓冲区来改变函数指针。在某一时刻,当程序通过函数指针调用函数时,程序的流程就按攻击者的意图实现了。它的一个攻击范例就是在 Linux 系统下的 superprobe 程序。

(3) 长跳转缓冲区(Longjmp Buffers)。在 C 语言中包含了一个简单的检验/恢复系统,称为 setjmp/longjmp。意思是在检验点设定 setjmp(buffer),用 longjmp(buffer)来恢复检验点。然而,如果攻击者能够进入缓冲区的空间,那么 longjmp(buffer)实际上是跳转到攻击者的代码。像函数指针一样,长跳转缓冲区能够指向任何地方,所以攻击者所要做的就是找到一个可供溢出的缓冲区。一个典型的例子就是 Perl 5.003 的缓冲区溢出漏洞。攻击者首先进入用来恢复缓冲区溢出的长跳转缓冲区,然后诱导进入恢复模式,这样就使 Perl 的解释器跳转到攻击代码上了。

5.5.3 防范缓冲区溢出

在 C 语言中,指针和数组越界不保护是缓冲区溢出的根源,而且在 C 语言标准库中就有许多能提供溢出的函数,如 strcat()、strcpy()、sprintf()、vsprintf()、gets()和 scanf()等。虽然大家都认为缓冲区溢出可以在编程阶段得到避免,但在实际编程操作中却并没有那么简单。这主要在于,有些开发人员没有意识到问题的存在;有些开发人员不愿意使用边界检查,因为这样做会影响到程序的效率和性能。

综合起来,防范缓冲区溢出主要有以下方法。

(1) 编写正确的代码。在开发过程中,尽量使用带有边界检查的函数版本,或者自己进行边界检查是防止缓冲区溢出的基本方法。

(2) 及时安装漏洞补丁。缓冲区溢出是代码中固有的漏洞,除了在开发阶段注意编写正确的代码外,对于用户的一般防范措施就是关闭不必要的端口和服务,并及时安装厂商提供的补丁,这是解决缓冲区溢出问题最有效的方法。

(3) 借助于防火墙阻止缓冲区溢出。在防火墙上过滤特殊的流量也是一种防范的基本方法,但使用防火墙无法阻止来自内部人员的溢出攻击。此外,为了限制黑客溢出成功的权限,以所需要的最小权限运行软件也是一种很好的防范方法。

5.6　注入式攻击

注入式攻击是一种比较常见、危害严重的网络攻击,其主要针对 Web 服务器端的特定数据库系统。注入式攻击的基本特征主要表现在从一个数据库获得未授权的访问与直接检索。注入式攻击的手段是在 Web 访问请求中插入 SQL 语句,针对的是 Web 服务器程序开发过程中的漏洞,如是否做输入数据的合法性检查等。

由于注入式攻击利用的是 SQL 语法,因此这种攻击具有广泛的应用基础。从理论上来讲,对于所有的基于 SQL 的数据库软件,如 Access、SQL Server、Oracle、DB2、MySQL 等,注入式攻击都是有效的攻击方法。当然,根据各种不同的数据库软件,最终的攻击代码也会有一定的区别。

注入式攻击的基本流程如下。

(1) 判断是否存在漏洞。在浏览器地址栏中,输入“http://www.*.*/*.asp?nid=12 and 1=1”,返回正常结果;而输入“http://www.*.*/*.asp? nid=12 and 1=2”,提示 BOF 或 EOF 等信息,则说明该网站存在注入漏洞。

(2) 判断数据库软件的类型。在浏览器地址栏中输入“http://www.*.*/*.asp?nid=12 and user>0”,提示 JET,说明数据库软件是 Access;提示 OLEDB,说明数据库软件是 SQL Server。

(3) 猜测数据库中的表以及表中的字段与字段中的值。在浏览器地址栏中,输入“http://www.*.*/*.asp? nid=12 and (select count(*) from Admin)>0”,返回正常结果,说明数据库中存在 Admin 表;输入“http://www.*.*/*.asp? nid=12 and (select count(admin) from Admin)>0”,返回正常结果,说明 Admin 表中存在 admin 字段;输入“http://www.*.*/*.asp? nid=12 and exists (select id from Admin where id=1)”,返回正常结果,说明 admin 字段中存在 id 为 1 的值。

(4) 猜测用户名及长度。在浏览器地址栏中,输入“http://www.*.*/*.asp?nid=12 and (select top 1 len(username)from Admin)>n”,返回错误结果,说明用户名的长度为 $n-1$;输入“http://www.*.*/*.asp? nid=12 and exists(select id from Admin where id=1 and asc(mid(admin,n,1))=97)”,返回正常结果,由于 97 为字符 a 的 ASCII 码值,说明用户名的第 n 位为 a。

(5) 猜测用户密码及其长度。测试方法类似步骤(4)。

(6) 登录网站后台系统,进一步执行攻击行为。

由于多数网站都使用SQL Server等数据库软件,并且很多程序员在编写程序的时候没有做输入数据的合法性检查,因此注入式攻击成为针对网站系统的常见攻击手段。由于注入式攻击是在Web的输入地址中提交SQL语句,其访问行为与正常Web页面访问没有区别,因此多数防火墙系统无法有效检测注入式攻击。但是,注入式攻击会导致网站出现一些可疑现象,如Web页面混乱、数据内容丢失、访问速度下降等,这些现象都有助于发现注入式攻击。

