第5章

Web服务器的攻防

随着社交网络、微博、电子商务等各类 Web 应用的快速发展，针对众多 Web 业务平台的网络攻击频繁发生，Web 安全问题开始引起大家的普遍关注。由于 Web 应用程序的访问只需要通过客户端浏览器就可以完成，这就形成了一种新型的 B/S(Browser/Server，浏览器/服务器)结构。它在继承了传统 C/S(Client/Server，客户机/服务器)结构应用优势的基础上，根据 Web 应用需求进行了功能扩展和结构优化。同样地，各类网络攻击行为也随着体系结构和工作模式的变化而变化，新的应用环境不仅要解决传统网络中存在的安全问题，同时还要应对针对新应用而出现的新型攻击行为。考虑到浏览器/服务器的结构特点，本章重点介绍 Web 服务器的攻防，有关 Web 浏览器的攻防将在第 6 章单独介绍。

5.1 Web 应用的结构

体系结构是用于定义一个系统的结构组成及系统成员间相互关系的一套规则。从互联网应用发展来看，从早期的终端/主机模式到后来的共享数据模式，再到 C/S 模式，发展到目前以 B/S 模式为主，在电子商务等应用中使用的三层或多层模式，基于互联网应用的结构发生着巨大的变化。

5.1.1 C/S 结构

C/S 结构虽然不是目前互联网应用中的主流结构，但它是 B/S 结构的基础，一些在 B/S 结构中存在的攻击行为也源自于 C/S 结构。因此，本节将对 C/S 结构进行必要的介绍。

1. C/S 结构的实现方法

面向终端的网络以大型机为核心，而 C/S 结构打破了大型机在网络中所处的核心位置，通过充分发挥个人计算机(PC)、大型数据库系统和专业服务器操作系统(UNIX/Linux、

NetWare 和 Windows Server)的功能,实现了真正意义上的分布式计算模式。C/S 结构是指将事务处理分开进行的网络系统,即 C/S 的工作模式采用了两层结构:第一层在客户机系统上有机融合了表示与业务逻辑;第二层通过网络结合了数据库服务器。更具体地讲,C/S 结构将与用户交互的图形用户界面(Graphical User Interface,GUI)、业务应用处理与数据库访问、处理相分离,服务器与客户机之间通过消息传递机制进行对话,由客户机向服务器发出请求,服务器在进行相应的处理后经传递机制向客户机返回应答。

2. C/S 结构的特点

大多数情况下,C/S 结构是以数据库应用为主,即业务数据库(如 Oracle、MS SQL、MySQL 等)运行在服务器端,而数据库应用程序运行在客户端。基于这一特定的应用环境,C/S 结构存在如下优缺点:

(1) C/S 结构的主要优点。

① 交互性强。在 C/S 结构中,客户端运行有一套完整的应用程序,大量的业务应用一般在客户端完成,只有在进行数据调用和写入时才与服务器进行交互。

② 具有较强的数据操纵与事务处理能力。C/S 结构通过将任务合理分配到客户端和服务器端,降低了系统对网络带宽的占用,并可以充分利用客户机和服务器的硬件资源。

③ 可有效保护数据的安全性。每一个用户与服务器之间采用点对点通信模式,可采用安全性较高的通信协议或加密方式,以保护数据的安全性。

(2) C/S 结构的主要缺点。

① 可扩展性较差。当因系统功能扩展等原因需要升级系统软件时,将同时涉及服务器端和客户端,而且还会造成业务的中断,系统的升级和维护都较复杂,可扩展性较差。

② 应用规模受限。随着网络规模不断扩大,应用程序的复杂度越来越高,客户端与后台数据库之间需要频繁的交换数据,服务器容易成为应用的瓶颈。

目前,C/S 结构一般应用于用户群相对固定、对数据安全要求相对较高的企业内部业务系统。

5.1.2 B/S 结构

随着 Internet 的迅速普及和发展,尤其是 Web 技术的不断成熟和应用领域的快速拓展,传统 C/S 结构存在的不足逐渐显现,导致了整个互联网应用系统体系结构从 C/S 主从结构向更加灵活的 B/S 多级分布式结构的演进。

