第5章

TCP/IP体系的协议安全

随着以 Internet 为主的互联网络的广泛应用，TCP/IP 体系成为目前计算机网络的基础，IP 网络已基本成为现代计算机网络的代名词。然而，由于当初设计 TCP/IP 体系时存在的局限性以及随后信息技术对 IP 网络的依赖性，致使目前 IP 网络存在的安全问题日渐突出，各类安全隐患日益严重。本章以 IPv4 为基础，首先简要介绍 TCP/IP 体系的分层结构和各层的功能定义以及 TCP/IP 体系的主要安全问题，然后联系实际应用，重点介绍 TCP/IP 协议栈中 ARP、DHCP、TCP、DNS 等子协议的安全问题及目前行之有效的安全防范方法。


视频讲解

5.1 TCP/IP 体系

OSI 参考模型的研究初衷是希望为网络体系与协议发展提供一种国际标准。但是，Internet 在全球范围的飞速发展将 TCP/IP 体系推向了研究和应用的前台。在 TCP/IP 协议栈中已集成了大量的子协议并以 Internet 标准发布，同时随着 Internet 应用的不断发展，新的子协议及对已有协议的升级版本也将不断出现，并成为 TCP/IP 协议栈的成员。本节将简要介绍 TCP/IP 体系的分层特点及各层次的功能划分。

5.1.1 TCP/IP 体系的分层特点

ARPAnet 是应用最早的计算机网络类型之一，现代计算机网络的许多概念源自 ARPAnet。ARPAnet 是由美国国防部资助的一个研究性网络，当初它通过租用的电话线将几百所大学和政府部门的计算机设备连接起来，要求通过一种灵活的网络体系结构实现不同设备、不同网络的互联和互通。后来，卫星通信系统和无线电通信系统得到发展并应用到 ARPAnet 中，ARPAnet 最初开发的网络协议使用在通信可靠性较差的通信子网中时出现了问题，这就导致了 TCP/IP 协议的出现。

TCP/IP 开始仅有两个协议：传输控制协议(Transfer Control Protocol，TCP)和网际

协议(Internet Protocol,IP)。后来,TCP/IP 演变为一种体系结构,即 TCP/IP 参考模型。现在的 TCP/IP 已成为一个工业标准的协议集,它最早应用于 ARPAnet。因为当时的 ARPAnet 要求在任何条件下甚至是战争中都可以正常运行,这就决定了运行 TCP/IP 的网络具有很好的兼容性,并可以使用铜缆、光纤、微波以及通信卫星等多种链路。同时,在 TCP/IP 网络中所有的数据都以分组的形式传输。

与 OSI 参考模型不同,TCP/IP 模型由应用层(Application Layer)、传输层(Transport Layer)、网际层(Internet Layer,也称为 Internet 层)和网络接口层(Network Interface Layer)4 部分组成,如图 5-1 所示。这 4 层大致对应 OSI 参考模型的 7 层。但与 OSI 模型不同的是,TCP/IP 协议栈更加侧重于互联设备间的数据传送,而不是严格的功能层次的划分。

为了便于对 TCP/IP 体系的理解,可以将 TCP/IP 体系分为协议层和网络层两层,如图 5-2 所示。其中,协议层包括 TCP/IP 体系的上面两层(应用层和传输层),具体定义了有关网络通信协议的类型;而网络层包括 TCP/IP 模型的下面两层(网际层和网络接口层),具体定义了网络的类型(如局域网和广域网)和设备之间的路径选择。

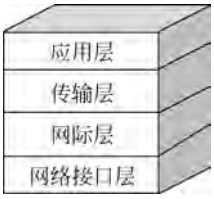

图 5-1　TCP/IP 体系

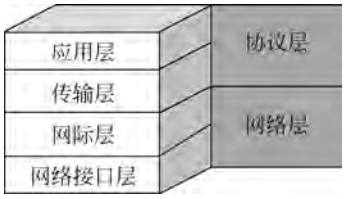

图 5-2　将 TCP/IP 体系划分为协议层和网络层

TCP/IP 是一个协议簇或协议栈,它是由多个子协议组成的集合。图 5-3 列出了 TCP/IP 体系中包括的一些主要协议以及与 TCP/IP 体系的对应关系。理解这个图中的结构(尤其是每一层对应的协议)对于后面的学习非常重要。

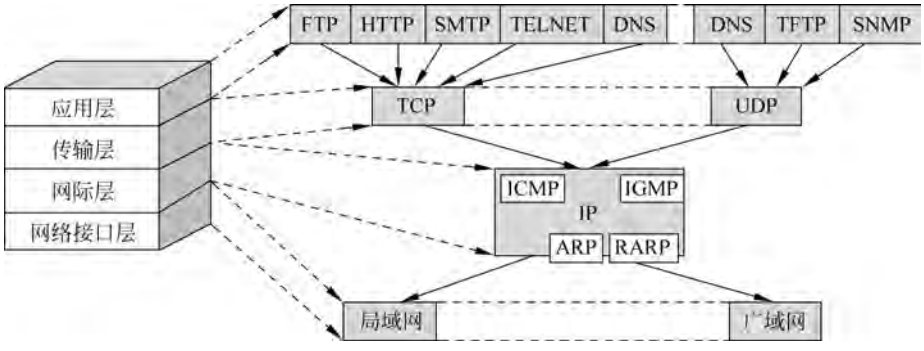

图 5-3　TCP/IP 体系中的主要协议及与各层的对应关系

5.1.2　TCP/IP 各层的主要功能

TCP/IP 体系也称为 TCP/IP 参考模型,该模型从下到上共分为网络接口层、网际层、传输层和应用层,共 4 个子层。各层的主要功能如下。

1. 网络接口层

在 TCP/IP 参考模型中，网络接口层属于最低的一层，它负责通过网络发送和接收分组。由于 TCP/IP 参考模型并没有明确规定在网络接口层应该使用哪些设备、网络和协议，这就带来了以下的好处：一是对于网际层及以上各层来说网络接口层是透明的，网际层及以上各层在功能实现过程中不需要考虑网络接口层使用什么类型的网络、设备或协议；二是有利于 TCP/IP 网络的发展。由于在 TCP/IP 参考模型中，网络接口层的定义是空白的，所以已有的各种类型的物理网络都可以作为 TCP/IP 的网络接口层存在，如目前已经使用的电路交换机、分组交换网（如 X.25、帧中继等）和局域网（如以太网、令牌网、光纤分布式数据接口等）。

2. 网际层

网际层也称为“Internet 层”，它相当于 OSI 参考模型网络层的无连接网络服务。网际层的任务是允许位于同一网络或不同网络中的两台主机之间以分组的形式进行通信。更具体地讲，网际层提供了以下服务功能：一是处理从传输层接收的报文段发送请求，然后将报文段封装到 IP 数据报中，并根据源主机和目的主机的 IP 地址来填充报头，然后根据目的主机的 IP 地址选择一条链路将封装后的 IP 数据报发送出去；二是处理从网络接口层接收到的由其他主机发送过来的数据报，根据数据报中的目的地址决定这个数据报是发送给本主机的还是要发送给其他主机的，如果是发送给本主机的，则在去掉报头信息后提交给传输层，如果是发送给其他主机的，则根据数据报的目的 IP 地址选择一条链路进行转发；三是当本主机连接两个不同的网络（称为“子网”）时，对接收到的数据报进行路由选择和转发，并进行流量控制和阻塞管理。

TCP/IP 参考模型中网际层的协议是唯一的，即 IP 协议。IP 协议是一个面向非连接的、不可靠的数据报传输服务协议，所以网际层的 IP 协议提供了一种“尽力而为”（Best-Effort）的服务。IP 协议的协议数据单元（Protocol Data Unit，PDU）称为分组，所以 TCP/IP 网络也称为分组交换网络。

3. 传输层

在 TCP/IP 参考模型中，传输层位于网际层与应用层之间，其设计目标是：允许在源和目的主机的对等体之间进行会话，负责会话对等体的应用进程之间的通信。TCP/IP 参考模型的传输层功能类似于 OSI 参考模型传输层的功能。

在 TCP/IP 参考模型的传输层中定义了两个端到端的传输协议：传输控制协议（TCP）和用户数据报协议（UDP）。

其中，TCP 协议是一个可靠的、面向连接的协议，它实现了将一台主机发送出去的字节流（Byte Stream）无差错地传输到目的主机。在这一过程中，发送主机的传输层首先把从应用层接收到的数据流划分成为多个小的字节段（Byte Segment），并对每一个字节段进行编号。然后，每一个字节段可以通过不同的路径到达目的主机，如果某一个字节段在传输过程中出错或丢失，则要求发送端进行重传。当目的主机接收到字节段后，根据其编号重组为原来的字节流，并提交给应用层进行处理。TCP 协议还负责处理流量控制，当发送端的发送速率与接收端的接收速度不匹配时（一般是发送速率大于接收速率），协调收、发双方的速率，以确保字节段的可靠传输。

UDP是一个不可靠的、面向非连接的传输层协议，主要用于不要求分组顺序到达的传输，分组的先后顺序检查与排列由应用层的应用程序完成。同时，UDP主要应用于“快速交付比精确交付更加重要”的应用，如语音传输、视频传输等。

4. 应用层

应用层属于TCP/IP参考模型的最高层。应用层主要包括根据应用需要开发的一些高层协议，如Telnet、FTP、SMTP、DNS、SNMP、HTTP等。而且，随着网络应用的不断发展，新的应用层协议还会不断出现。

需要说明的是，在OSI参考模型中，在传输层之上还定义了会话层和表示层，而在TCP/IP参考模型中却没有这两层。这是因为在当初设计时，研究人员认为OSI参考模型的高层划分过于复杂，而且每一层的功能设计并不明确或过于单一，这样在设计TCP/IP参考模型时就去掉了这两层。从目前的应用来看，TCP/IP参考模型当初的这种设计是正确的。

5.1.3　TCP/IP网络中分组的传输示例

在掌握了TCP/IP参考模型的分层特点及各层的功能后，下面通过一个具体的实例介绍TCP/IP网络中分组的传输过程，网络拓扑如图5-4所示。

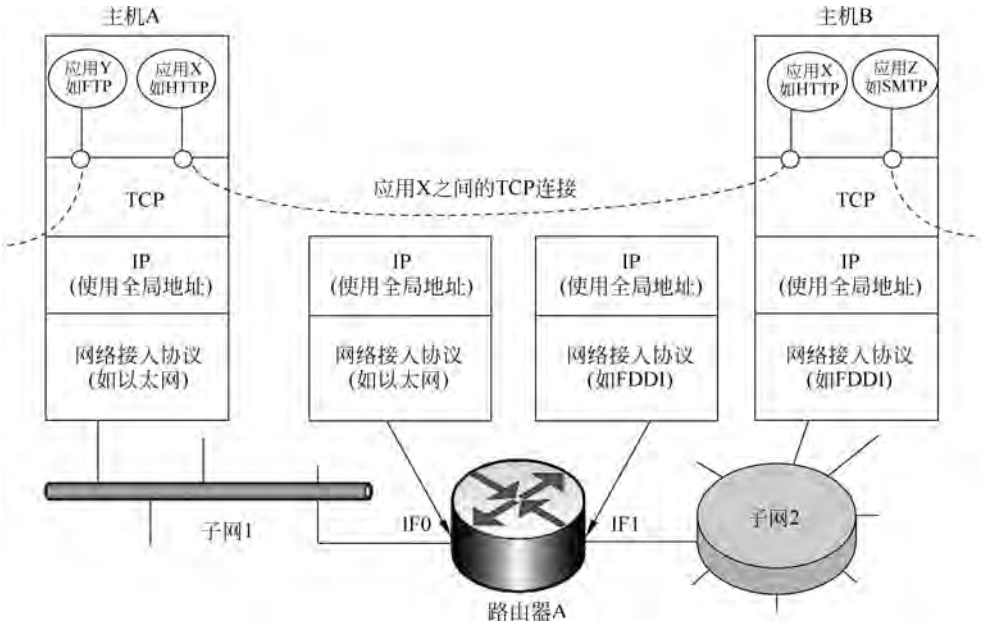

图5-4　TCP/IP网络中数据的传输过程

1. 重要概念

在如图5-4所示的通信过程中，涉及一些关键技术和概念。为便于对操作过程的描述，下面对一些重要概念进行简要介绍。

(1) 子网。一个大型的通信网络由多个子网(Subnetwork)组成，每一个子网属于某一

种特定类型的网络，如局域网中的以太网、令牌环网、FDDI，广域网中的 x.25、帧中继等。不同子网之间一般需要路由器进行连接，由路由器负责 IP 分组在不同子网之间的转发。

(2) 网络接入协议。当计算机接入网络中时，必须使用这一子网中规定的接入协议。通过网络接入协议，可以让一台主机将数据通过子网发送到其他的主机。例如，当计算机接入以太网时，就需要使用以太网的接入协议，通过主机的 MAC 地址进行数据的转发。

(3) 路由器。它是连接不同子网的设备，一台路由器相当于一个中继站，将一个 IP 分组从某一子网中的一台主机通过一个或多个子网发送到目的主机。路由器转发分组的依据是位于每一台路由器中的路由表。当源主机向目的主机发送数据时，为了保证数据能够正确到达，其首要条件是每一台路由器中的路由表必须是完善的，即在任何一台路由器中都有同一网络中其他子网的路由信息。路由表就好像电话查号台的电话号码簿，如果要保证能够通过查号台查到用户需要的电话，前提条件是电话号码簿中的记录是完整的。路由器工作在 TCP/IP 参考模型的网际层。

(4) 全局地址。对于 Internet 等互联网络，每一台主机必须拥有一个全网唯一的 IP 地址作为其身份的唯一标识，这个 IP 地址称为全局地址。当源主机发送数据到目的主机时，源主机首先要知道目的主机的 IP 地址。

(5) 端口。主机中的每一个进程必须具有一个在本主机中唯一的地址，这个地址称为端口(Port)。通过端口，端到端的协议(如 TCP)才能够将数据正确地交付给相应的进程。IP 地址确定了网络中唯一的一台主机，而端口确定了主机中唯一的一个进程。

2. 操作过程

如图 5-4 所示，下面简要描述主机 A 与主机 B 进程之间的通信过程，即数据在主机 A 与主机 B 之间的转发过程。假设通信中使用了 TCP 协议，主要过程如下。

(1) 假设主机 A 中的某一个应用程序(进程)要向主机 B 发送数据，这时主机 A 中的这个进程将在本机中获得一个端口，之后这个进程就使用这个确定的端口进行通信，直到本次通信过程结束，该端口被释放。由于 TCP 是一个可靠的、面向连接的通信协议，所以在主机 A 与主机 B 之间正式传输数据之前，主机 B 也需要为这一次通信过程建立一个唯一的端口。