针对注入式攻击的防范措施主要包括在编写代码时做好数据的合法性检查,增强数据库软件的安全设置,启用Web服务器的审计日志等,从而有效防范注入式攻击行为。

5.7 拒绝服务攻击

DoS(Denial of Service,拒绝服务)攻击是一种既简单又有效的攻击方式。它是针对系统的可用性发起的攻击,通过某些手段使得目标系统或网络不能提供正常的服务。该攻击主要是利用了TCP/IP中存在的设计缺陷,或者操作系统及网络设备的网络协议栈存在的实现缺陷。

一些商业及政府网站都曾经遭受拒绝服务攻击。在2000年2月发生的一次针对某些高利润的站点(如雅虎、易趣等)的拒绝服务攻击持续了近两天,使这些公司遭受了很大的损失,事后这些攻击确定为分布式的拒绝服务攻击。

从攻击技术来看,DoS攻击表现为带宽消耗、系统资源消耗、程序实现上的缺陷、系统策略的修改等几种。带宽消耗是通过网络发送大量信息,用足够的传输信息消耗掉有限的带宽资源。系统资源消耗是向系统发送大量信息,针对操作系统中有限的资源,如进程数、磁盘、CPU、内存、文件句柄等。利用程序实现上的缺陷,对异常行为的不正确处理,通过发送一些非法数据包使系统死机或重启,如Ping of Death。修改或篡改系统策略也可以使得它不能提供正常的服务。

从攻击目标来看,有通用类型的DoS攻击和系统相关的攻击。通用类型的DoS攻击往往是与具体系统无关的,如针对协议设计缺陷的攻击。系统相关的攻击往往与具体的实现有关。最终,所有的攻击都是与系统相关的,因为有些系统可以针对协议的缺陷提供一些补救措施,从而免受此类攻击。

一些典型的DoS攻击有Ping of Death、Teardrop、UDP Flooding、Land、SYN Flooding和Smurf等。

5.7.1 IP碎片攻击

1. IP碎片是如何产生的

链路层具有最大传输单元(MTU)这个特性,它限制了数据帧的最大长度(不同的网络类型都有一个上限值)。以太网的MTU是1500字节,可以用netstat -i命令查看这个值(在Linux下)。如果IP层有数据包要传,而且数据包的长度超过了MTU,那么IP层就要对数据包进行分片(Fragmentation)操作,使每一片的长度都小于或等于MTU。假设要传

输一个 UDP 数据包，以太网的 MTU 为 1500 字节，一般 IP 首部为 20 字节，UDP 首部为 8 字节，数据的净荷(Payload)部分预留是 1500 字节－20 字节－8 字节＝1472 字节。如果数据部分大于 1472 字节，就会出现分片现象。

IP 首部包含了分片和重组所需的信息：

```
| Identification |R|DF|MF| Fragment Offset |<-16>|<3>|<-13>|
```

参数解释：

(1) Identification：发送端发送的 IP 数据包标识字段，是一个唯一值，该值在分片时被复制到每个片中。

(2) R：保留未用。

(3) DF：Dont Fragment，“不分片”位，如果将这一位置 1，IP 层将不对数据包进行分片。

(4) MF：More Fragment，“更多的分片”，除了最后一片外，其他每个组成数据包的片都要把该位置为 1。

(5) Fragment Offset：该片偏移原始数据包开始处的位置。偏移的字节数是该值乘以 8。

了解了分片，也分析了 IP 头的一些信息。下面介绍 IP 碎片是怎样被运用在网络攻击上的。

2. IP 碎片攻击

IP 首部有 2 字节表示整个 IP 数据包的长度，所以 IP 数据包最长只能为 0xFFFF，就是 65 535 字节。如果有意发送总长度超过 65 535 字节的 IP 碎片，一些旧的系统内核在处理的时候就会出现问题，导致崩溃或拒绝服务。另外，如果分片之间偏移量经过精心构造，一些系统就无法处理，导致死机。所以说，漏洞的起因是出在重组算法上。下面通过逐个分析一些著名的碎片攻击程序来了解如何人为制造 IP 碎片以攻击系统。

1) 攻击方式之 Ping of Death

Ping of Death 是利用 ICMP 的一种碎片攻击。攻击者发送一个长度超过 65 535 字节的 Echo Request 数据包，目标主机在重组分片的时候会造成事先分配的 65 535 字节缓冲区溢出，系统通常会崩溃或挂起。尝试把 IP 和 ICMP 首部长度设为 65 535 字节，发送一个数据包：

```
# ping 192.168.0.1 -1 65535
    Error: packet size 65535 is too large. Maximum is 65507
```

一般来说，Linux 自带的 ping 是不允许做这个坏事的。65 507 字节是它计算好的，即 65 535 字节－20 字节－8 字节＝65 507 字节。Windows 2000 下的 ping 数据只允许 65 500 字节大小。所以，必须找另外的程序来发这个数据包。在目前新版本的操作系统中已经修补了这个漏洞。

2) 攻击方式之 jolt2 攻击

jolt2.c 是在一个死循环中不停地发送一个 ICMP/UDP 的 IP 碎片，可以使 Windows 系统的机器死锁。测试没打补丁的 Windows 2000 系统，CPU 利用率会立即上升到 100%，鼠标指针无法移动。

用 Snort 分别抓取采用 ICMP 和 UDP 发送的数据包。发送的 ICMP 包：

```
01/07-15:33:26.974096 192.168.0.9 -> 192.168.0.1
    ICMP TTL:255 TOS:0x0 ID:1109 IpLen:20 DgmLen:29
    Frag Offset:0x1FFE Frag Size:0x9
    08 00 00 00 00 00 00 00 00 …
```