1. B/S 结构的实现方法

基于 B/S 结构的应用系统由前端的浏览器(Browser)和后端的服务器(Server)组成,数据和应用程序都存放在服务器上,浏览器是一种通用的客户端软件,结合多种脚本语言(如 VBScript、JavaScript 等)和 Active X 等技术,实现了原来需要复杂的专用软件才能实现的功能。在 B/S 结构中,用户界面完全通过 Web 浏览器软件(如 IE、Firefox、Chrome 等)实现,一部分事务逻辑在前端实现,而主要事务逻辑在服务器端实现。客户端利用 Web 浏览器下载应用,并在浏览器上执行和显示。因此,B/S 结构是对传统 C/S 结构的发展和演进。

如图 5-1 所示,B/S 结构是由表示层、业务逻辑层和数据层组成的典型的三层体系结

构。表示层为浏览器，仅承担网页(page)信息的浏览功能，以 HTML 实现信息的浏览和输入，一般不具备业务处理能力。业务逻辑层由服务器承担业务处理逻辑和页面的存储管理，接收客户端浏览器的任务请求，并根据请求类型执行相应的事务处理程序。而数据层由数据库服务器承担数据处理逻辑，其任务是接收服务器对数据库服务器提出的数据操作请求，由数据库服务器完成数据的查询、修改、统计、更新等工作，并将数据处理结果提交给服务器。

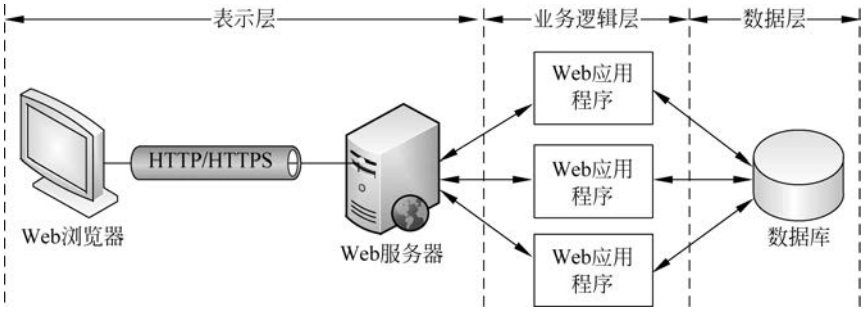

图 5-1　B/S 的三层(3-tiers)体系结构的组成

服务器端由 Web 服务器软件、Web 应用程序及后端数据库构成。Web 服务器(Web Server)软件通常被称为 HTTP 守护程序，接收 Web 客户端对资源的请求，在这些请求上执行一些基本的解析处理以确定资源是否存在，然后将结果传送给 Web 应用程序来执行，在 Web 应用程序执行结束并返回响应时，Web 服务器再将这个响应返回给 Web 客户端。浏览器使用 HTTP/HTTPS，HTML 语言与 Web 服务器进行交互，获取 Web 服务器上的信息和应用程序，并在本地执行、渲染和显示。

Web 应用程序(Web Application)是处于服务器端的业务逻辑，是现代 Web 应用的核心。早期静态 Web 应用程序只有一层，用于提供客户端浏览器显示的页面。随着 Web 技术的发展，Web 应用程序的功能越来越强大，同时结构也越来越复杂，出现了多层(n-tiers)体系结构的概念。

数据库也称为后台数据库，是 Web 应用程序多层体系结构的最后一层。目前，典型的后台数据库管理系统软件主要有 MS SQL、MySQL、Oracle 等，它们都支持统一的数据库查询语言 SQL。随着互联网技术的发展，数据库技术实现了与 Web 技术的融合，促使 Web 应用程序从早期由 HTML 为主体的静态应用向由各种 Web 应用技术所驱动的动态应用的转变，促进了支持信息检索和在线电子交易等 Web 应用的快速发展。

目前，在具体的应用中多采用如图 5-2 所示的由 B/S 和 C/S 组成的混合体系结构。其中，类似于信息发布和查询等满足大部分用户需要的应用以 B/S 方式实现，而一般由系统管理员负责的后台数据库管理与系统维护等操作采用 C/S 结构，通过 ODBC 连接。该结构充分发挥了 B/S 结构和 C/S 结构的优点，弥补了各自存在的不足。