(2) 主机 A 上的进程通过端口将字节流(数据)交给 TCP 协议，TCP 协议根据网络中的约定将字节流划分为字节段，即将大块数据划分为小块数据。然后给每一个字节段添加控制信息，即 TCP 首部，添加了 TCP 首部后的字节段称为 TCP 报文段。TCP 首部主要包括以下内容。

① 目的端口(Destination Port)。主机 B 上对应的进程使用的端口，这个端口是在主机 A 与主机 B 正式发送数据之前双方协商建立的。主机 B 在接收到 TCP 报文段时，根据端口交付给对应的进程。主机 B 除了与主机 A 之间的这一进程通信之外，还有可能与主机 A 或其他主机的不同进程之间在同时进行通信。

② 序号(Sequence Number)。根据在字节流中位置的先后顺序对字节段进行编号。编号的目的之一是每一个字节段单独选择自己的路由在网络中传输，目的之二是当某一个字节段在传输过程中丢失或出错时，接收方可以让发送方重传该字节段，而不需要重传整个字节流。

③ 校验和(Checksum)。校验和是对每一个字节段(不是报文段，因为不包括 TCP 首

部)利用某一函数运行产生的值。当接收端(主机 B)接收到该字节段时,也会使用相同的函数进行运算,并将运算值与校验和进行比较,如果相同,说明该字节段在传输过程中没有出错,否则需要让对方进行重传。

(3) 主机 A 将 TCP 报文段下传给 IP 层,并要求它将数据发送到主机 B。这时主机 A 会添加一个 IP 首部,IP 首部包括了源主机(主机 A)的 IP 地址和目的主机(主机 B)的 IP 地址。添加了 IP 首部的数据称为 IP 数据报或 IP 分组。

(4) 当 IP 分组到达网络接口层时,网络接口层又加上自己的首部,即网络首部,这时数据进入了第一个子网(子网 1)。网络首部由接入网络类型确定,如以太网、令牌环、FDDI 等。但在网络首部中一般要包含以下的内容。

① 目的子网地址。子网必须知道应将数据帧发送给哪一个相连设备,如路由器 A 上物理接口 IF0 的物理地址。

② 设施请求。网络接入协议可能请求使用特定的子网设施,如优先级等。

将添加了网络首部的数据单元称为网络级分组或数据帧。

(5) 数据帧到了路由器 A 后,首先被去掉网络首部,得到 IP 数据报,并根据 IP 首部的信息得知目的主机的 IP 地址。然后在路由表中查询该 IP 地址,如果找到对应的表项,则根据表项中的信息(其中主要包括路由器上对应的物理接口,如 IF1)将该 IP 数据报转发出去。为了实现这一过程,在路由器 A 上还需要根据下一子网的类型添加网络首部。如果网络中存在多个子网,连接不同子网的路由器都要进行类似于路由器 A 的操作。

(6) 当数据到达主机 B 后,其操作过程与主机 A 的相好相反。在每一层都要去掉相应的首部,并进行相应的约定操作(如校验等),然后将数据交付给上层,直到将原始数据(字节流)交付给指定的进程。

在 TCP/IP 参考模型中,每一层的数据称为协议数据单元(PDU),例如,TCP 报文段也称为 TCP PDU。在数据发送端,在每一层添加首部信息的过程称为数据封装,如图 5-5 所示。在数据接收端,每一层去掉首部信息的过程称为数据解封。

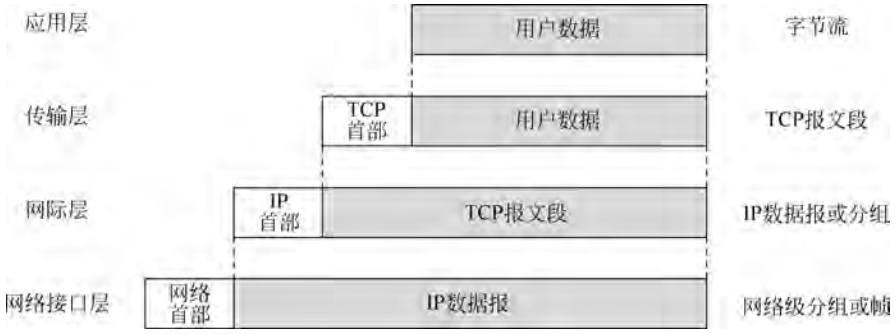

图 5-5　TCP/IP 网络中数据的封装过程

5.2　TCP/IP 体系的安全

TCP/IP 是目前互联网遵循的分层模型,受历史环境、设计要求以及设计者自身因素等方面的影响,协议和应用在设计与实现中都会存在漏洞和缺陷,成为攻击者实施攻击行为的

选择和依据。本节立足于漏洞和缺陷的产生根源，将 TCP/IP 体系的安全划分为针对头部的安全、针对协议实现的安全、针对验证的安全和针对流量的安全 4 个方面进行介绍。

5.2.1 针对头部的安全

不管是计算机应用程序还是网络通信，不同用户、程序和进程都是根据头部信息识别数据类型和来源，并根据头部字段约定对数据进行处理。每一类应用、每一个协议、相同协议在不同层的实现，其头部都不相同。

基于头部的漏洞是指实际的协议头部与标准之间发生了冲突。例如，对于某一协议，其头部每个字段都有严格的字义（如字段长度、允许填充的内容等），但攻击者可以构造一个特殊的头部，使其字段内容不按照协议要求来设置，出现无效值，这样就形成了一个针对头部字段的攻击行为。

在 TCP/IP 体系中，每一层从其上层接收到数据后都要给它添加一个本层的头部，从而形成本层的协议数据单元（PDU）。PDU 由本层头部（添加的协议）和数据（上层的 PDU）组成，其中头部就是由本层执行的协议功能，以便于与其另一端的同层（对等层）之间进行通信。基于这一实现原理，攻击者可以违背协议约定（规范），设置头部某个控制字段，实施针对协议执行的攻击。例如，协议中规定某个字段不能全部为 0，而攻击者却将其全部设置为 0，从而产生一个无效头部。针对无效头部，不同的协议或操作系统，处理方式和结果不尽相同，有些协议或系统会将其作为出错信息而丢弃，而有些协议或系统会做进一步分析处理。不管采取哪一种处理方式，都会占用系统资源。对于一个协议或系统，如果大量的资源用于处理这些包含无效头部数据，轻则效率下降，重则导致资源耗尽，形成典型的拒绝服务攻击效果。

一个典型的针对头部漏洞的攻击方式为“碎片攻击”。针对 IP 协议头部的攻击行为很多，但安全问题最多的是 IP 报文头部的长度、标识、偏移量等字段。其他许多字段如果无效会造成数据包被拒绝。“碎片攻击”就是借助于 IP 报文头部的“总长度”和“偏移量”这两个字段来进行。IP 报文头部的“总长度”用于告诉接收端该报文的总长度，该字段占用 16b，所以一个 IP 报文的总长度最大应为 $2^{16}=65536$B（64KB）；而“偏移量”用于指出在接收端的缓冲区进行分片重组时该分片的具体位置，该字段占用 13b，在缓冲区中进行重组时需要乘以 8（因为在发送端填写该字段时将该分片的第 1 个字节的编号除以 8）。如果攻击者构造一个“偏移量”为 8191（分片的第 1 个字节编号应为 $8191\times8=65528$，这是偏移量的最大值）、数据包的长度大于 8B（本例假设为 9B）、“标识”字段的第 3 个位为 1（表示是最后一个分片）的攻击包，在接收端的缓冲区中等待重组时，因为操作系统分配给该 IP 报文的缓冲区值最大为 64KB，而实际已超出了该限制值（见图 5-6），从而出现操作系统瘫痪等现象。

在图 5-6 所示的攻击报文中，正好最后一个字节无法放入操作系统的缓冲区中。同时，攻击者还可以构造一个 IP 报文的多个分片，让分片之间存在重叠，这样在接收端的缓存区中同样无法正常对 IP 报文进行重组，达到攻击效果。

典型的“死亡之 Ping”（Death of Ping）攻击也是基于“碎片攻击”的原理。Internet 控制报文协议（Internet Control Message Protocol，ICMP）报文通过 IP 报文进行传输，由于 IP 报文的最大长度为 65536B，因此早期路由器也限定 ICMP 报文的最大长度为 64KB（65536B），并在读取 ICMP 头部后，根据其中的“类型”和“代码”字段判断为哪一种 ICMP（如主机不可

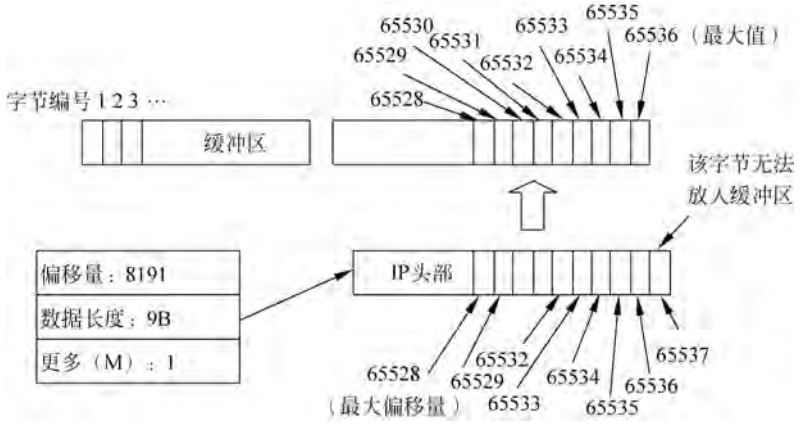

图 5-6　“碎片攻击”缓冲区重组 IP 报文时的情形

到达、网络不可到达等)报文,并分配相应的内存作为缓冲区。当攻击者构造一个不符合协议规范的 ICMP 报文(如 ICMP 报文总长度超过 64KB)时,就会使探测对象出现内存分配错误,导致协议崩溃。

基于头部的攻击利用了头部字段在设计及软件实现上存在的缺陷或不严格的约束等特点,攻击易于实施,而且很难发现和防范。

5.2.2　针对协议实现的安全

针对协议实现的安全是利用协议规范及协议实现过程中存在的漏洞所产生的安全问题。协议漏洞指所有的数据包都是符合协议的规范的、有效的,但它们与协议的执行过程之间存在冲突。一个协议就是为了实现某个功能而设计的按照一定顺序交换的一串数据包,它涉及如何建立连接、如何互相识别等环节。只有遵守这个约定,计算机之间才能相互通信交流。协议的 3 个要素包括语法、语义和时序。为了使协议的局部改变不会影响到整个协议的操作,协议的实现往往分成几个层次进行定义,各层在实际细节上具有相对的独立性。

通过对协议概念的理解,不难发现:协议是一个整体的概念,它的每一个实现细节都可以因为考虑不周而存在漏洞,而且某一层的实现由于对其他层是透明的,所以也可能隐藏着一些不安全因素。针对协议实现过程中存在漏洞的攻击主要包括以下几个方面。

(1) 不按序发送数据包。因为协议的实现是有序的,通信双方数据包的收发应该严格按照协议约定来执行。但是,如果一方不按照协议约定的顺序发送数据包,就会引起协议执行的错误。例如,TCP 协议建立连接时需要进行连接请求、请求应答和连接建立 3 个过程,它是一个封闭的环节,如果缺少一个环节整个协议将无法完整的执行。假设在发起通信一方发送了连接请求(第一次握手)后,通信的另一方返回了请求应答(第二次握手),但发起通信的一方迟迟不给予连接建立(第三次握手),而是频繁地发送连接请求,将会使另一方长期处于等待(等待第三次握手)状态而使资源耗尽。无序数据包的另一个例子是向指定对象连续发送不必要的数据包。例如,在某个连接已经打开的情况下,不断发送打开该连接的请求数据包,很显然这些数据包是多余的,但会消耗资源。

(2) 数据包到达太快或太慢。协议在执行过程中一般会进行一系列的交互,如请求、应答、确认等。其中,任何一个环节都应在约定的时间范围内执行结束,如果大于要求的最大时间,就会产生数据包到达太慢的现象,否则就会出现数据包到达太快的现象。数据包到达太慢的攻击是最常见的,如在双方共享资源的过程中,如果一方太慢,将会使对方长时间处于等待状态;如果太快,也会影响对方后续操作的正常执行,容易产生拒绝服务攻击。

(3) 数据包丢失。数据包丢失的产生原因很多,如网络线路的质量问题、协议中超时计时器的设置等。在不同协议的实现过程中,对丢包的处理不尽相同。如果丢包后要求对方重传,就需要对双方的缓存区设计提出严格要求,可能存在缓存区溢出攻击。

针对协议漏洞的网络攻击现象非常普遍,因为协议是网络通信的基础,而协议是设计者(人)为特定的功能实现制定的一系列规范。其间,至少有两个因素会导致漏洞的产生:人和功能描述。人是协议设计的主导者,其设计理念、技术路线、实现方法等都会因个人认知能力等因素而存在差异,常说的"山外有山,人外有人"就是这个道理;另一个是设计功能的描述,协议规范和实现过程中都会出现协议制定和实现上的漏洞,在某些情况下,规范本身的设计就存在缺陷,设计中存在的漏洞往往被利用,成为安全隐患。

目前广泛使用的无线局域网(Wireless Local Area Network,WLAN)的实现要比有线网络(以太网)复杂,越是复杂的协议越容易存在安全隐患,所以针对 WLAN 协议的攻击事件要比针对以太网协议的攻击事件多。例如,在 WLAN 中为了帮助无线移动终端(如智能手机、笔记本电脑等)找到无线访问节点(Access Point,AP)接入网络,可以在 AP 上设置用于标识当前网络接入服务的服务集标识(Service Set Identifier,SSID),并将其广播出去。这样,当移动终端探测到 SSID 后,就可以选择其中的一个(因为同一台移动终端可能同时会探测到多个 SSID,如图 5-7 所示)为其提供接入服务。很显然,无线 AP 中使用 SSID 的目的是便于用户选择无线接入服务,但攻击者利用该协议可以实施多种攻击行为。例如,攻击者通过提供虚假的 AP,当用户接入后收集用户的上网信息;再如,通过分析和破解 AP 的登录账号信息(用户名和密码)假冒合法用户等。