发送的UDP包:

```
01/10-14:21:00.298282 192.168.0.9 -> 192.168.0.1
    UDP TTL:255 TOS:0x0 ID:1109 IpLen:20 DgmLen:29
    Frag Offset:0x1FFE Frag Size:0x9
    04 D3 04 D2 00 09 00 00 61 …a
```

从上面的结果可以看出:分片标志位MF=0,说明是最后一个分片。偏移量为0x1FFE,计算重组后的长度为(0x1FFE * 8) + 29 = 65 549 > 65 535,溢出。

ICMP包:类型为8,代码为0,是Echo Request。校验和为0x0000,程序没有计算校验,所以确切地说,这个ICMP包是非法的。

UDP包:目的端口由用户在命令参数中指定。源端口是目的端口和1235进行OR的结果。校验和为0x0000,与ICMP的一样,没有计算,非法的UDP。净荷部分只有一个字符a。

jolt2应该可以伪造源IP地址,但是源程序中并没有把用户试图伪装的IP地址赋值给src_addr。

jolt2的影响相当大,通过不停地发送这个偏移量很大的数据包,不仅死锁未打补丁的Windows系统,同时也大大增加了网络流量。曾经有人利用jolt2模拟网络流量,测试IDS在高负载流量下的攻击检测效率,就是利用这个特性。

3) 攻击方式之Teardrop

Teardrop是一种IP碎片攻击,也是一种常见的DoS攻击方式。它的攻击方式非常简单:通过发送一些IP分片异常的数据包,在IP包的分片装配过程中,由于分片重叠,计算过程中出现长度为负值,在执行memcpy的时候导致系统崩溃。当网络分组穿越不同的网络时,需要根据网络的最大传输单元来把它们分割成较小的片。早期的Linux系统在处理IP分片重组问题时,尽管对片断是否过长进行检查,但对过短片段却没有进行验证,所以导致了Teardrop形式的攻击。该攻击主要影响Linux和Windows NT/95系统。

如图5.12所示,在Linux 2.0内核中有以下处理:当发现有位置重合时(offset2<end1),将offset向后调到end1(offset=end1),然后更改len2的值,即len2=end2-offset2,此时len2变成了一个小于0的值,会导致以后处理时出现溢出。

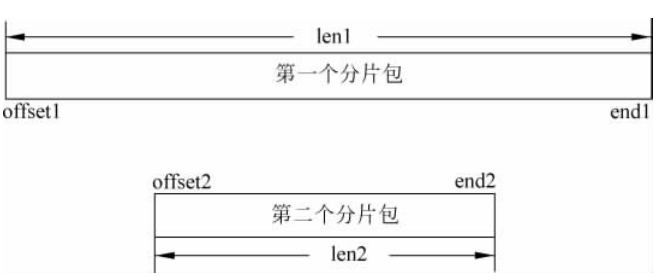

图5.12 异常分片重组

3. 如何防止 IP 碎片攻击

为了防止 IP 碎片攻击，Windows 系统在升级补丁 Service Pack 后可以解决这个问题，目前的 Linux 内核已经不受影响。如果可能，在网络边界上禁止碎片包通过，或者用 IPtables 限制每秒通过碎片包的数目。如果防火墙有重组碎片的功能，确保自身的算法没有问题；否则，受到 DoS 攻击就会影响整个网络。在 Windows 2000 系统中，自定义了 IP 安全策略，并设置了“碎片检查”，以防止 IP 碎片攻击。

在很多路由器上也有“IP 碎片(Fragment) 攻击防御”的设置，网络规模在 150 台左右，建议 IP 碎片值设置在 3000 包/秒。

5.7.2　UDP 洪泛

UDP 洪泛攻击的原理是各种各样的假冒攻击利用简单的 TCP/IP 服务，如 Chargen 和 Echo 来传送毫无用处的占满带宽的数据。通过伪造与某一主机的 Chargen 服务之间的一次 UDP 连接，回复地址指向开着 Echo 服务的一台主机，这样就在两台主机之间生成足够多的无用数据流，导致带宽耗尽的拒绝服务攻击。

关掉不必要的 TCP/IP 服务，或者对防火墙进行配置，阻断来自 Internet 的对这些服务响应的 UDP 请求都可以防范 UDP 洪泛攻击。

5.7.3　SYN 洪泛

SYN 洪泛攻击利用 TCP/IP 连接三次握手过程，打开大量的半开 TCP 连接，使得目标机器不能进一步接受 TCP 连接。每台机器都需要为这种半开连接分配一定的资源，并且这种半开连接的数量是有限制的，达到最大数量时，CPU 满负荷或内存不足，机器就不再接受进来的连接请求，如图 5.13 所示。在 SYN 洪泛攻击中，连接请求是正常的，但是源 IP 地址往往是伪造的，并且是一台不可到达机器的 IP 地址，否则被伪造地址的机器会重置这些半开连接。一般半开连接超时之后会自动清除，所以攻击者的系统发出 SYN 报的速度都要比目标机器清除半开连接的速度要快。任何连接到 Internet 上并提供基于 TCP 的网络服务都有可能成为攻击的目标。这样的攻击很难跟踪，因为源地址往往不可信，而且不在线。