2. B/S 结构的特点

(1) B/S 结构的主要优点。

① 统一了客户端应用软件。B/S 客户端只需要安装统一的浏览器软件，避免了 C/S 结构中安装功能各异的各类数据库客户端软件和应用软件带来的管理和维护上的困难。

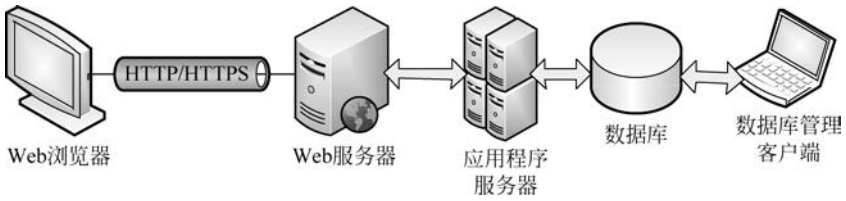

图 5-2 B/S 和 C/S 组成的混合体系结构

② 易于部署和维护。由于客户端不需要安装专用软件，应用系统的升级只需要考虑服务器端，用户在连接到服务器时只需要下载更新就可以实现升级，大大降低了总体拥有成本(Total Cost of Ownership，TCO)。

③ 可扩展性好。在 B/S 结构中，Web 浏览器和 Web 站点(由 Web 服务器、Web 应用程序及数据库所构成)之间的通信采用了标准的 HTTP/HTTPS，具有良好的可扩展性。

④ 信息共享度高。HTML 是一个开放的标准和规范，它通过标记符号来标记要显示的网页中的各个部分，网页文件通过在文本文件中添加标记符来告诉浏览器如何显示其中的内容，浏览器按顺序阅读网页文件，然后根据标记符解释和显示其标记的内容。这种模式提供了更加丰富的显示内容和便捷的信息交互方式。

(2) B/S 结构的主要缺点。

① 功能受限。由于浏览器只是为了进行 Web 浏览而设计的通用客户端软件，而对部分客户端需要进行的在线大数据量处理(如数据录入、特殊要求界面的显示等)等功能无法实现或实现起来较为困难。

② 复杂的应用构造困难。虽然可以使用 ActiveX、Java 等技术开发较为复杂的应用，但是实现起来较为复杂。

③ 安全隐患较大。根据软件任务的不同，有些应用在用户首次访问时需要通过安装“插件”来实现，这为计算机病毒、木马等恶意代码的入侵提供了便利条件。另外，在应用系统没有升级到 HTTPS 的情况下，HTTP 在传输敏感信息时存在安全隐患。

5.1.3 Web 应用安全结构概述

结合 Web 应用体系，本节从攻防的角度提供如图 5-3 所示的安全结构。其中，Web 浏览器安全主要涉及 Web 浏览器软件安全、Web 用户安全和客户端操作系统安全 3 个方面。有关操作系统的安全在第 2 章和第 3 章已经进行了介绍，在第 6 章将重点介绍 Web 浏览器软件安全和 Web 用户安全两部分内容。

Windows Server 和 Linux 是目前典型的两款 Web 服务器操作系统，从整体上来讲，Linux 的安全性要优于 Windows Server，但两类操作系统同样都存在着远程渗透攻击和本地渗透攻击等安全威胁。

严格地讲，涉及计算机网络安全的所有内容几乎都适用于 Web 应用环境，但是为了突出 Web 应用的特点，下面重点针对 HTTP 自身存在的安全威胁进行分析，并提出相应的解决办法。例如，针对 HTTP 明文传输带来的敏感信息被监听甚至被篡改这一安全威胁，可通过对重要数据(如用户名、密码等)进行加密，或者直接使用 HTTPS 等方式解决。针对