图 5-7 同一个移动终端探测到的多个 SSID

5.2.3 针对验证的安全

验证是一个用户对另一个用户进行身份识别的过程,如在访问受限系统时要求输入用户名和密码等。在网络安全中,验证是指一个实体对另一个实体的识别,并执行该实体的功能。例如,本章后续介绍的 ARP 欺骗、DHCP 欺骗、DNS 欺骗等,都是针对某个实体验证的攻击现象。

验证的实现方法多种多样,最常见的是某个用户对另一个用户证明自己的身份,即用户到用户的验证;另外,当一个用户在访问某个受限资源时,需要向某个应用程序、主机或协

议层证明自己的身份，即用户到主机的验证。其间，验证者都会向被验证者提交能够证明自己身份合法性的信息，常用的有用户名/密码、数字证书等。但不管采取哪一种验证方法，在实现过程中都存在安全隐患。

通信子网涉及 TCP/IP 体系网络层及以下各层，通信子网中节点之间的验证属于主机与主机之间的验证，通常需要借助主机的 IP 地址或 MAC 地址来实现。但是，由于 IP 地址和 MAC 地址都是可以伪造的，存在 IP 地址欺骗攻击和 MAC 地址欺骗攻击等安全威胁，所以仅从验证方式来看，基于地址的验证是不可靠的。对于任何一个 IP 分组来说，其包含的 IP 地址（源 IP 地址和目标 IP 地址）是在发送端主机上添加的，在到达目标主机之前任何节点都不允许修改（除网络地址转换（Network Address Translation，NAT）外），但是攻击者可以冒充为合法的数据发送者伪造一个 IP 分组，也可以在截获一个 IP 分组后修改其中的 IP 地址后重新发送。

图 5-8 所示的是一个典型的 IP 欺骗攻击方式，其中攻击者向计算机 A 发送一个返回地址为计算机 B 的数据包，即该数据包的源 IP 地址为计算机 B 的 IP 地址。根据协议约定，计算机 A 在接收到该数据包后，会根据数据包中的源 IP 地址向计算机 B 返回一个数据包。这就产生了一个针对 IP 地址欺骗的攻击。

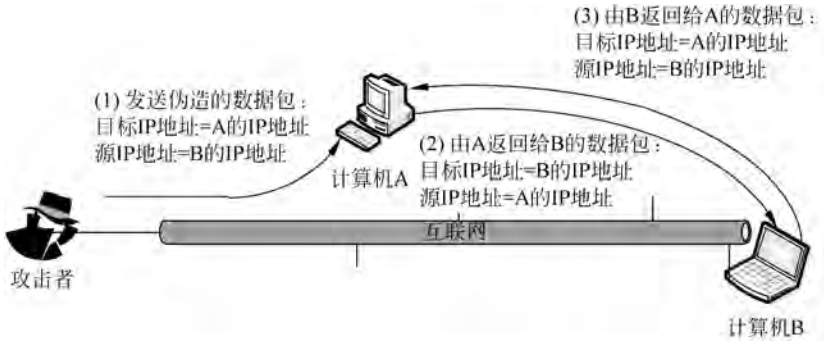

图 5-8　基于 IP 地址欺骗的攻击

经常遇到的一类基于 IP 欺骗的攻击是攻击者使用一个欺骗性 IP 地址向网络中发送一个 ICMP 应答请求报文。这将引起目标计算机向欺骗性 IP 地址（受害者）返回一个 ICMP 应答报文。当只有一个 ICMP 欺骗发生时并不会对网络安全产生影响，但攻击者可以用多种方法放大这种攻击行为。例如，当存在一个或多个攻击者向受害者产生大量的 ICMP 应答报文时，受害者将遭受 ICMP 应答报文的拒绝服务攻击；再如，可以向一个网络直接广播 ICMP 报文，连接该网络的路由器接收到该广播 ICMP 报文后，在没有进行对进入网络的 ICMP 报文限制的情况下，路由器将向该网络中的所有主机广播该 ICMP 报文。然后，该网络中的所有主机要对该 ICMP 请求报文进行应答，大量的 ICMP 应答报文将对路由器产生拒绝服务攻击。

5.2.4　针对流量的安全

针对流量漏洞的利用是流量安全的主要表现形式，是指通过对网络流量的嗅探和分析，窃取有价值信息的一种行为。在互联网中，所有的信息都以比特形式在网络中传输或存储，

一旦获取完整信息的流量后对其进行分析，就有可能获得流量中包含的真实数据。

针对流量漏洞的利用的另一个表现是大量的数据被发送到节点后，由该节点的某层或多层处理，但节点提供的资源（如缓存区大小、CPU 处理能力等）是有限的，无法满足超出一定数量的处理要求，所以引起数据包的丢失或节点崩溃等现象。

与针对流量的嗅探窃取信息相比较，利用流量使网络处理节点瘫痪所带来的危害性更大，而且防范方法很难。例如，通过数据加密技术可以防止数据的窃取，但单纯地通过增加节点的数据存储空间和提高处理能力应对流量攻击在互联网中是不现实的。

利用广播协议的特点，通过发送一个单一的请求报文就可以产生大量的应答报文，从而产生泛洪（雪崩）流量。图 5-9 所示为基于流量的漏洞，由单一请求报文产生大量应答报文实现攻击行为的例子。其中，攻击者发送一个广播报文到远程网络，该广播报文要求接收者必须进行应答。在图 5-9 所示的网络中，攻击者发送一个经过特殊设计的报文到目标网络，当连接该网络的路由器将该广播报文广播到每一个用户端设备时，每一个接收到该广播报文的用户端设备都会向转发该广播报文的路由器返回一个应答报文。如果目标网络中有大量的正在运行的用户端设备，路由器将收到大量的应答报文，当应答报文的流量超出路由器的处理能力时，目标网络将瘫痪。

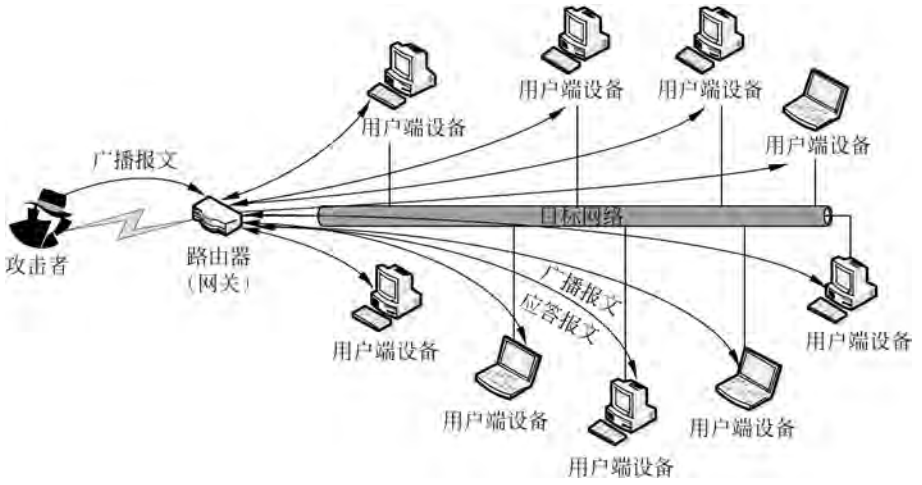

图 5-9　由广播报文产生的泛洪攻击


视频讲解

5.3　ARP 安全

地址解析协议（ARP）用来将 IP 地址映射到 MAC 地址，以便设备能够在共享介质的网络（如以太网）中通信。

5.3.1　ARP 概述

下面的例子很好地说明 ARP 是如何工作的。老师要将一封信交给教室里的某个学生，但是她并不认识这个学生，她只知道这个学生的姓名（IP 地址），于是她对教室里所有的人说："谁是王××，有你的信！"（ARP 请求），当王××听到这个消息时（地址匹配），他站

起来回答，然后老师就知道了他坐在几排几座（MAC 地址），最后把信送到他座位上。

在 ARP 协议的实现中还有如下一些注意事项。

（1）每台计算机上都有一个 ARP 缓存，它保存了一定数量的从 IP 地址到 MAC 地址的映射，同时当一个 ARP 广播到来时，虽然这个 ARP 广播可能与它无关，但 ARP 协议软件也会将其中的物理地址与 IP 地址的映射记录下来，这样做的好处是能够减少 ARP 报文在局域网上发送的次数。

（2）按照默认设置，ARP 高速缓存中的项目是动态的，ARP 缓存中 IP 地址与物理地址之间的映射并不是一旦生成就永久有效的，每一个 ARP 映射表项都有自己的寿命，如果在一段时间内没有使用，那么这个 ARP 映射就会从缓存中被删除，这一点和交换机 MAC 地址表的原理一样。这种老化机制大大减少了 ARP 缓存表的长度，加快了查询速度。

在以太网中，当主机要确定某个 IP 地址的 MAC 地址时，它会先检查自己的 ARP 缓存表，如果目标地址不包含在该缓存表中，主机就会发送一个 ARP 请求（广播形式），网段上的任何主机都可以接收到该广播，但是只有目标主机才会响应此 ARP 请求。由于目标主机在收到 ARP 请求时可以学习到发送方的 IP 地址到 MAC 地址的映射，因此它采用一个单播消息来回应请求。这个过程如图 5-10 所示。

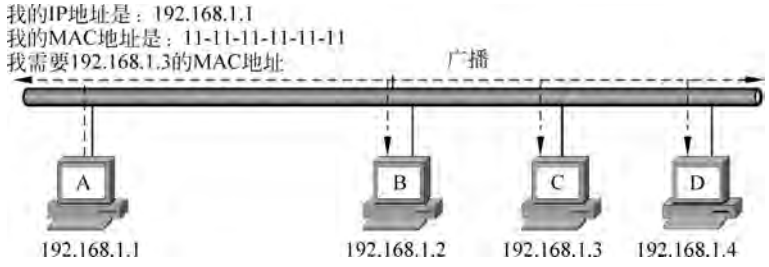

图 5-10　ARP 请求的过程

在图 5-10 中，主机 A 以广播形式发送 ARP 请求查询 IP 地址为 192.168.1.3 的主机的 MAC 地址，网段上所有的主机都会收到该 ARP 请求。

如图 5-11 所示，主机 B、主机 D 收到主机 A 发来的 ARP 请求时，它们发现这个请求不是发给自己的，因此它们忽略这个请求，但是它们还是将主机 A 的 IP 地址到 MAC 地址的映射记录到自己的 ARP 缓存表中。当主机 C 收到主机 A 发来的 ARP 请求时，它发现这个 ARP 请求是发给自己的，于是它用单播消息回应 ARP 请求，同时记录下其 IP 地址到 MAC 地址的映射。

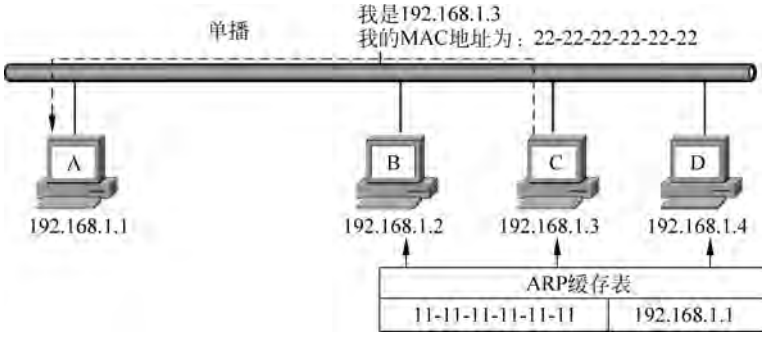

图 5-11　ARP 回应的过程

反向 ARP(Reverse Address Resolution Protocol,RARP)是 ARP 的逆过程,即通过 MAC 地址找到对应的 IP 地址。

5.3.2 ARP 欺骗

通过对 5.3.1 节内容的学习,读者会发现 ARP 本来是局域网中计算机之间通信时所采用的一种非常有效的协议。但是,由于一台主机在向另一台主机发送 ARP 响应时,并不一定首先要得到另一台主机的 ARP 请求,局域网中的任何一台主机都可以向其他主机发送公告:我的 IP 地址是××,我的 MAC 地址是××。这种协议设计上的漏洞便为网络攻击提供了可乘之机。

1. ARP 欺骗的概念和现状

由于 ARP 协议在设计中存在主动发送 ARP 报文的漏洞,使得主机可以发送虚假的 ARP 请求报文或响应报文,报文中的源 IP 地址和源 MAC 地址均可以进行伪造。在局域网中,可以伪造成某一台主机(如服务器)的 IP 地址和 MAC 地址的组合,也可以伪造成网关的 IP 地址和 MAC 地址的组合,等等。这种组合可以根据攻击者的意图进行任意搭配,而现有的局域网中却没有相应的机制和协议防止这种伪造行为。近几年来,局域网中的 ARP 欺骗已经泛滥成灾,几乎没有一个局域网未遭受过 ARP 欺骗的侵害。

由于 ARP 协议工作在 TCP/IP 参考模型的网际层与网络接口层之间(对应 OSI 参考模型的网络层与数据链路层之间),所以现有的网管软件和防病毒软件几乎对 ARP 欺骗无能为力,网络管理员只能通过地址绑定等最简单和原始的方法来防御 ARP 欺骗,而缺乏一种行之有效的全网解决方案。

从大量的 ARP 欺骗行为来看,虽然一部分是为了窃取他人计算机上发送的报文信息,但占的比例并不大。目前,绝大部分 ARP 欺骗是为了扰乱局域网中合法主机保存的 ARP 缓存表,使得网络中的合法主机无法正常通信,如表示为计算机无法上网或上网时断时续等。ARP 欺骗中的主机是指主要以 MAC 地址作为通信地址的设备,如局域网中的计算机、交换机等。所以,下面将分别针对计算机和交换机介绍 ARP 欺骗的现象。

2. 针对计算机的 ARP 欺骗

要全面理解 ARP 欺骗,首先要掌握如图 5-3 所示的 TCP/IP 体系结构的工作特点,即明确 ARP 协议在 TCP/IP 参考模型中的位置——网际层与网络接口层之间。目前,局域网中的 ARP 欺骗形式多种多样,下面仅以最常见的一种 ARP 欺骗现象为例进行介绍。

如图 5-12 所示,假设主机 A 向主机 B 发送数据。在主机 A 中,当应用程序要发送的数据到了 TCP/IP 参考模型的网际层与网络接口层之间时,主机 A 在 ARP 缓存表中查找是否有主机 B 的 MAC 地址(其实是主机 B 的 IP 地址与 MAC 地址的对应关系),如果有,则直接将该 MAC 地址(22-22-22-22-22-22)作为目的 MAC 地址添加到数据单元的网络首部(位于网络接口层),成为数据帧。在局域网(同一 IP 网段,如本例的 192.168.1.x)中,主机利用 MAC 地址作为寻址的依据,所以主机 A 根据主机 B 的 MAC 地址,将数据帧发送给主机 B。