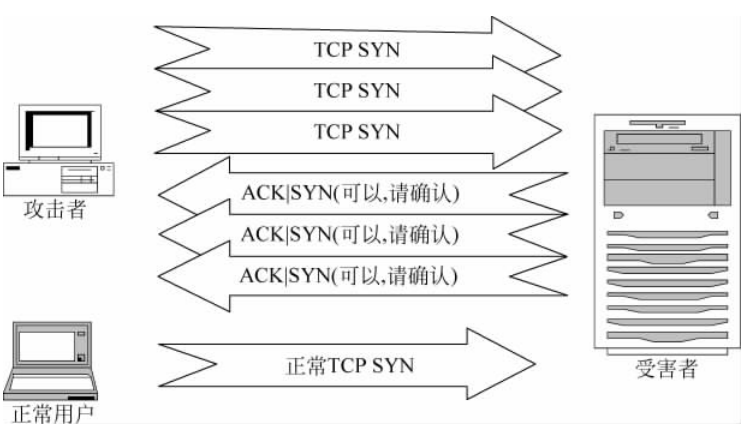

图 5.13　SYN 洪泛攻击示意图

SYN 洪泛攻击的特征是目标主机的网络上出现大量的 SYN 包,而没有相应的应答包;SYN 包的源地址可能是伪造的,甚至无规律可循。

可以在主机和网络上采取措施来防止 SYN 洪泛攻击。防火墙或路由器可以在给定的时间内只允许有限数量的半开连接,入侵检测可以发现这样的 DoS 攻击行为。主机上可以限制 SYN Timeout 的时间。此外,一些操作系统也实现了防止 SYN 洪泛攻击的功能,如 Linux 和 Solaris 使用了一种称为 SYN cookie 的技术来解决 SYN 洪泛攻击:在半开连接队列之外另设置一套机制,使得合法连接得以正常继续。

5.7.4 Smurf 攻击

在 Smurf 攻击中,攻击者向一个广播地址发送 ICMP Echo 请求,并且用受害者的 IP 地址作为源地址,于是广播地址网络上的每台机器响应这些 Echo 请求,同时向受害者主机发送 ICMP Echo Reply 应答。受害者主机会被这些大量的应答包所淹没,如图 5.14 所示。此类攻击还有一个变种叫做 fraggle 或 udpsmurf,使用 UDP 包。例如,攻击者向 7 号端口发送 ICMP Echo 请求,如果目标机器的端口开放,则发送 ICMP Echo Reply,否则产生 ICMP 不可到达消息。该攻击的两个主要特点是使用伪造的数据包和使用广播地址。在该攻击中,不仅被伪造地址的机器受害,目标网络本身也是受害者,因为它们要发送大量的应答包。Smurf 攻击涉及三方:攻击者、中间目标网络和受害者。它以较小的网络带宽资源,通过放大作用,攻击具有较大带宽的受害者系统。

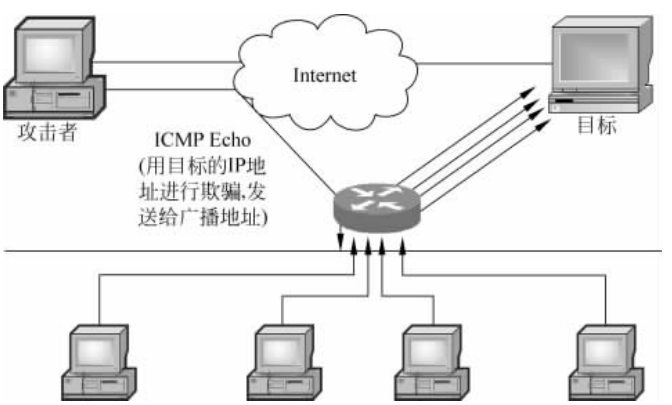

图 5.14 Smurf 攻击示意图

可采取的防范措施如下:

(1) 配置路由器,禁止 IP 广播包进网。

(2) 配置网络上所有计算机的操作系统,禁止对目标地址为广播地址的 ICMP 包响应。

(3) 被攻击目标与 ISP 协商,让 ISP 暂时阻止这些流量。

(4) 对于从本网络向外部网络发送的数据包,本网络应该将其源地址为其他网络的这部分数据包过滤掉。

5.7.5 分布式拒绝服务攻击

传统的拒绝服务是一台机器向受害者发起攻击,分布式拒绝服务(Distributed Denial of Service,DDoS)攻击不仅是一台机器,而是多台机器合作,同时向一个目标发起攻击。DDoS 攻击模型如图 5.15 所示。该攻击过程涉及 3 个层次,即攻击者、主控端和代理端。攻击者

所用的计算机是攻击主控台，可以是网络上的任何一台主机。攻击者操纵整个攻击过程，它向主控端发送攻击指令。主控端是攻击者非法侵入并控制的一些主机，这些主机还分别控制着大量的客户机。主控端主机上面安装了特定的程序，可以接收来自攻击者的特殊指令，并将这些命令发送到代理端。代理端同样也是攻击者侵入并控制的一批主机，它们上面运行攻击程序。在一个特定的时间，主控程序与大量的代理程序通信，代理程序收到指令后就进行攻击。利用客户机/服务器技术，主控程序能在几秒内激活成百上千个代理程序进行攻击。