Web浏览器软件安全	网络安全	Web数据安全
		Web应用程序安全
Web用户安全		Web服务器软件安全
客户端操作系统安全		服务器操作系统安全

图 5-3 Web 应用的安全结构

网络层、传输层和应用层存在的拒绝服务攻击、假冒身份攻击等威胁，将在第 7 章进行介绍。

由于 Web 应用的安全同时涉及客户端、服务器端和网络各个环节，每一个环节又涉及具体的细节，因此，Web 应用安全是一个同时涉及计算机网络、操作系统、应用程序、数据库等方面的复杂的安全系统。本章重点讨论 Web 服务器的安全，主要包括 Web 数据安全、Web 应用程序安全、Web 服务器软件安全和服务器操作系统安全等，并结合具体应用介绍相应的安全防范方法。

5.2 针对 Web 服务器的信息收集

“知己知彼，百战不殆”，意思为如果要想打起仗来不会有危险，就需要事先对敌我双方的详细情况进行全面的了解和分析。攻击者在确定了攻击对象后，只有在实施攻击前全面掌握被攻击 Web 服务器的详细配置信息，才能从中发现可利用的安全漏洞，进而确定具体的攻击方法。

5.2.1 需要收集的信息内容

在实施攻击之前或攻击过程中需要收集的信息很多，不同的攻击目标需要收集的信息侧重点也不尽相同。针对 Web 服务器的攻击，可收集的信息主要包括以下几类。

（1）地址信息：包括服务器的 IP 地址、DNS 域名、打开的端口号及对应的服务进程等。

（2）系统信息：包括操作系统类型及版本、Web 服务器软件类型及版本、Web 应用程序及版本、Web 应用程序的开发工具及版本、Web 应用程序架构（是静态 HTML 页面，还是 PHP、APS、JSP 动态页面等）、数据库管理系统的类型及版本等。

（3）账户信息：包括操作系统的登录账户、数据库管理系统的账户、应用系统的管理账户等。

（4）配置信息：包括网络拓扑结构、地址映射表（当 Web 服务器位于内部局域网中使用私有 IP 地址时）、服务配置信息、共享资源、防火墙类型及配置信息、身份认证与访问控制方式、加密及密码管理机制等。

（5）其他信息：包括安全漏洞（软件漏洞和管理漏洞）、DNS 注册信息、网络管理员联系

方式等。

5.2.2 网络踩点

针对被攻击对象信息的收集方法和途径很多，收集信息的效率和效果也各不相同。本章介绍常用的3类方法，即网络踩点、网络扫描和网络查点。

网络踩点(footprinting)是指攻击者对被攻击目标进行有目的、有计划、分步骤的信息收集和分析过程。通过网络踩点可以掌握被攻击目标的完整信息，并从中发现存在的安全隐患，为进一步实施攻击提供帮助。网络踩点常用的技术分为Web信息收集、地址信息查询和网络拓扑探测3种方式。

1. Web信息收集

Web信息收集是利用Web搜索引擎提供的功能，对被攻击目标(组织或个人)的公开信息进行收集并发现为进一步实施攻击有用的信息。对于攻击者来说，从互联网的海量信息中收集与攻击目标有关的信息是一种最为直接的网络踩点方法，强大的Web搜索引擎可以帮助攻击者实现这一目的。

目前，Google、Baidu、Yahoo、Bing等搜索引擎都提供了相应的信息搜索功能，综合运用这些功能可以帮助攻击者获得所需要的网上信息。例如，利用搜索引擎的基本搜索功能，攻击者可以方便地找到被攻击组织的Web主页或个人的博客等信息，这些页面一般会向攻击者提供大量的有用信息，为进一步挖掘相关信息提供帮助。

每一种搜索引擎都存在搜索功能、范围和效果上的差异性，为了能够综合不同搜索引擎的优势进行信息的收集，提出了“元搜索引擎”的概念。所谓元搜索引擎，又称为集合型搜索引擎，是指将多个单一搜索引擎集成在一起，提供统一的检索接口，将用户的检索提问同时提交给多个独立的搜索引擎，并进一步对多个独立搜索引擎的检索结果进行处理(包括去重、排序等)，最后将处理结果提交给攻击者。