如果主机 A 在 ARP 缓存表中没有找到目标主机 B 的 IP 地址对应的 MAC 地址,主机 A

主机	IP地址	MAC地址
主机A	192.168.1.1	11-11-11-11-11-11
主机B	192.168.1.2	22-22-22-22-22-22
主机C	192.168.1.3	33-33-33-33-33-33
主机D	192.168.1.4	44-44-44-44-44-44
主机E	192.168.1.5	55-55-55-55-55-55

图 5-12　主机中 IP 地址与 MAC 地址的对应关系

就会在网络上发送一个广播帧，该广播帧的目的 MAC 地址是“FF. FF. FF. FF. FF. FF”，表示向局域网内的所有主机发出这样的询问：IP 地址为 192. 168. 1. 2 的 MAC 地址是什么？在局域网中所有的主机都会接收到该广播帧，但在正常情况下因为只有主机 B 的 IP 地址是 192. 168. 1. 2，所以主机 B 会对该广播帧进行 ARP 响应，即向主机 A 发送一个 ARP 响应帧：我(IP 地址是 192. 168. 1. 2)的 MAC 地址是 22-22-22-22-22-22。

这样，主机 A 就知道了主机 B 的 MAC 地址，它就可以向主机 B 发送数据了。同时主机 A 还会更新自己的 ARP 缓存表，将主机 B 的 IP 地址与 MAC 地址的对应关系保存在自己的 ARP 缓存表中，以供下次通信时直接使用，避免进行广播查询。

但是，主机的 ARP 缓存表中并不会保存所有参与过通信的主机的 IP 地址和 MAC 地址的对应关系，而是采用了老化机制防止 ARP 缓存表过于庞大，因为过于庞大的 ARP 缓存表会影响主机的通信效率。在一段时间内，如果 ARP 缓存表中的某一条记录没有使用，就会被删除。

从上面的例子可以看出，ARP 协议的基础就是信任局域网内所有的主机，这样就很容易实现在局域网内的 ARP 欺骗。假设现在主机 D 要对主机 A 进行 ARP 欺骗，冒充自己是主机 C。具体实施中，当主机 A 要与主机 C 进行通信时，主机 D 主动告诉主机 A 自己的 IP 地址和 MAC 地址的组合是“192. 168. 1. 3＋44-44-44-44-44-44”，这样当主机 A 要发送给主机 C 数据时，会将主机 D 的 MAC 地址 44-44-44-44-44-44 添加到数据帧的目的 MAC 地址中，从而将本来要发给主机 C 的数据发给了主机 D，实现了 ARP 欺骗。在整个 ARP 欺骗过程中，主机 D 称为“中间人”(Man in the Middle)，对于这一中间人的存在，主机 A 根本没有意识到。

通过以上的 ARP 欺骗，主机 A 与主机 C 之间断开了联系。如图 5-13 所示，现在假设主机 C 是局域网中的网关，而主机 D 为 ARP 欺骗者。这样，当局域网中的计算机要与其他网络进行通信(如访问 Internet)时，所有发往其他网络的数据全部发给了主机 D，而主机 D 并非真正的网关，这样整个网络将无法与其他网络进行通信。这种现象在 ARP 欺骗中非常普遍。

3. 针对交换机的 ARP 欺骗

交换机的工作原理是通过主动学习下连设备的 MAC 地址，建立和维护端口和 MAC 地址的对应表，即交换机中的 MAC 地址表。通过 MAC 地址表，实现下连设备之间的通信。交换机中的 MAC 地址表也称为内容可寻址存储器(Content Addressable Memory，CAM)，如图 5-14 所示。

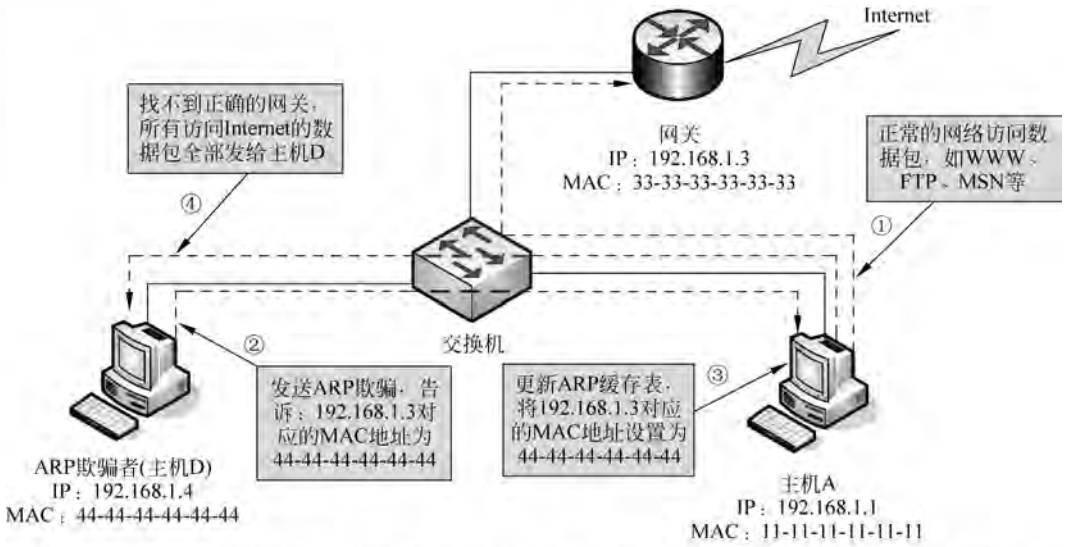

图 5-13　ARP 欺骗的实现过程

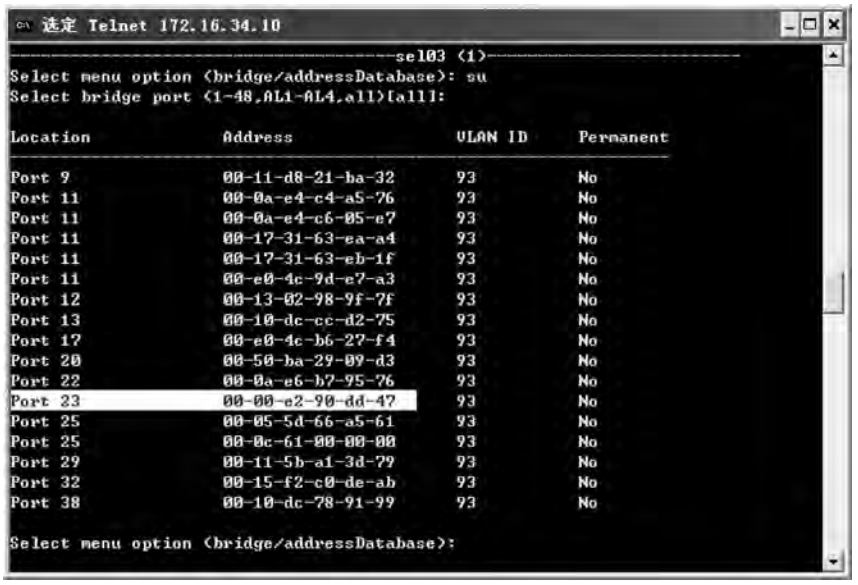

图 5-14　交换机中的 MAC 地址表

交换机中的 CAM 表详细地记录了参与通信的下连设备的 MAC 与交换机端口的一一对应关系。CAM 表在交换机加电启动时是空的，当下连的某一台主机要通信时，交换机会自动将该主机的 MAC 地址与下连端口的对应关系记录下来，在 CAM 表中形成一条记录，将这一过程称为交换机的学习。CAM 表的大小是固定的，不同交换机的 CAM 表大小可能不同。

在进行 ARP 欺骗时，ARP 欺骗者利用工具产生欺骗 MAC，并快速填满 CAM 表。交换机的 CAM 表被填满后，交换机便以广播方式处理通过交换机的数据帧，这时 ARP 欺骗

者可以利用各种嗅探攻击获取网络信息。CAM 表被填满后，流量便以洪泛(Flood)方式发送到所端口，其中交换机上连端口(Trunk 端口)上的流量也会发送给所有端口和邻接交换机。这时的交换机其实已成为一台集线器。与集线器不同，由于交换机上有 CPU 和内存，大量的 ARP 欺骗流量会对交换机产生流量过载，其结果是下连主机的网络速度变慢，并造成数据包丢失，甚至产生网络瘫痪。

在计算机网络中曾经出现的 SQL 蠕虫病毒就是利用组播功能，构造虚假的目的 MAC 地址将交换机的 CAM 表填满，对网络安全运行造成了非常大的威胁。

5.3.3 ARP 欺骗的防范

由于 ARP 欺骗方式多种多样，所以对 ARP 欺骗的防范方法也不尽相同。下面，针对前面介绍的针对计算机的 ARP 欺骗和针对交换机的 ARP 欺骗，分别介绍与之对应的防范方法。

1. 针对计算机的 ARP 欺骗的防范

ARP 缓存表中的记录可以是动态的(基于前面介绍的 ARP 响应)，也可以是静态的。如果 ARP 缓存表中的记录是动态的，即为了减少 ARP 缓存表的长度，加快查询速度，ARP 缓存表采用了老化机制，在一段时间内如果表中的某一条记录没有使用就会被删除。其中，Windows 操作系统的老化时间默认为 2min，而网络设备(如路由器、交换机等)默认为 5min。

静态 ARP 缓存表中的记录是永久性的，用户可以使用 TCP/IP 工具创建和修改，如 Windows 操作系统自带的 ARP 工具。下面，我们要用类似于图 5-13 所示的网络环境，以 Windows 操作系统为例，通过在用户计算机上绑定网卡的 IP 地址和 MAC 地址的方法防范出现网关地址的 ARP 欺骗。具体操作如下。

(1) 进入“命令提示符”窗口，在确保网络连接正常的情况下，使用 Ping 命令 Ping 网关的 IP 地址，如“ping 172.16.2.1”。

(2) 在保证 Ping 网关 IP 地址正常的情况下，输入“arp -a”命令，可以获得网关 IP 地址对应的 MAC 地址，如图 5-15 所示。

图 5-15 使用“arp -a”命令显示网关 IP 地址对应的 MAC 地址

读者会发现，这时该计算机上网关对应的 ARP 记录类型(Type)是动态(Dynamic)的。

(3) 利用“arp -s 网关 IP 地址 网关 MAC 地址”命令将本机中 ARP 缓存表中网关的记录类型设置为静态(Static)，如图 5-16 所示。

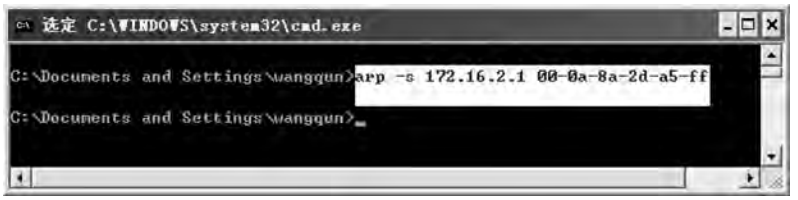

图 5-16 将 ARP 缓存表中的网关记录设置为静态类型

(4) 如果再次输入“arp -a”命令,就会发现 ARP 缓存表中网关的记录已被设置为静态类型。

以上操作仅适用于实验环境,因为利用以上手工设置方式修改 ARP 缓存表中的记录,会在计算机重新启动后失效,需要再次绑定。这显然在实际的网络环境中是不适用的。为解决这一问题,针对以上操作,可以编写一个批处理文件(如 arp.bat),然后将该批处理文件添加到 Windows 操作系统的“启动”栏中,这样每次开机后系统便会进行自动绑定。批处理文件的内容如下。

```
@echo off
arp -d
arp -s 172.16.2.1 00-0a-8a-2d-a5-ff
```

以上方法是针对网关进行设置的。如果用户的计算机需要经常与另一台计算机进行可靠的通信,则可以将对方计算机的 ARP 记录以静态方式添加到本机的 ARP 缓存表中。

2. 针对交换机的 ARP 欺骗的防范

在交换机上防范 ARP 欺骗的方法与在计算机上防范 ARP 欺骗的方法基本相同,还是使用将下连设备的 MAC 地址与交换机端口进行一一绑定的方法实现。在不同交换机上实现地址绑定的操作方法可能不同,思科系列交换机上实现地址绑定的操作方法可参考本书 4.3.1 节的内容。

目前,主流的交换机(如 Cisco、H3C、3COM 等)都提供了端口安全功能(Port Security Feature)。通过使用端口安全功能,可以进行如下控制。

(1) 指定端口上最大可以通过的 MAC 地址数量。

(2) 端口上只能使用指定的 MAC 地址。

对于不符合以上规定的 MAC 地址,进行相应的违背规则的处理,一般有以下 3 种方式(交换机类型和型号不同,具体方式可能有所不同)。

(1) Shutdown。即关闭端口,虽然这种方式是最有效的一种保护方式,但会给管理员带来许多不便,因为被关闭的端口一般需要通过手工方式进行重启。

(2) Protect。直接丢弃非法流量,但不报警。

(3) Restrict。丢弃非法流量,但产生报警。

通过利用端口安全功能,可以防范交换机 MAC/CAM 攻击。下面,以思科系列交换机为例进行介绍,其他交换机的配置原理与此基本相同,具体的配置命令可参阅相关的技术文档。

在进行端口安全功能设置时,端口上的 MAC 地址既可以通过交换机的自动学习功能

获得，也可以通过手工方式进行 MAC 地址与端口的绑定。当通过自动学习功能获得 MAC 地址时，交换机重启后会主动学习下连端口设备的 MAC 地址，直到学习到的 MAC 地址数达到设置的数量。但是，当交换机关机或重启后又要进行重新学习。下面以交换机的端口 fastethernet0/1 为例，介绍端口安全的配置方法。

```
Switch #conf t  (进入配置模式)
Switch(config) #interface fastethernet0/1  (选择 fastethernet0/1 端口,进入该端口配置状态)
Switch(config-if) #switchport port-security maximum 2  (设置最大 MAC 地址数为 2)
Switch(config-if) # switchport port-security violation shutdown  (当违背安全规则后自动关闭端口)
Switch(config-if) #end  (退出配置模式)
Switch #wr  (保存设置)
```

目前较新的端口安全技术是 Sticky Port Security，它克服了 Port Security Feature 存在的交换机重启后 CAM 表中自动学习获得的 MAC 地址会丢失的不足，交换机可以将学到的 MAC 地址写入端口配置中，即使交换机重启或关机，配置仍然存在。