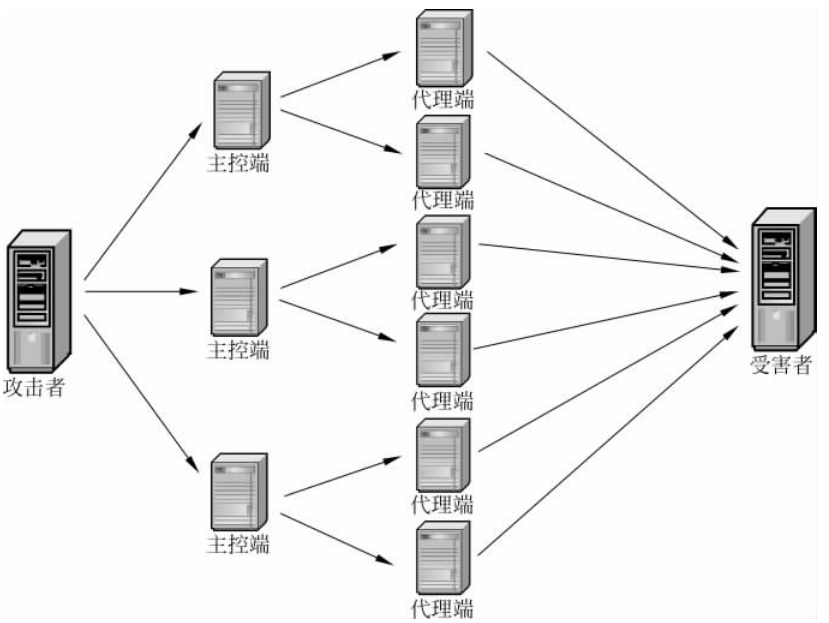

图 5.15　DDoS 拒绝服务攻击原理

DDoS 攻击的主要工具有 TFN(Tribe Flood Network)、TFN2K 和 Stacheldraht 等。

由于 DDoS 攻击具有隐蔽性，到目前为止还没有找到对 DDoS 攻击行之有效的解决方法，因此只能加强安全防范意识，提高网络系统的安全性。主要的防御策略有以下几方面。

(1) 及早发现系统存在的漏洞，及时安装系统补丁程序。对一些系统的重要信息建立和完善备份机制。对一些特权账号的密码设置要谨慎。

(2) 经常检查系统的物理环境，禁止不必要的网络服务。建立边界安全界限，确保输出的包受到正确限制。经常检查系统配置信息，并注意查看每天的安全日志。

(3) 充分利用防火墙等网络安全设备，加固网络的安全性，配置好它们的安全规则，过滤掉所有可能伪造的数据包。

5.8　欺骗攻击与防范

欺骗攻击是利用 TCP/IP 等本身的漏洞而进行的攻击行为。这些攻击包括 IP 欺骗、DNS 欺骗、ARP 欺骗等。欺骗攻击本身不是攻击的目的，而是为实现攻击目标所采取的手

段。欺骗攻击往往基于相互之间的信任关系。两台计算机进行相互通信时,往往需要首先进行认证。认证是网络上的计算机用于相互之间进行识别的过程,经过认证的过程,获准相互交流的计算机之间就建立起相互信任的关系。信任和认证具有逆反关系,即如果计算机之间存在高度信任关系,交流时就不会要求严格的认证。相反,如果计算机之间没有很好的信任关系,交流时就会要求进行严格的认证。

实质上,欺骗就是一种冒充他人身份通过计算机认证骗取计算机信任的攻击方式。攻击者针对认证机制的缺陷,将自己伪装成可信任方,从而与受害者进行交流,最终窃取信息或展开进一步的攻击。欺骗的种类很多,下面具体介绍IP欺骗和ARP欺骗,其他类型的欺骗攻击,感兴趣的读者可以查找相关方面的材料。

5.8.1 IP欺骗攻击与防范

IP欺骗(IP Spoofing)就是伪造某台主机IP地址的技术。通过IP地址的伪装使得某台主机能够伪装成另外一台主机,其实质就是让一台主机扮演另一台主机,而这台主机往往具有某种特权,或者被另外的主机所信任。IP欺骗大多是利用主机之间的信任关系发动的,所以在介绍IP欺骗攻击之前,先说明一下什么是信任关系,以及信任关系的建立。

1. IP欺骗攻击中的信任关系

在UNIX主机中,存在一种特殊的信任关系。假设有两台主机A和B,上面各有一个账户Alice,在使用中会发现,在A上使用,要输入在A上的相应账户Alice,主机A和B把Alice当作两个互不相关的用户,显然有些不方便。为了减少这种不便,可以在主机A和B中建立起两个账户的相互信任关系。

(1) 在A和B的/home/Alice目录中创建.rhosts文件。

(2) 从主机A的home目录中用命令echo "B Alice">~./rhosts实现A和B的信任关系。

这时,从主机B上就能毫无阻碍地使用任何以r开头的远程调用命令,如rlogin、rsh和rcp等,而无须输入口令验证就可以直接登录到A上。这些命令将允许以地址为基础的验证,允许或拒绝以IP地址为基础的存取服务。rlogin是一个简单的客户机/服务器程序,它的作用和Telnet差不多,不同的是Telnet完全依赖口令验证,而rlogin是基于信任关系的验证。它使用了TCP进行传输。当用户从一台主机登录到另一台主机上,并且目标主机信任它,rlogin将允许在不应答口令的情况下使用目标主机上的资源,验证完全基于源主机的IP地址。

2. IP欺骗的原理

IP欺骗通过利用主机之间的正常信任关系来发动。既然A和B之间的信任关系是基于IP地址的,如果能够冒充B的IP,那么就可以使用rlogin登录到A,而不需要任何口令验证。这就是IP欺骗最根本的理论依据,如图5.16所示。但TCP对IP进行了进一步的封装,它是一种相对可靠的协议。下面看一下正常的TCP/IP的会话过程。