还需要说明的是，由于 ARP 欺骗的严重性，许多交换机设备制造商纷纷推出了具有防范 ARP 欺骗功能的交换机产品。这些产品的效果确实不错，但有一个前提是该网络中所有的交换机都必须使用同一个厂商的产品，而且对交换机的型号也有一定的要求，有些早期的交换机可能无法支持此功能。


视频讲解

5.4　DHCP 安全

动态主机配置协议(DHCP)是一个客户端/服务器协议，在 TCP/IP 网络中对客户端动态分配和管理 IP 地址等配置信息，以简化网络配置，方便用户使用及管理员的管理。

5.4.1　DHCP 概述

一台 DHCP 服务器可以是一台运行 Windows Server 2012/2016、UNIX 或 Linux 的计算机，也可以是一台路由器或交换机。DHCP 的工作过程如图 5-17 所示。

图 5-17　DHCP 的工作过程

具体操作过程如下。

(1) DHCP 客户端首次初始化时会向 DHCP 服务器发送一个请求(DHCPDISCOVER)，请求获得 IP 寻址信息，这个寻址信息包括 IP 地址、子网掩码、默认网关、DNS 服务器地址等，请求中同时也包含了客户端自己的 MAC 地址信息。DHCPDISCOVER 以广播形式发送，网段上的所有设备都会收到这个请求。

(2) 当 DHCP 服务器接收到请求时，它会从自己的地址池中选择一个 IP 地址分配给客户端，并且把其他 TCP/IP 配置一起发送过去(DHCPOFFER)。DHCPOFFER 以单播形式发送，因为它是针对某个具体主机的消息，DHCP 服务器可以从 DHCPDISCOVER 消息

中获得客户端的 MAC 地址。

(3) 当客户端接收到服务器所提供的信息时,它又以广播方式发送一个 DHCPREQUEST 消息,指明:我需要得到你的服务。

需要注意的是,为什么还要以广播形式发送 DHCPREQUEST 消息呢?如果一个网段上存在多个 DHCP 服务器,那么 DHCP 客户端可能收到多个 DHCP 服务器响应的 DHCPOFFER 消息,DHCP 客户端只会选择最先收到的那个 DHCPOFFER 消息。所以,以广播方式发送 DHCPREQUEST 消息有两个作用:一是通知那个服务器已经收到它所提供的 IP 地址,以及需要它的服务;二是通知网络上其他 DHCP 服务器,拒绝它们提供的 IP 寻址信息。

(4) DHCP 服务器接收到 DHCPREQUEST 消息后,它会将所提供的 IP 地址和其他设置交给数据库,并且向 DHCP 客户端以单播形式发送一个 DHCPACK 消息,确认 DHCP 过程已经完成。

经过以上几个步骤,这个 IP 地址就会租给这个客户端一段时间,在租用期间,客户端每次登录时都会向服务器发出这个 IP 地址的续定请求(DHCPREQUEST)。如果租用期到了,但是客户端没有续租,这个 IP 地址就会退回到 DHCP 服务器的地址池中等待重新分配。

5.4.2 DHCP 的安全问题

在通过 DHCP 提供客户端 IP 地址等信息分配的网络中,存在一个非常大的安全隐患:当一台运行有 DHCP 客户端程序的计算机连接到网络中时,即使是一个没有权限使用网络的非法用户也能很容易地从 DHCP 服务器获得一个 IP 地址及网关、DNS 等信息,成为网络的合法使用者。由于在 TCP/IP 网络中,很多权限是基于 IP 地址来设置的,与设备的 MAC 地址不同的是 IP 地址属于逻辑地址,IP 地址具有不确定性,所以如果要进行基于 IP 的认证或权限控制是没有意义的(此问题已在第 4 章进行了讨论)。

由于 DHCP 客户端在获得 DHCP 服务器的 IP 地址等信息时,系统没有提供对合法 DHCP 服务器的认证,所以 DHCP 客户端从首先得到 DHCP 响应(DHCPOFFER)的 DHCP 服务器处获得 IP 地址等信息。为此,不管是人为的网络攻击或破坏,还是无意的误操作,一旦在网络中接入了一台 DHCP 服务器,该 DHCP 服务器就可以为 DHCP 客户端提供 IP 地址等信息的服务。其结果是客户端从非法 DHCP 服务器获得了不正确的 IP 地址、网关、DNS 等参数,无法实现正常的网络连接;或客户端从非法 DHCP 服务器处获得的 IP 地址与网络中正常用户使用的 IP 地址冲突,影响了网络的正常运行。尤其当客户端获得的 IP 地址与网络中某些重要的服务器的 IP 地址冲突时,整个网络将处于混乱状态。

如图 5-18 所示,一台非法 DHCP 服务器接入网络中,并"冒充"为这个网段中的合法 DHCP 服务器。这时,如果有一台 DHCP 客户端接入网络,将向网络中广播一个 DHCPDISCOVER 的请求信息,由于非法 DHCP 服务器与 DHCP 客户端处于同一个网段,而正确的 DHCP 服务器位于其他网段,所以一般情况下非法 DHCP 服务器优先发送 DHCPOFFER 响应给 DHCP 客户端,而 DHCP 客户端并不采用后到的正确的 DHCP 服务器的 DHCPOFFER 响应。这样,DHCP 客户端将从非法 DHCP 服务器处获得不正确的 IP 地址、网关、DNS 等配置参数。

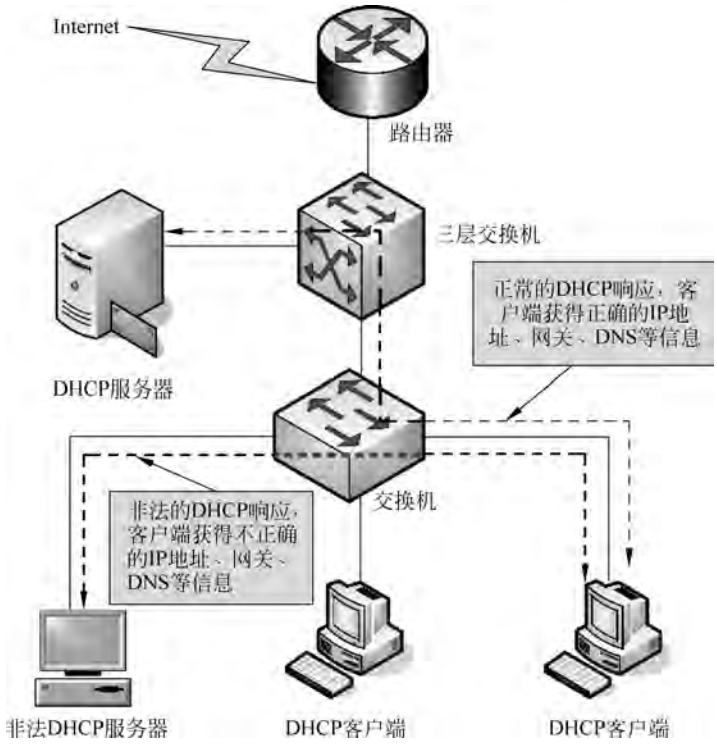

图 5-18　非法 DHCP 服务器的工作原理

5.4.3　非法 DHCP 服务的防范

非法 DHCP 服务存在大量的安全隐患，如果将非法 DHCP 服务器与一些攻击程序结合使用，则可以很方便地获得网络中用户的有用信息，如操作系统的用户账号和密码等。本节将结合应用实际，介绍几种防范非法 DHCP 服务的有效方法。

1. 使用 DHCP Snooping 信任端口

DHCP Snooping 能够过滤来自网络中非法 DHCP 服务器或其他设备的非信任 DHCP 响应报文。在交换机上，当某一端口设置为非信任端口时，可以限制客户端特定的 IP 地址、MAC 地址或 VLAN ID 等报文通过。为此，可以使用 DHCP Snooping 特性中的可信任端口防止用户私置 DHCP 服务器或 DHCP 代理。一旦将交换机的某一端口设置为指向正确 DHCP 服务器的接入端口，则交换机会自动丢失从其他端口上接收到的 DHCP 响应报文，如图 5-19 所示。

在配置时，可将交换机上与 DHCP 服务器连接的端口设置为 DHCP Snooping 的信任端口，其他端口默认情况下都为非信任端口。在如图 5-19 所示的网络中，DHCP 客户端发出 DHCPDISCOVER 请求报文，由于非信任端口并不限制该请求报文，所以非法 DHCP 服务器也会接收到该请求并发送 DHCPOFFER 响应报文。但是，非信任端口会阻断 DHCPOFFER 响应报文的通过，所以 DHCP 客户端只能接收到正确的 DHCP 服务器的响应，避免了非法 DHCP 服务器提供 IP 地址等信息给客户端。

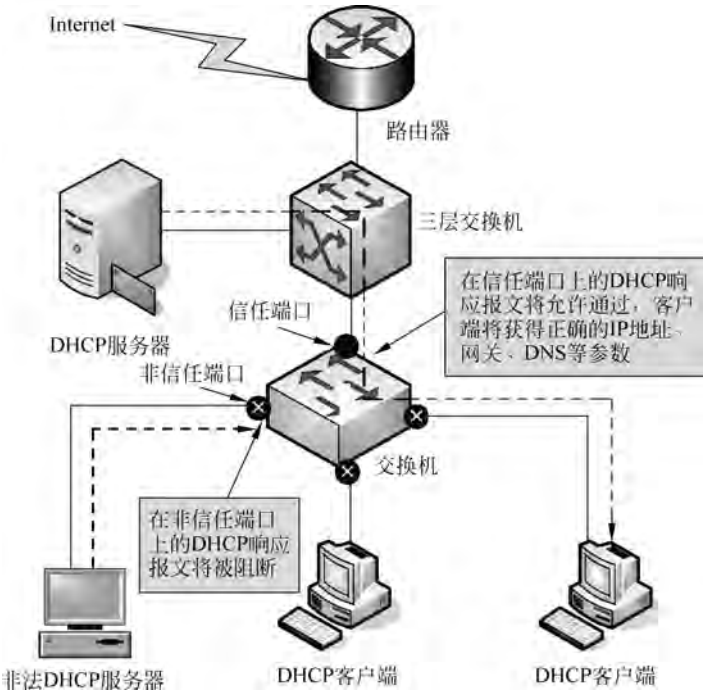

图 5-19　DHCP Snooping 的工作原理

下面介绍在思科交换机上 DHCP Snooping 特性的实现方法。假设要将交换机的第一个端口 fastethernet0/1 设置为信任端口，DHCP 服务器将连接在该端口上。具体配置方法如下。

```
Switch #conf t  (进入配置模式)
Switch(config) #interface fastethernet0/1  (选择 fastethernet0/1 端口,进入该端口配置状态)
Switch(config-if) # ip dhcp snooping trust  (将该端口设置为受信任端口)
Switch(config-if) # ip dhcp snooping limit rate 500  (设置每秒钟最多处理 500 个 DHCP 报文)
Switch(config-if) #end
Switch #wr
Switch #sh ip dhcp snooping  (显示交换机上 DHCP Snooping 的配置情况)
```

2. 在 DHCP 服务器上进行 IP 与 MAC 地址的绑定

在通过 DHCP 服务器进行客户端 IP 地址等参数分配的网络中，对于一些重要部门的用户，可以通过在 DHCP 服务器上绑定 IP 与 MAC 地址，实现对指定计算机 IP 地址的安全分配。下面以 Windows Server 2012 操作系统集成的 DHCP 服务为例，介绍实现方法。

(1) 确保 DHCP 服务器的运行正常。如果读者的计算机上还没有安装 DHCP 服务器，在以 Administrator 身份登录系统后，可通过选择“服务器管理器→仪表板→添加角色和功能”，在出现的“选择服务器角色”界面中选取“DHCP 服务器”进行安装。安装好 DHCP 组件后，还要通过“新建作用域”指定为客户端分配的 IP 地址段、网关、DNS 等信息。读者可参阅 Windows Server 2012 系统操作手册或帮助文档来完成该操作。

(2) 选择“开始→系统管理工具→DHCP”，打开“DHCP”窗口。

（3）选择“作用域→保留”，右击，在出现的快捷菜单中选择“新建保留”，打开如图 5-20 所示的对话框。在“保留名称”后面输入客户端用户的名称，在“IP 地址”后面输入该作用域中一个未分配的 IP 地址，在“MAC 地址”后面输入指定客户端计算机的 MAC 地址，在“支持的类型”下面选择“两者”或“仅 DHCP”。

图 5-20　进行 IP 地址与客户端 MAC 地址的绑定

（4）单击“添加”按钮，完成对该客户端 IP 地址与 MAC 地址的绑定操作。

采用相同的方法，完成对其他 DHCP 客户端 IP 地址与 MAC 地址的绑定操作，如图 5-21 所示。

图 5-21　显示已创建的客户端 IP 地址与 MAC 地址的绑定关系

加强对网络中 DHCP 服务器的安全管理，可以防止出现 DHCP 攻击或欺骗。针对网络中所使用的交换机等设备和 DHCP 服务器软件的不同，所采取的安全技术和策略也不尽相同。读者可通过参阅相关设备和软件的技术文档，加强对 DHCP 服务的管理。

5.5 TCP 安全


视频讲解

UDP 和 TCP 是 TCP/IP 参考模型传输层的两个通信协议。其中，UDP 是一种不可靠的、面向非连接的通信协议，而 TCP 是一种可靠的、面向连接的通信协议。将通过 UDP 协议传输的数据单位称为数据报，而将通过 TCP 协议传输的数据单位称为报文段。由于两种协议的功能不同，所以传输层 UDP 的首部格式非常简单，而 TCP 的首部格式非常复杂。与 UDP 不同的是，TCP 是为可靠的通信过程而开发的协议，但在实际应用中却出现了针对 TCP 协议漏洞的不安全因素，甚至是网络攻击。

5.5.1 TCP 概述

TCP 协议涉及 TCP 报文段的结构、TCP 连接的建立与终止、TCP 数据的传输、流量控制、差错控制、数据重传等内容。由于本章主要讨论的是协议的安全问题，所以在这里仅关注 TCP 面向连接的传输所需要的 3 个阶段：连接建立、数据传输和连接终止，对其工作过程进行介绍，并发现存在的安全问题。