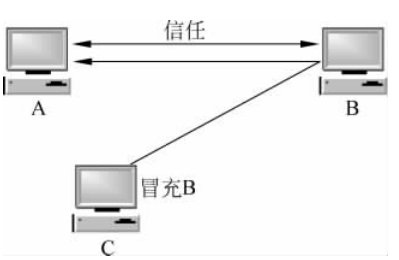

图5.16 IP欺骗示意图

由于TCP是面向连接的协议,因此双方正式传输数据之前需要三次握手来建立连接。假设还是A和B

两台主机进行通信，B 首先发送带有 SYN 标志的数据通知 A 建立 TCP 连接。TCP 的可靠性就是由数据包中的数据序列 SYN 和数据确认标志 ACK 来保证的。B 将 TCP 包头中的 SYN 设为自己本次连接中的初始值(ISN)。

当 A 收到 B 的 SYN 包之后，A 会发送给 B 一个带有 SYN＋ACK 标志的数据段，告知自己的 ISN，并确认 B 发送来的第一个数据段，将 ACK 设置为 B 的 SYN＋1。

当 B 确认收到 A 的 SYN＋ACK 数据包后，将 ACK 设置成 A 的 SYN＋1。A 收到 B 的 ACK 后，连接成功建立，双方即可正式传输数据。图 5.17 所示为 TCP 三次握手的连接过程。

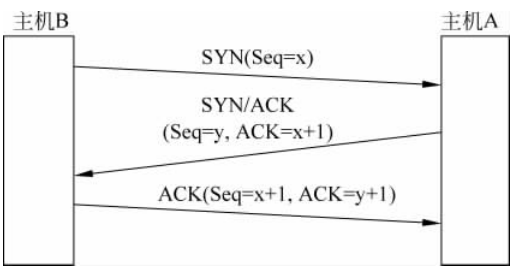

图 5.17　TCP 三次握手的连接过程

很明显，假如冒充 B 对 A 进行攻击，就要先使用 B 的 IP 地址发送 SYN 标志给 A，但是当 A 收到 SYN 标志后，并不会把 SYN＋ACK 发送到攻击者主机上，而是发送到真正的 B 上，这时 IP 欺骗就失败了，因为 B 根本无法发送 SYN 请求。所以要冒充 B，首先要让 B 失去工作能力，也就是所谓的拒绝服务攻击，设法使 B 瘫痪。

前面已经提到，要对目标主机进行攻击，必须知道目标主机使用的数据包序列号。攻击者首先与被攻击主机的一个端口(SMTP 是一个很好的选择)建立起正常连接。通常这个过程被重复若干次，并将目标主机最后所发送的 ISN 存储起来。黑客还需要估计他的主机与被信任主机之间的 RTT 时间(往返时间)，这个 RTT 时间是通过多次统计平均求出的。RTT 对于估计下一个 ISN 非常重要，因为每秒钟 ISN 增加 128 000，每次连接增加 64 000。现在就不难估计出 ISN 的大小了，它是 128 000 乘以 RTT 的一半，如果此时目标主机刚刚建立过一个连接，那么再加上一个 64 000。在估计 ISN 大小后，立即就开始攻击。当黑客虚假的 TCP 数据包进入目标主机时，根据估计的准确程度会发生以下不同情况。

(1) 如果估计的序列号是准确的，进入的数据将被放置在接收缓冲区以供使用。如果估计的序列号小于期待的数字，那么将被放弃。

(2) 如果估计的序列号大于期待的数字，并且在滑动窗口(缓冲)之内，那么该数据被认为是一个未来的数据，TCP 模块将等待其他缺少的数据。

(3) 如果估计的序列号大于期待的数字，并且不在滑动窗口之内，那么 TCP 将会放弃该数据，并返回一个期望获得的数据序列号。

攻击者伪装成被信任的主机 IP，然后向目标主机的 513 端口(rlogin)发送连接请求。目标主机立刻对连接请求作出响应，并更新 SYN＋ACK 确认包给被信任主机，因为此时被信任主机仍然处于瘫痪状态，它当然无法收到这个包，紧接着攻击者向目标主机发送 ACK 数据包，该包使用前面估计的序列号加 1。如果攻击者估计正确，则目标主机将会接收该

ACK,连接就可正式建立。这时就可以将 cat '++'>>~/.rhosts 命令发送过去,这样完成本次攻击后就可以不用口令直接登录到目标主机上。如果达到这一步,一次完整的 IP 欺骗就完成了。黑客已经在目标主机上得到了一个 Shell 权限,接下来就是利用系统的溢出或错误配置扩大权限。当然,黑客的最终目的还是获得服务器的 root 权限。

从上面的攻击过程可以看出,一般地,一个 IP 欺骗攻击的整个步骤如下。

(1) 让被信任主机的网络暂时瘫痪,以免对攻击造成干扰。

(2) 连接到目标主机的某个端口,猜测 ISN 基值和增加规律。

(3) 把源地址伪装成被信任主机,发送带有 SYN 标志的数据段请求连接。

(4) 等待目标主机发送 SYN+ACK 包给已经瘫痪的主机。

(5) 再次伪装成被信任主机向目标主机发送 ACK,此时发送的数据段带有预测目标主机的 ISN+1。

(6) 连接建立,发送命令请求。

3. IP 欺骗的防范

对于来自网络外部的欺骗,防范的方法很简单,只需要在局域网的对外路由器上加一个限制设置就可以实现了,即在路由器的设置里面禁止运行由外部来的但声称来自于网络内部的信息包。

对于来自局域网外部的 IP 欺骗攻击,也可以通过防火墙进行防范。但对于来自内部的攻击,通过设置防火墙起不了什么作用,这时应该注意内部网的路由器是否支持内部接口。如果路由器支持内部网络子网的两个接口,则必须提高警惕,因为它很容易受到 IP 欺骗。