1. 连接建立

TCP 是面向连接的。在面向连接的环境中，开始传输数据之前，在两个终端之间必须先建立一个连接。建立连接的过程可以确保通信双方在发送用户数据之前已经准备好了传送和接收数据。对于一个要建立的连接，通信双方必须用彼此的初始化序列号 SEQ 和来自对方成功传输确认的确认序号 ACK 进行同步（ACK 号指明希望收到的下一个字节的编号）。习惯上将同步信号写为 SYN，应答信号写为 ACK。整个同步的过程称为三次握手，图 5-22 说明了这个过程，具体如下。

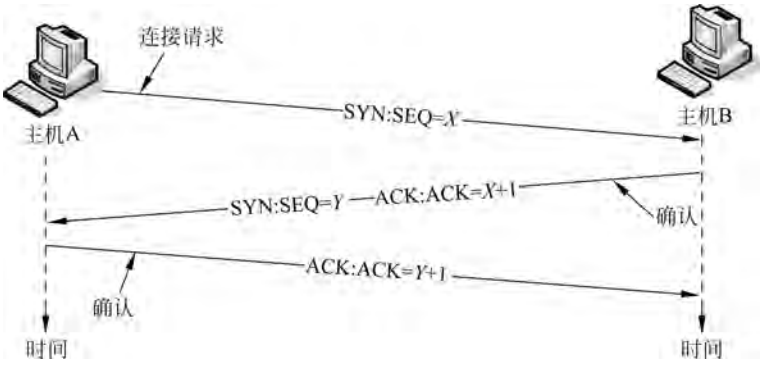

图 5-22 TCP 连接建立时的三次握手

（1）主机 A 发送 SYN 给主机 B：我的初始化序列号 SEQ 是 X。主机 A 通过向主机 B 发送 SYS 报文段，实现从主机 A 到主机 B 的序列号的同步，即确定 SEQ 中的 X。

（2）主机 B 发送 SYN、ACK 给主机 A：我的初始化序列号 SEQ 是 Y（如果主机 B 同意与主机 A 建立连接），确认序号 ACK 是 $X+1$（等待接收第 $X+1$ 号字节的数据流）。主机 B 向主机 A 发送 SYN 报文段的目的是实现从主机 B 到主机 A 的序列号的同步，即确定

SEQ 中的 Y。主机 B 向主机 A 发送确认信息 ACK＝X＋1，这是因为在 TCP 连接过程中，把正确接收到的最后一个序列号再加 1 的和，作为现在的确认号。

(3) 主机 A 发送 ACK 给主机 B：我的确认序号 ACK 是 Y＋1。

通过以上 3 个步骤(三次握手)，TCP 连接建立，开始传输数据。

2. 数据传输

在连接建立后，TCP 将以全双工方式传送数据，在同一时间主机 A 与主机 B 之间可以同时进行 TCP 报文段的传输，并对接收到的 TCP 报文段进行确认。如图 5-23 所示，当通过三次握手建立了主机 A 与主机 B 之间的 TCP 连接后，现在假设主机 A 要向主机 B 发送 1800B 的数据，主机 B 要向主机 A 发送 1500B 的数据。

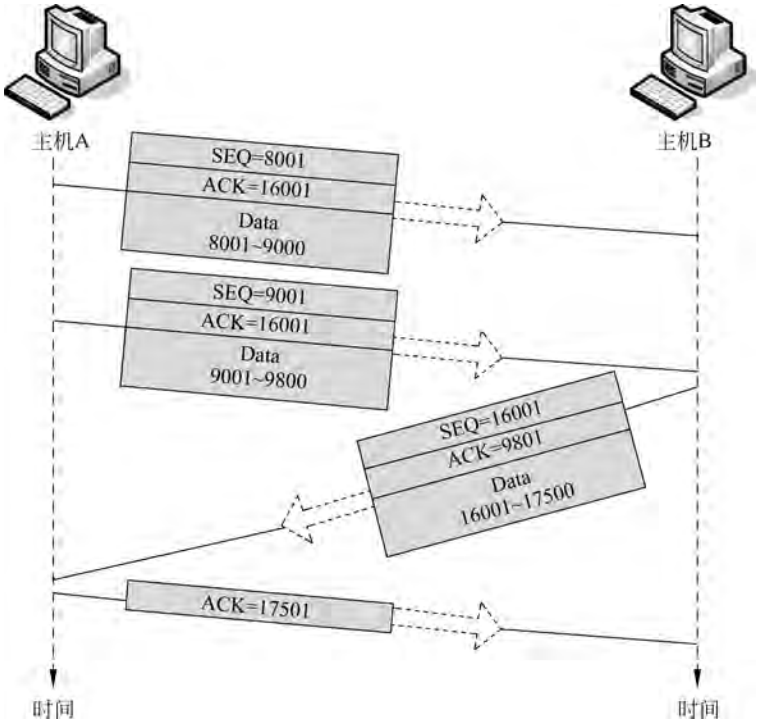

图 5-23　TCP 报文段的传输过程

在 TCP 报文段的首部信息中有一个序号字段和一个确认序号字段，在数据传输过程中要用到这两个字段的功能特性。TCP 把一个连接中发送的所有数据都按字节进行编号，而且在两个方向上的编号是互不影响的。当主机要发送数据时，TCP 从应用进程中接收数据，并将其存储在发送缓存中，然后按字节进行编号，这也是为什么将 TCP 报文称为字节流的原因。编号并不一定从 0 开始，而是在 0～(2^{32}－1)之间取一个随机数作为第一个字节的编号。例如，在本例中主机 A 正好取了 8001 作为第一个字节的编号，由于数据总长度为 1800，所以字节的编号从 8001～9800。同理，主机 B 的字节编号假设为 16001～17500。

当对字节进行编号后，TCP 就给每一个报文段分配一个序号，该序号即这个报文段中的第一个字节的编号。在本例中，主机 A 发送的数据被分成两个报文段(每 1000B 为一段)，由于第一个字节的编号为 8001，所以第一个报文段的序号 SEQ＝8001。第二个报文段

只有 800B，第二个报文段中的第一个字节的编号为 9001，所以第二个报文段的序号 SEQ＝9001。主机 B 正好以 1500B 为一个报文段，所以主机 B 发送给主机 A 的数据正好存放在一个报文段中，该报文段的序号 SEQ＝16001。

每一个报文段可以选择不同的路径在网络中进行传输，在接收端需要对接收到的报文段进行确认。前面已经提到，在 TCP 中确认序号被定义为下一个希望接收到的字节的编号，所以在本例中当主机 B 成功接收到主机 A 发送过来的第二个报文段时，由于该报文段中的字节编号为 9001～9800，所以主机 B 发送给主机 A 的确认序号 ACK＝9801。

另外，在本例中还有 3 个问题需要说明：一是确认信息由发送信息同时捎带，每一个报文段中的 ACK 序号就是对已成功接收到的报文段的确认；二是为了提高 TCP 的传输效率，主机并不会对接收到的每一个报文段报送确认信息，而是当同时接收到多个报文段后再发送确认信息，所以在本例中主机 B 只对主机 A 发送了一个确认信息；三是主机 A 在最后一次只发送了一个 ACK＝17501 的确认信息，表示已成功接收到主机 B 发送过来的报文段。这是因为主机 A 在本次 TCP 连接中已经没有数据进行发送。

3. 连接终止

对于一个已经建立的连接，TCP 使用改进的三次握手释放连接（使用一个带有 FIN 附加标记的报文段，即在 TCP 报文段首部中将 FIN 字段的值置为 1）。TCP 关闭连接的步骤如图 5-24 所示。

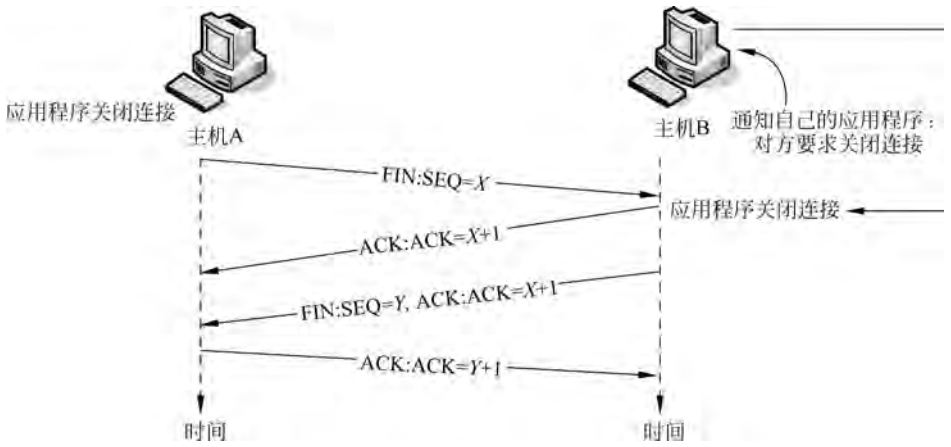

图 5-24　TCP 使用改进的三次握手释放连接

（1）当主机 A 的应用程序通知 TCP 数据已经发送完毕时，TCP 向主机 B 发送一个带有 FIN 附加标记的报文段（FIN 表示英文 Finish）。

（2）主机 B 收到这个 FIN 报文段之后，并不立即用 FIN 报文段回复主机 A，而是先向主机 A 发送一个确认序号 ACK，同时通知自己相应的应用程序对方要求关闭连接（先发送 ACK 的目的是防止在这段时间内，对方重传 FIN 报文段）。

（3）主机 B 的应用程序告诉 TCP：我要彻底关闭连接。TCP 向主机 A 送一个 FIN 报文段。

（4）主机 A 收到这个 FIN 报文段后，向主机 B 发送一个 ACK 报文段，表示彻底释放连接。

5.5.2　TCP 的安全问题

在 TCP/IP 网络中，如果两台主机之间要实现可靠的数据传输，首先要通过三次握手方式建立主机之间的 TCP 连接，但在 TCP 连接过程中很容易出现一个严重的安全问题——TCP SYN 泛洪攻击。

按照 TCP 连接建立时三次握手的协议约定，当源主机 A 要建立与目的主机 B 之间的 TCP 连接时，源主机 A 首先发送一个用于同步的 SYN 报文段（第一次握手）。当目的主机 B 接收到这个报文段时，在正常情况下目的主机会打开连接端口，并向源主机 A 返回一个 SYN＋ACK 的报文段（第二次握手）。同时，目的主机 B 将这个处于“半开放状态”的连接放在等待队列中，等待源主机 A 的 ACK 确认报文段（即等待第三次握手的实现）。这段等待时间一般为 75s～25min。

TCP SYN 泛洪攻击的工作过程如图 5-25 所示。如果在第一次握手过程中，源主机 A 发送给目的主机 B 的 SYN 报文段中的 IP 地址是伪造的，同时源主机 A 向目的主机 B 发送大量的 SYN 报文段，这时，目的主机 B 会正常接收这些 SYN 报文段，并发送 SYN＋ACK 确认报文段。由于目的主机 B 接收到的 SYN 报文段中的 IP 地址都是伪造的，所以发送出去的 SYN＋ACK 确认报文段全部得不到回复。在目的主机 B 的队列中存在大量的“半开放状态”的连接，最终将队列的存储空间填满，主机 B 因资源耗尽而瘫痪。

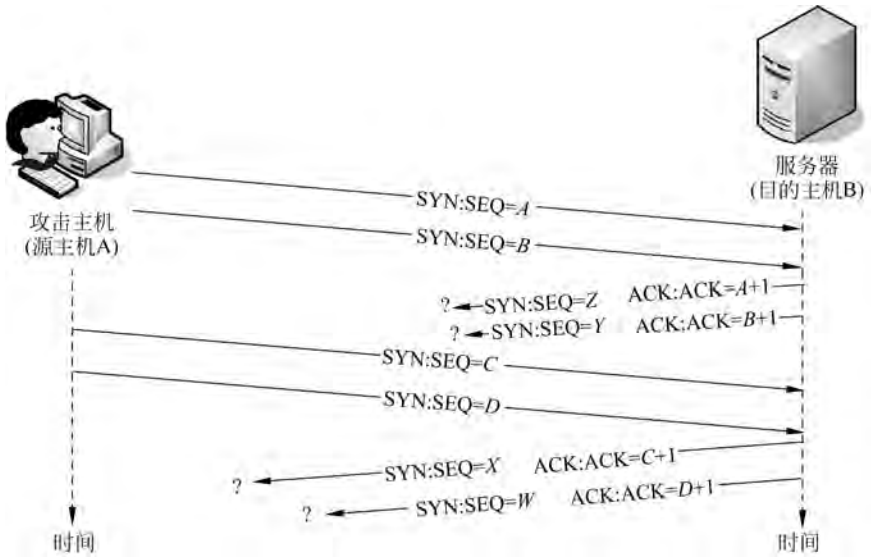

图 5-25　TCP SYN 泛洪攻击的工作过程

这种 TCP SYN 泛洪攻击属于一种典型的拒绝服务攻击（DOS 攻击），即攻击者使用大量的服务请求耗尽了服务器的资源，使服务器无法处理正常的服务请求，最终造成系统的瘫痪。

5.5.3 操作系统中 TCP SYN 泛洪的防范

可以在 Windows 注册表内配置 TCP/IP 参数，以保护服务器免遭网络级别的 TCP SYN 泛洪攻击。下面以 Windows Server 2012 为例进行介绍。

1. 启用 TCP SYN 攻击保护

启用 TCP SYN 泛洪攻击保护的命名值，位于注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services 下，如表 5-1 所示。

表 5-1 启用 TCP SYN 攻击保护

值 名 称	建议值	有效值	说 明
SynAttackProtect	2	0～2	使 TCP 调整 SYN＋ACK 的重传。配置此值后，在遇到 TCP SYN 攻击时，对连接超时的响应将更快速。在超过 TcpMaxHalfOpen 或 TcpMaxHalfOpenRetried 的值后，将触发 SYN 攻击保护

2. 设置 TCP SYN 保护阈值

下列值确定触发 SYN 保护的阈值。这一部分中的所有注册表项和值都位于注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services 下。这些注册表项和值如表 5-2 所示。

表 5-2 设置 TCP SYN 保护阈值

值 名 称	建议值	有效值	说 明
TcpMaxPortsExhausted	5	0～65535	指定触发 TCP SYN 泛洪攻击保护所必须超过的 TCP 连接请求数的阈值
TcpMaxHalfOpen	500	100～65535	在启用 SynAttackProtect 后，该值指定处于 SYN_RCVD 状态的 TCP 连接数的阈值。在超过 SynAttackProtect 后，将触发 TCP SYN 泛洪攻击保护
TcpMaxHalfOpenRetried	400	80～65535	在启用 SynAttackProtect 后，该值指定处于至少已发送一次重传的 SYN_RCVD 状态中的 TCP 连接数的阈值。在超过 SynAttackProtect 后，将触发 TCP SYN 泛洪攻击保护