通过对信息包的监控来检查 IP 欺骗攻击是非常有效的方法,使用 netlog 等信息包检查工具对信息的源地址和目的地址进行验证,如果发现了信息包来自两个以上的不同地址,则说明系统有可能受到了 IP 欺骗攻击。

5.8.2 ARP 欺骗攻击与防范

在局域网中,实际传输的数据是按照帧进行传输的,帧里面有目标主机的 MAC 地址。一台主机要与另一台主机进行直接通信,必须知道目标主机的 MAC 地址,目标 MAC 地址就是通过 ARP(Address Resolution Protocol,地址解析协议)获得的。地址解析,就是主机在发送帧之前将目标 IP 地址转换成目标 MAC 地址的过程。ARP 的基本功能就是通过目标设备的 IP 地址,查询目标设备的 MAC 地址,以保证通信的顺利进行。

ARP 欺骗攻击是针对 ARP 的一种攻击技术,可以造成内部网络的混乱,让某些被欺骗的计算机无法正常访问网络,让网关无法同客户机正常通信。一般来说,IP 地址的冲突可以通过多种方法和手段来避免,而 ARP 工作在最底层,当 ARP 缓存出错时,系统并不会判断 ARP 缓存正确与否,无法像 IP 冲突那样给出提示,而且很多黑客工具可以随时发送 ARP 欺骗数据包和 ARP 恢复数据包,这样就可以实现在一台普通计算机上通过发送 ARP 数据包的方式来控制网络中任何一台计算机的网络连接,甚至还可以直接对网关进行攻击,让所有连接网络的计算机都无法正常上网。

1. ARP 欺骗攻击的原理

当某机器 A 要向机器 B 发送报文,会查询本地的 ARP 缓存表,找到 B 的 IP 地址对应

的 MAC 地址后就进行数据传输，如果未找到，则广播一个 ARP 请求报文，请求 IP 地址为 B 的主机应答其物理地址。网上所有主机包括 B 都收到 ARP 请求，但只有主机 B 响应，于是向 A 主机发送一个 ARP 响应报文，其中就包含 B 的 MAC 地址。A 接收到 B 的应答后，就会更新本地 ARP 缓存，接着使用这个 MAC 地址发送数据。因此，本地高速缓存的这个 ARP 表是本地网络畅通的基础，并且这个缓存是动态的。

ARP 欺骗攻击就是通过伪造 IP 地址和 MAC 地址实现 ARP 欺骗，过程如下。

(1) 假设有这样一个网络，包含一个交换机，连接了 3 台机器，依次是计算机 A、B、C。

① A 的地址为 IP：192.168.1.1，MAC：AA-AA-AA-AA-AA-AA。

② B 的地址为 IP：192.168.1.2，MAC：BB-BB-BB-BB-BB-BB。

③ C 的地址为 IP：192.168.1.3，MAC：CC-CC-CC-CC-CC-CC。

(2) 正常情况下，在 A 计算机上运行 ARP-A，查询 ARP 缓存表，应该出现如下信息：

```
Interface: 192.168.1.1 on Interface 0x1000003
Internet Address Physical Address Type
192.168.1.3 CC-CC-CC-CC-CC-CC dynamic
```

(3) 在计算机 B 上运行 ARP 欺骗程序，发送 ARP 欺骗包。B 向 A 发送一个伪造的 ARP 应答，这个应答中的数据为：发送方 IP 地址是 192.168.1.3(C 的 IP 地址)，MAC 地址是 DD-DD-DD-DD-DD-DD(C 的 MAC 地址本来应该是 CC-CC-CC-CC-CC-CC)。当 A 接收到 B 伪造的 ARP 应答，就会更新本地的 ARP 缓存。A 不知道这是从 B 发过来的，A 这里只有 192.168.1.3(C 的 IP 地址)和无效的 MAC 地址 DD-DD-DD-DD-DD-DD。

(4) 在 A 计算机上运行 ARP-A 查询 ARP 缓存信息，原来正确的信息现在也出现了错误。

```
Interface: 192.168.1.1 on Interface 0x1000003
Internet Address Physical Address Type
192.168.1.3 DD-DD-DD-DD-DD-DD dynamic
```

(5) 当 A 计算机访问 C 计算机时，MAC 地址会被 ARP 协议错误地解析为 DD-DD-DD-DD-DD-DD。

当局域网中的一台机器反复向其他机器，特别是网关发送这样无效的假冒 ARP 应答信息包，严重的阻塞就会开始。由于网关 MAC 地址错误，因此从网络中计算机发来的数据无法正常发送到网关，自然无法正常上网，就造成了无法访问外网的问题。另外，由于很多时候网关还控制着局域网，这时 LAN 访问也就出问题了。

2. ARP 攻击防护

目前，对于 ARP 攻击防护主要有两种方法：绑定 IP 和 MAC，使用 ARP 防护软件。

1) 静态绑定

ARP 攻击防护最常用的方法就是做 IP 和 MAC 的静态绑定，在局域网内把主机和网关都做 IP 和 MAC 绑定。欺骗是通过 ARP 的动态实时的规则欺骗内网机器，所以把 ARP 全部设置为静态，可以解决对内网计算机的欺骗。同时在网关也要进行 IP 和 MAC 地址的静态绑定，这样双向绑定才比较保险。

IP 和 MAC 静态绑定可以通过命令“arp -s IP MAC 地址”来实现，如 arp -s 192.168.1.1 AA-AA-AA-AA-AA-AA。

当然,对于网络中的每台主机都做静态绑定,工作量非常大,而且在计算机每次启动以后都必须重新绑定,因此操作上不是很方便。

2) 使用 ARP 防护软件

ARP 类防护软件的工作原理是过滤所有的 ARP 数据包,对每个 ARP 应答进行判断,只有符合规则的 ARP 包才会被进一步处理,这样就防止了计算机被欺骗。同时对每个发出去的 ARP 应答都进行检测,只有符合规则的 ARP 包才会被发送出去,这样就实现了对发送攻击的拦截。例如,360ARP 防火墙就可以实现该功能。