3. 设置其他保护

这一部分中的所有注册表项和值都位于注册表项 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services 下。这些注册表项和值如表 5-3 所示。

表 5-3 设置其他保护

值 名 称	建议值	有效值	说 明
TcpMaxConnectResponseRetransmissions	2	0～255	控制在响应一次 SYN 请求之后、在取消重传尝试之前 SYN+ACK 的重传次数
TcpMaxDataRetransmissions	2	0～65535	指定在终止连接之前 TCP 重传一个数据段(不是连接请求段)的次数
EnablePMTUDiscovery	0	0,1	将该值设置为 1(默认值)可强制 TCP 查找在通向远程主机的路径上的最大传输单元或最大数据包大小。攻击者可能将数据包强制分段,这会使堆栈不堪重负。对于不是来自本地子网的主机的连接,将该值指定为 0 可将最大传输单元强制设为 576B
KeepAliveTime	300000	80～4294967295	指定 TCP 尝试通过发送持续存活的数据包来验证空闲连接是否仍然未被触动的频率
NoNameReleaseOnDemand	1	0,1	指定计算机在收到名称发布请求时是否发布其 NetBIOS 名称

使用表 5-4 中汇总的值可获得最大限度的保护。

表 5-4 建议值

值 名 称	值(REG_DWORD)
SynAttackProtect	2
TcpMaxPortsExhausted	1
TcpMaxHalfOpen	500
TcpMaxHalfOpenRetried	400
TcpMaxConnectResponseRetransmissions	2
TcpMaxDataRetransmissions	2
EnablePMTUDiscovery	0
KeepAliveTime	300000(5min)
NoNameReleaseOnDemand	1

5.6 DNS 安全


视频讲解

为了解决主机 IP 地址与主机名之间的对应关系,Internet 网络信息中心(Internet Network Information Center,InterNIC)制定了一套称为域名系统(DNS)的分层名字解析

方案，当 DNS 用户提出 IP 地址查询请求时，就可以由 DNS 服务器中的数据库提供所需的数据。DNS 技术目前已广泛地应用于 Internet 和 Intranet 中。

5.6.1 DNS 概述

DNS 是一组协议和服务，它允许用户在查找网络资源时使用层次化的对用户友好的名字取代 IP 地址。当 DNS 客户端向 DNS 服务器发出 IP 地址的查询请求时，DNS 服务器可以从其数据库内寻找所需要的 IP 地址给 DNS 客户端。这种由 DNS 服务器在其数据库中找出客户端 IP 地址的过程叫作“主机名称解析”。

1. DNS 的功能及组成

简单地讲，DNS 协议的最基本功能是为主机名与对应的 IP 地址之间建立映射关系。例如，新浪网站的一个 IP 地址是 202.106.184.200，几乎所有浏览该网站的用户都是使用 www.sina.com.cn，而并非使用 IP 地址访问。使用主机名（域名）具有以下两个优点。

（1）主机名便于记忆，如 sina.com.cn；

（2）数字形式的 IP 地址可能由于各种原因而改变，而主机名可以保持不变。

DNS 的工作任务是在计算机主机名与 IP 地址之间进行映射。DNS 位于 TCP 参考模型的最高层，使用 TCP 和 UDP 作为传输协议（一般多使用 UDP，因为 UDP 的系统开销较小）。DNS 模型相当简单，客户端向 DNS 服务器提出访问请求（如 www.sina.com.cn），DNS 服务器在收到客户端的请求后在数据库中查找相对的 IP 地址（202.106.184.200），并作出反应。如果该 DNS 服务器无法提供对应的 IP 地址（如数据库中没有该客户端主机名对应的 IP 地址）时，就转给下一个它认为更好的 DNS 服务器去处理。

当需要给某人打电话时，你可能知道这个人的姓名，而不知道他的电话号码。这时，可以通过查看电话号码簿查得他的电话号码，从而与他进行通话。由此可以看出，电话号码簿的功能便是建立姓名与电话号码之间的映射关系。而 DNS 的功能与这里的电话号码簿很类似。

DNS 是为 TCP/IP 网络提供的一套协议和服务，是由名字分布数据库组成的。它建立了名为域名空间的逻辑树结构，是负责分配、改写、查询域名的综合性服务系统。该空间中的每个节点或域都有一个唯一的名字。

组成 DNS 系统的核心是 DNS 服务器，它是回答域名服务查询的计算机，它允许为私人 TCP/IP 网络和连接公共 Internet 的用户提供并管理 DNS 服务，维护 DNS 名字数据并处理 DNS 客户端主机名的查询。DNS 服务器保存了包含主机名和相应 IP 地址的数据库。例如，如果提供了域名 www.sina.com，DNS 服务器将返回新浪网站的 IP 地址 202.106.184.200。

DNS 是一种看起来与磁盘文件系统的目录结构类似的命名方案，域名也通过使用“.”分隔每个分支，标识一个域在逻辑 DNS 层次中相对于其父域的位置。但是，当定位一个文件位置时，是从根目录到子目录再到文件名，如 C:\winnt\win.exe；而当定位一个主机名时，是从最终位置到父域再到根域，如 sina.com。

图 5-26 显示了顶级域的域名空间及下一级子域之间的树状结构关系，图中的每一个节点以及其下的所有节点叫作一个域。域可以有主机（计算机）和其他域（子域）。例如，在图 5-26 中，pc1.sd.cninfo.net 就是一个主机，而 sd.cninfo.net 则是一个子域。一般在子域

中含有多个主机，例如，ah. cninfo. net 子域下就含有 pc1. ah. cninfo. net 和 pc30. ah. cninfo. net 两台主机。

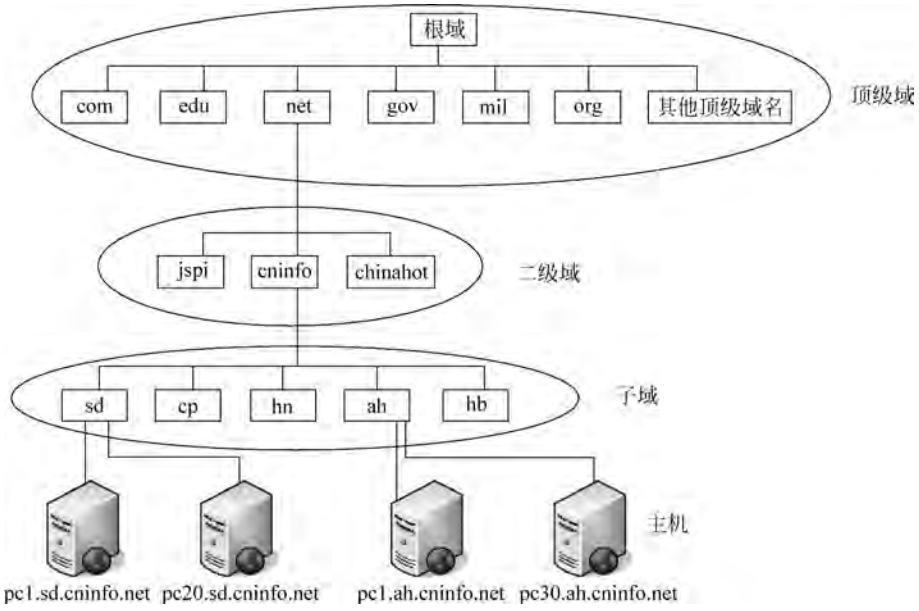

图 5-26　Internet 的域名结构

（1）根域。代表域名命名空间的根，这里为空。

（2）顶级域。直接处于根域下面的域，代表一种类型的组织和一些国家。在 Internet 中，由 InterNIC 进行管理和维护。例如，在顶级域名中，com 代表商业组织，edu 代表教育和学术机构等；在域名的国家代码中，cn 代表中国，us 代表美国等。

（3）二级域。在顶级域下面，用来标明顶级域以内的一个特定的组织。在 Internet 中，也是由 InterNIC 负责对二级域名进行管理和维护，以保证二级域名的唯一性。

（4）子域。在二级域下面所创建的域，它一般由各个组织根据自己的要求自行创建和维护。

（5）主机。域名空间中的最底层，它被称为完全合格的域名（Fully Qualified Domain Name，FQDN）。

2. DNS 的解析过程

现在假设客户端 Web 浏览器要访问网站 www. sina. com，整个访问过程如图 5-27 所示，具体描述如下。

（1）Web 浏览器调用 DNS 客户端程序（该程序称为“解析器”），先在本地的 DNS 缓存中查询是否有 www. sina. com 的记录。如果有该记录（例如，Web 浏览器刚刚访问过 www. sina. com，缓存中的记录系统还没有删除），则直接访问。

（2）如果在本地的缓存中没有找到相关的记录，客户端就会根据已设置的 DNS 服务器记录，向 DNS 服务器发出查询请求。如果该 DNS 服务器正好是创建 www. sina. com 记录的服务器，或在特定的时间段内处理过相同的查询，那么它就会从自己的区域记录或缓存中检索到该域名相应的资源记录（Resource Record，RR），并返回给客户端。

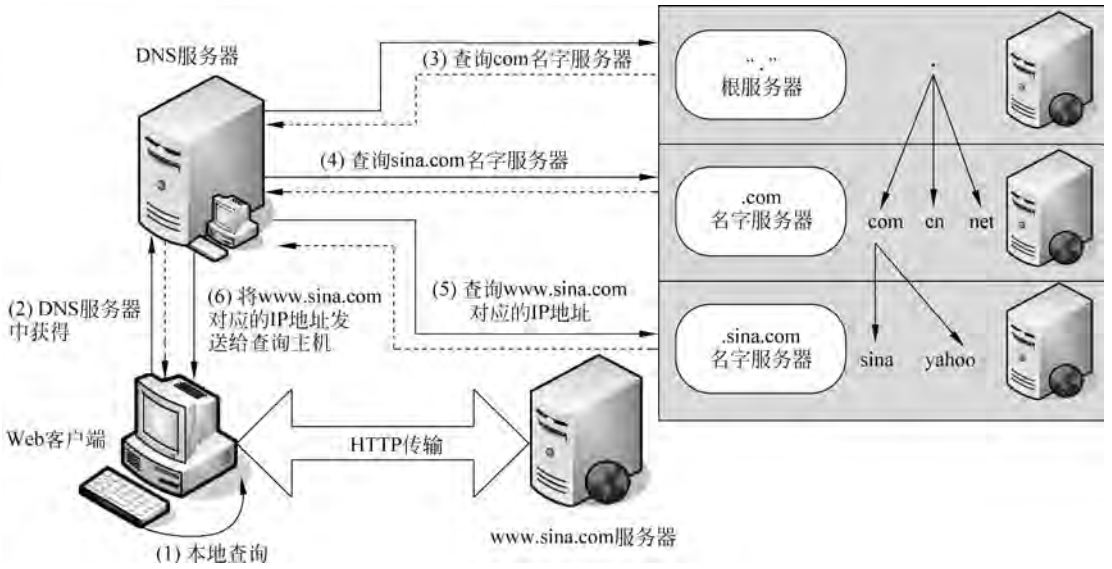

图 5-27 Internet 上对 www.sina.com 的访问过程

(3) 否则 DNS 服务器就将查询转发给根域服务器,由根域服务器找到 com 名字服务器地址,并发送给 DNS 服务器。

(4) DNS 服务器向 com 名字服务器继续发出查询 sina.com 地址的请求,com 名字服务器在找到 sina.com 的地址后,将结果发送给 DNS 服务器。

(5) DNS 服务器向 sina.com 名字服务器发出查询 www.sina.com 的请求,sina.com 名字服务器检索到 www.sina.com 对应的 IP 地址,并将结果发送给 DNS 服务器。

(6) DNS 服务器将 www.sina.com 对应的资源记录发送给 Web 客户端,Web 客户端利用 IP 地址访问相应的主机。

同时,在以上的递归查询过程中,Web 客户端、DNS 服务器以及各级的名字服务器都会记录这一次查询结果,以便下一次查询时直接调用。

5.6.2 DNS 的安全问题

DNS 服务是一种最基础的网络服务,但是 DNS 在设计之初并没有考虑安全问题,只是为了方便人们使用,简单地在域名与 IP 地址之间进行映射,并将映射记录提供给人们查询,DNS 内部没有为数据提供任何安全认证和数据完整性检查,留下了极大的安全隐患。如果 DNS 服务器被入侵者控制,则有可能篡改 DNS 服务器数据中 IP 地址与主机名之间的映射关系,从而会使主机遭受各类攻击(如 DOS 攻击、Web 欺骗攻击等),严重时有可能造成单位内部网络(Intranet)或 Internet 中域名解析的混乱。下面,介绍几种常见的 DNS 安全威胁。

1. 缓存中毒

DNS 为了提高查询效率,采用了缓存机制,把用户查询过的最新记录存放在缓存中,并设置生存周期(Time To Live,TTL)。在记录没有超过 TTL 之前,DNS 缓存中的记录一旦

被客户端查询，DNS服务器（包括各级名字服务器）将把缓存区中的记录直接返回给客户端，而不需要进行逐级查询，提高了查询速率。

DNS缓存中毒利用了DNS缓存机制，在DNS服务器的缓存中存入大量错误的数据记录主动供用户查询。由于缓存中大量错误的记录是攻击者伪造的，而伪造者可能根据不同的意图伪造不同的记录。例如，将查询指向某一个特定的服务器，使所有通过该DNS查询的用户都访问某一个网站的主页；或将所有的邮件指向某一台邮件服务器，拦截利用该DNS进行解析的邮件，等等。

由于DNS服务器之间会进行记录的同步复制，所以在TTL内，缓存中毒的DNS服务器有可能将错误的记录发送给其他的DNS服务器，导致更多的DNS服务器中毒。正如DNS的发明者Paul Mockapetris所说：中毒的缓存就像是“使人们走错方向的假冒路牌”。

2005年8月，数十万台互联网上的DNS服务器遭受到DNS缓存中毒的攻击，攻击者将存储在DNS服务器上的流行网站的IP地址更换为恶意网站的IP地址，将毫不知情的互联网用户由合法的网站引导到恶意网站，并要求用户透露机密信息或安装恶意软件。

DNS数据库对互联网上的用户是完全开放的，它既没有在DNS内部对数据提供认证机制和完整性检查，也没有对DNS服务器提供的服务进行访问控制和限制。所以攻击者可以将一些未经验证的数据存入DNS服务器的缓存中，同时当用户在DNS服务器上进行地址查询时，DNS服务器也不对用户进行任何身份认证。DNS的这种工作机制造成了大量的安全漏洞，使DNS遭受到了各种各样的安全攻击。