习 题 5

一、选择题

1. (　　)是使计算机疲于响应这些经过伪装的不可到达客户的请求,从而使计算机不能响应正常的客户请求等,达到切断正常连接的目的。

A. 包攻击　　B. 拒绝服务攻击

C. 缓冲区溢出攻击　　D. 口令攻击

2. (　　)就是要确定你的 IP 地址是否可以到达,运行哪种操作系统,运行哪些服务器程序,是否有后门存在。

A. 对各种软件漏洞的攻击　　B. 缓冲区溢出攻击

C. IP 地址和端口扫描　　D. 服务型攻击

3. 分布式拒绝服务攻击(DDoS) 分为 3 层:(　　)、主控端、代理端。三者在攻击中扮演着不同的角色。

A. 其他　　B. 防火墙　　C. 攻击者　　D. 受害主机

4. 有一种称为嗅探器(　　) 的软件,它是通过捕获网络上传送的数据包来收集敏感数据,这些数据可能是用户的账号和密码,或者一些机密数据等。

A. softice　　B. Unicode　　C. W32Dasm　　D. Sniffer

5. 攻击者在攻击之前的首要任务就是要明确攻击目标,这个过程通常称为(　　)。

A. 安全扫描　　B. 目标探测　　C. 网络监听　　D. 缓冲区溢出

6. 从技术上来讲,网络容易受到攻击的原因主要是由于网络软件不完善和(　　)本身存在安全缺陷造成的。

A. 网络协议　　B. 硬件设备　　C. 操作系统　　D. 人为破坏

7. 每当新的操作系统、服务器程序等软件发布之后,黑客就会利用(　　)寻找软件漏洞,从而达到导致计算机泄密、被非法使用,甚至崩溃的目的。

A. IP 地址和端口扫描　　B. 口令攻击

C. 各种软件漏洞攻击程序　　D. 服务型攻击

8. (　　)攻击是指借助于客户机/服务器技术,将多个计算机联合起来作为攻击平台,对一个或多个目标发动 DoS 攻击,从而成倍地提高拒绝服务攻击的威力。

A. 分布式拒绝服务　　B. 拒绝服务

C. 缓冲区溢出攻击　　D. 口令攻击

9.（　　）是一种破坏网络服务的技术，其根本目的是使受害主机或网络失去及时接收处理外界请求，或者及时回应外界请求的能力。

A. 包攻击　　B. 拒绝服务

C. 缓冲区溢出攻击　　D. 口令攻击

二、填空题

1. 分布式拒绝服务攻击的英文缩写是_____。

2. 窃听与分析网络中传输数据包的程序通常称为_____。

3. _____是一种既简单又有效的攻击方式，通过某些手段使得目标系统或网络不能提供正常的服务。

4. _____是针对 ARP 的一种攻击技术，可以造成内部网络的混乱，让某些被欺骗的计算机无法正常访问网络。

5. _____是一种比较常见、危害严重的网络攻击，它主要针对 Web 服务器端的特定数据库系统。

三、简答题

1. 什么是目标探测？目标探测的方法主要有哪些？

2. 从整个信息安全角度来看，目前扫描器主要有哪几种类型？

3. 如何有效防止端口扫描？

4. 网络监听的主要原理是什么？

5. 如何检测网络监听？如何防范网络监听？

6. 指出下述程序段存在的问题，并修改它。

```
char str[10];
char bigstr[20];
……
while(scanf("%20s",bigstr)!=NULL)
{
    bigstr[20]='\0';
    strcpy(str,bigstr);
    ……
}
```

7. 下面的程序是一个缓冲区溢出演示程序，请编译和执行一下，逐渐增加输入字符个数，分析程序执行结果。如何执行 hacker 函数？

```
#include <stdio.h>
#include <string.h>
void function(const char *input)
{
  char buffer[5];
  printf("my stack looks:\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n\n");
  strcpy(buffer,input);
  printf("%s\n",buffer);
  printf("Now my stack looks like:\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n\n");
}
void hacker(void)
{
  printf("Oh,I've been hacked!\n");
```

```
}
int main(int argc,char * argv[])
{
  printf("address of function = %p \n",function);
  printf("address of hacker = %p \n",hacker);
  function(argv[1]);
  return 0;
}
```

提示:

(1) 在 Visual C++环境中,由于 Debug 模式包含了对栈问题进行检测的操作,因此需要在 Release 模式下编译和运行。

(2) 根据屏幕显示结果找到 EBP 和 RET 的地址。

(3) 为了能使程序执行 hacker 函数,可编写一段名为 hacker.pl 的 pearl 脚本。

```
$arg = "aaaaaaaaa…".hacker 函数地址";
$cmd = "该程序文件名", $arg;
system( $cmd);
pearl hacker.pl
```

这样,程序就可能会执行 hacker 函数(取决于所使用的编译器)。

8. 什么是拒绝服务(DoS) 攻击? 什么是分布式拒绝服务(DDoS)攻击?

9. 如何有效防范 DDoS 攻击?

10. 什么是欺骗攻击? 简述欺骗攻击的原理。

11. IP 欺骗主要是针对 UNIX 操作系统的,在 Windows 操作系统中有没有 IP 欺骗的问题?