2. 拒绝服务攻击

DNS服务器在互联网中的关键作用使它很容易成为攻击者进行攻击的目标，加上DNS服务器对大量的攻击没有相应的防御能力，所以攻击过程很容易实现，且造成的后果非常严重。现在使用的DNS采用了树状结构，一旦DNS服务器不能提供服务，其所辖的子域都将无法解析客户端的域名查询请求。

对DNS服务器进行拒绝服务攻击比较容易。目前针对DNS服务器的拒绝服务攻击主要有两种方式：一种是直接攻击DNS服务器，将DNS服务器作为被攻击对象，由多台攻击主机向被攻击的DNS服务器频繁发送大量的DNS查询请求，最终使DNS服务器崩溃；另一种是利用DNS服务器作为“中间人”，去攻击网络中的其他主机。攻击者可以向多个DNS服务器发送大量的查询请求，这些查询请求数据分组中的源IP地址为被攻击者的IP地址。DNS服务器将大量的查询结果发送给被攻击主机，使被攻击主机无法提供正常的服务，如使DNS服务器无法为用户提供正常的查询等。

3. 域名劫持

域名劫持通常是指采用非法手段获得某一个域名管理员的账号和密码，或者域名管理邮箱，然后将该域名的IP地址指向其他的主机（该主机的IP地址有可能不存在）。域名被劫持后，不仅有关该域名的记录会被改变，甚至该域名的所有权可能落到其他人的手里。

2001年3月25日，“我要”（51.com）电子商务网站遭到攻击，其域名被删除长达一天之久。这是我国第一例涉嫌域名劫持的事件。就其事件的整个过程，就是攻击者首先通过电子邮件获得51.com网站的域名管理员账号和密码，然后删除正常的域名解析记录。

5.6.3 域名系统安全扩展

为了弥补 DNS 最初设计时存在的安全缺陷，1994 年，IETF 成立了 DNSSEC（DNS Security）工作组，通过在原有协议上增添 DNSSEC 部分，从整体上解决 DNS 的安全问题。1999 年 3 月，IETF 以 RFC 2535 文档发布了域名系统安全扩展（Domain Name System Security Extensions，DNSSEC），提出了解决 DNS 安全问题的一系列措施。

1. DNSSEC 的基本原理

域名系统安全扩展（DNSSEC）是在原有的域名系统（DNS）上通过公钥技术，对 DNS 中的信息进行数字签名，从而提供 DNS 的安全认证和信息完整性检验。具体原理如下。

发送方：首先使用 Hash 函数对要发送的 DNS 信息进行计算，得到固定长度的“信息摘要”；然后对“信息摘要”用私钥进行加密，此过程实现了对“信息摘要”的数字签名；最后将要发送的 DNS 信息、该 DNS 信息的“信息摘要”以及该“信息摘要”的数字签名，一起发送出来。

接收方：首先采用公钥系统中的对应公钥对接收到的“信息摘要”的数字签名进行解密，得到解密后的“信息摘要”；接着用与发送方相同的 Hash 函数对接收到的 DNS 信息进行运算，得到运算后的“信息摘要”；最后，对解密后的 “信息摘要”和运算后的“信息摘要”进行比较，如果两者的值相同，就可以确认接收到的 DNS 信息是完整的，即是由正确的 DNS 服务器得到的响应。

由此可以看出，在 DNSSEC 中所有返回给域名解析器（DNS 客户端程序）的响应都附加了数字签名。域名解析器通过数字签名来验证这些记录与权威的域名服务器上的记录是否完全一致。数字签名采用的是公钥加密系统，它产生的密钥对分为公钥和私钥两部分。其中，私钥需要保密存储，用来对区域文件中的 DNS 信息的“数字摘要”进行加密；公钥需要在 DNS 服务器上公开发布，域名解析器接收到域名服务器发送的响应记录后，使用公钥对响应记录中的数字签名进行解密，将得到的值与所接收到的 DNS 信息进行 Hash 运算获得的值进行对比，如果相同，说明该记录是合法的。

为了实现上述功能，DNSSEC 定义了 3 种资源记录（RR）：用于存放 DNS 信息数字签名的 SIG RR；用于存放解密公钥的 KEY RR；用于存放否定应答（即不存在资源记录）的 NXT RR。

2. DNSSEC 的工作机制

在 DNS 系统中，每一个 DNS 服务器可以管理一个区域。例如，com. cn 就是一个区域（Zone），在该区域上再创建子区域（称为“子域”），如 sina. com. cn、yahoo. com. cn 等。

DNSSEC 对 DNS 区域中的记录进行签名和验证是建立在对该区域信任的基础上。为了实现对区域密钥的信任，需要由父域对子域进行验证。DNS 系统是一个树状结构，DNS 客户端通过递归方式进行域名的查询，在一个完整的 DNS 递归查询过程中，第一个要查询的名字服务器为根域服务器（见图 5-27）。在 DNSSEC 系统中，DNS 客户端的域名解析器首先确保根域是可信任的，然后信任由根域签名的子域，并以此类推。

这种从根域开始，由上到下逐级签名验证的方式称为信任链（Trusted Chain），即父域

与子域之间逐级建立信任关系。由此可以看出，根域是整个 DNSSEC 的安全入口，每一个支持 DNSSEC 的 DNS 客户端解析器都需要建立与根域之间的信任关系，即需要安装根域的公钥。同时，根域以下的 DNS 服务器之间也要通过公钥系统建立信任关系。一般情况下，每一个区域通过相应的密码算法产生一个公钥/私钥对，并保存在该区域的一个权威名字服务器内。在域名查询过程中，当查询到某一个区域时，由该区域的权威名字服务器用自己的私钥对 DNS 信息的“摘要信息”进行数字签名，该区域（父域）的下级区域（子域）用公钥对签名数据进行解密。

图 5-28 所示为一个 DNSSEC 系统的查询和应答过程。其中，系统中的所有客户端和服务器都支持 DNSSEC，区域 abc. cn 的权威名字服务器为 auth. abc. cn，区域 xyz. net 的权威名字服务器为 auth. xyz. net。

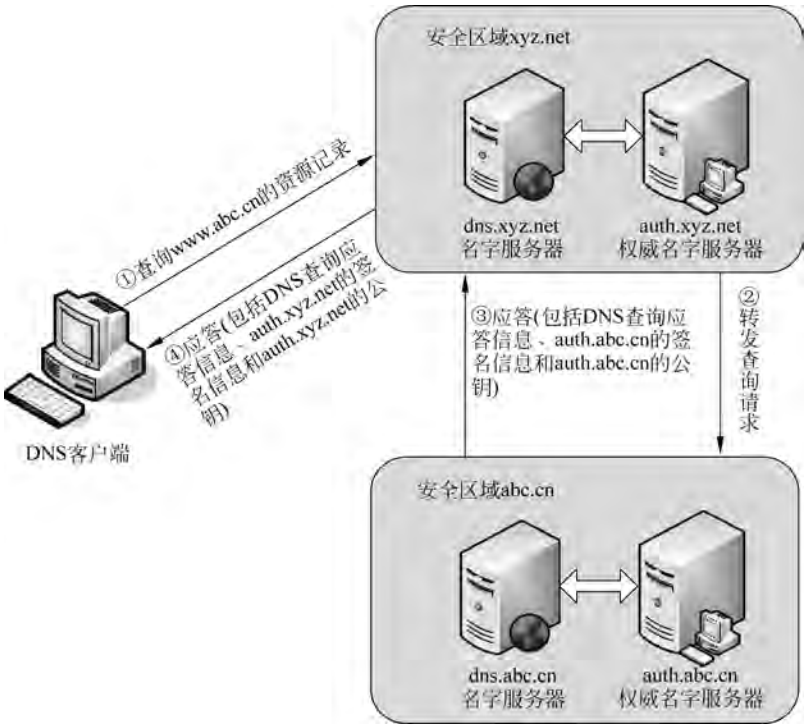

图 5-28　DNSSEC 系统的查询和应答过程

现在，客户端要查询 www. abc. cn，具体查询过程如下。由于 DNS 客户端设置的 DNS 服务器的地址指向了 dns. xyz. net，所以 DNS 客户端向名字服务器 dns. xyz. net 服务器发送一个查询请求，请求 www. abc. cn 的资源记录。假设 dns. xyz. net 服务器的缓存中没有该记录，dns. xyz. net 服务器便将该查询请求发给 dns. abc. cn 服务器，最后由 dns. abc. cn 服务器查询到了 www. abc. cn 的资源记录。因为安全区域 abc. cn 授权 auth. abc. cn 服务器管理本区域中的密钥，而安全区域 xyz. net 授权 auth. xyz. net 服务器管理本区域中的密钥，所以 auth. abc. cn 服务器将 www. abc. cn 查询的应答信息用自己的私钥进行签名并转发给 auth. xyz. net 服务器，auth. xyz. net 服务器用接收到的签名信息用 auth. abc. cn 服务器的公钥进行解密，以验证应答信息的安全性和完整性。当 auth. xyz. net 服务器通过验证后，再将应答信息转发给 DNS 客户端，此过程也要利用数字签名验证应答信息的安全性和完整性。

3. DNSSEC 的应用现状

DNSSEC 作为对目前 DNS 的安全扩展，可有效地防范 DNS 存在的各种攻击，保证客户端收到的 DNS 记录的真实性和完整性。此外，DNSSEC 与原有的 DNS 具有向下的兼容性，在实现上具有可行性。但是，由于 Internet 的特殊性，就像从 IPv4 到 IPv6 的迁移一样，从 DNS 到 DNSSEC 的转换不可能在短期内完成，需要一个渐进的过程。可以先有针对性地建立一些安全区域，如. cn、. net 等，然后再向其他区域扩展。当整个 Internet 部署了 DNSSEC 后，所有的信任将集中到根域下。

在 DNSSEC 系统中，不但 DNS 服务器要支持安全扩展功能，而且 DNS 客户端也要支持该功能。目前，基于 Windows Server 2012/2016、Linux、UNIX 等操作系统的 DNS 服务器都可以支持 DNSSEC，Windows XP/Vista、Linux 等较新操作系统的 DNS 客户端也已支持 DNSSEC。

但是，目前在推广 DNSSEC 上存在许多问题或困难。一是由于整个 Internet 上的 DNS 记录非常庞大，如果要部署适用于整个 Internet 的 DNSSEC，需要投入大量时间和设备，同时还要得到所有区域服务器提供商的支持；二是 DNSSEC 只是提供了对 DNS 记录真实性的验证，只在有限的程度上为用户通信的安全提供了保证，要完全保证用户信息的安全，还需要对应用程序和数据传输的各个环节加强其安全性；三是 DNSSEC 在 DNS 请求和应答中添加了数字签名，一方面增加了通信的流量和复杂性，另一方面，安全性主要依赖于公钥技术的安全性，所以对于 DNSSEC 系统来说是否会存在新的安全问题也是一个未知数。

5.6.4 DNS 系统的安全设置

DNS 系统的安全涉及 DNS 服务器、DNS 客户端、路由器、防火墙等设备或系统，下面仅介绍一些常用的安全技术和措施。

1. 选择安全性较高的 DNS 服务器软件

Internet 上大量的 DNS 服务器使用的是基于 UNIX/Linux 的 BIND 软件，目前最新版本为 BIND 9. x。最新版本的 BIND 软件支持许多安全特性，如支持 DNSSEC，解决了早期版本中存在的一些安全漏洞等。对于在 Internet 上的 DNS 服务器建议采用 BIND 软件，并将其升级为最新版本。

随着 Windows Server 操作系统的广泛使用，许多中小企业使用 Windows Server 操作系统自带的 DNS 软件组建 DNS 服务器。对于这些用户，一是建议使用 Windows Server 2012/2016 作为服务器操作系统，利用 Windows Server 2012/2016 操作系统自身的安全性增加 DNS 服务器的安全性；二是在一台 Windows Server 2012/2016 的域控制器上创建 DNS 服务器，这样可以利用活动目录的安全功能加强对 DNS 的安全管理。如果用户的 DNS 服务器建立在没有域的独立服务器上，建议将其升级为一台域控制器。

2. 限制端口

DNS 在工作时使用 UDP 53 和 TCP 53 端口进行通信。其中，DNS 服务器会同时监听这两个端口，DNS 客户端通过 UDP 53 端口与 DNS 服务器之间进行域名解析的请求和应答，而 TCP 53 端口用于 DNS 区域之间的数据复制。

为此，对于专用 DNS 服务器，可以通过防火墙或直接在操作系统只开放 UDP 53 和 TCP 53 两个端口，限制其他端口的通信，通过端口限制功能来加强系统的安全性。

根据教学需要，本章结合 TCP/IP 参考模型介绍重点介绍了 ARP、DHCP、TCP、DNS 协议的工作原理及存在的安全问题，并结合实际应用提出了一些可行的解决方法。其实，本章的内容仅是对 TCP/IP 参考模型中安全问题的一个初探，从目前的应用和研究来看，TCP/IP 中的每一个协议几乎都存在安全问题。为此，希望读者通过本章的学习，通过参阅相关资料，加深对通信协议的理解和分析，在以后的工作中实现提高网络的安全性。

习题 5

5-1 联系 OSI 参考模型，介绍 TCP/IP 参考模型的分层特点以及各层的功能。

5-2 结合图 5-4，描述网络中两台主机之间的通信过程。

5-3 结合 ARP 协议的工作特点，描述 ARP 欺骗的工作原理。

5-4 结合 TCP/IP 网络中计算机和交换机的工作原理，分别描述针对计算机和交换机的 ARP 欺骗的特点。

5-5 在中小网络中如何防范 ARP 欺骗？并通过实验进行验证。

5-6 结合 DHCP 的工作原理和网络运行实际，指出目前计算机网络中针对 DHCP 存在的主要安全问题及产生的危害。

5-7 如何通过对交换机和操作系统的设置加强 DHCP 服务的安全？

5-8 在掌握 TCP 传输 3 个过程的基础上，分析 TCP 协议存在的安全问题，并结合实际提出相应的解决方法。

5-9 结合 DNS 的工作过程，描述 DNS 缓存中毒、拒绝服务攻击、域名劫持的实现过程。

5-10 描述 DNSSEC 的工作原理和工作过程。

5-11 结合 TCP/IP 体系结构，简述存在安全隐患的原因。

5-12 结合互联网应用，试分别分析针对头部的安全、针对协议实现的安全、针对验证的安全和针对流量的安全产生的原因。