2. 地址信息查询

MAC、IP和DNS是互联网赖以运转的基础，MAC地址标识了物理设备的唯一性，IP地址用于标识互联网信息节点在全局范围内的位置，而DNS实现了网站域名与IP地址之间的映射。其中，由于MAC地址范围的局限性，其使用中的地址信息可以保存在所在局域网的设备数据库中，在需要时经授权后查询；而DNS和IP地址的相关信息都保存在互联网上的公共数据库中，以公开方式对外发布，供公众查询。

在网络踩点中多采用DNS和IP查询方法。利用DNS和IP地址信息，攻击者可以获得攻击目标的互联网位置信息及相关联的地理位置信息。例如，利用Whois可以在互联网上查询到DNS注册信息，一般包括注册人和注册商的详细信息，通过这些信息便可以获得注册人(单位或个人)的具体地理位置等信息。又如，利用IP Whois可以在互联网查询到DNS注册人所使用的IP地址、通信地址、联系电话等信息。有了这些信息，攻击者可以掌握攻击目标的网络空间和社会空间信息，为进一步实施物理攻击或社会工程学攻击提供帮助。

3. 网络拓扑探测

网络拓扑反映了网络中各信息节点之间的关系，网络边界如何部署，内部网络如何组织，这些信息都会反映在网络拓扑中。尤其在网络边界处是否部署了防火墙、IDS、IPS、网络态势感知等安全系统，对后续实施攻击的要求和步骤都会产生非常大的影响。

路由路径跟踪是实现网络拓扑探测的主要手段，Linux 操作系统中的 traceroute 和 Windows 环境中的 tracert 程序分别提供了不同平台上的路由路径跟踪功能。两者的实现原理相同，都是用 TTL（IP 生存时间）字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由，从而确定 IP 数据包访问目标 IP 所采取的路径。当对目标网络中的不同主机进行相同的路由跟踪后，攻击者就可以综合这些路径信息，绘制出目标网络的拓扑结构，并确定关键设备在网络拓扑中的具体位置信息。

5.2.3 网络扫描

网络踩点确定的是攻击目标所在的网络和地理位置，而网络扫描则针对的是攻击目标的细微信息。网络扫描（scanning）是网络攻击的一个重要环节，首先通过“主机扫描”发现攻击目标网络中存在的活跃主机，然后通过“端口扫描”找出活跃主机上所开放的端口及对应的网络服务，接着通过“系统类型探测”确定攻击主机的操作系统类型及版本号，最后通过“漏洞扫描”找到攻击主机上存在的安全漏洞。

1. 主机扫描

主机扫描（Host Scan）是指通过对目标网络（一般为一个或多个 IP 网段）主机 IP 地址的扫描，以确定目标网络中有哪些主机处于运行状态。主机扫描的实现一般是借助于 ICMP、TCP、UDP 等协议的工作机制，以此来探测并确定某一主机当前的运行状态。

（1）基于 ICMP 的扫描方法。ICMP（Internet Control Message Protocol，Internet 控制报文协议）是 TCP/IP 协议栈的网际层提供的一个为主机或路由器报告差错或异常情况的协议。PING（Packet InterNet Groper，分组网间探测）是 ICMP 的一个重要的应用功能，它是应用层直接调用网际层 ICMP 的一个特殊应用，通过使用 ICMP 回送请求与回送应答报文来探测两台主机之间网络的连通性。

目前，几乎所有的操作系统和路由器都集成了 PING 命令。如果要知道某一台主机当前是否处于运行状态，最简单的办法是用 PING 命令来探测从本机到目标主机之间的网络连通性是否正常。例如，如果要知道主机 www.sina.com.cn 是否处于运行状态，只需要在命令提示符下运行 ping www.sina.com.cn 命令，根据显示信息就可以做出判断，如图 5-4 所示。

PING 命令利用了 ICMP 中的 Echo Request（回送请求）报文进行连通性探测，如果目标主机处于运行状态，在收到 Echo Request 报文后将返回 Echo Reply（回送应答）报文；如果目标主机存在但当前未处于运行状态，则返回 Echo Request Timed Out（请求超时）报文；如果目标主机不存在，则返回 Destination Host Unreachable（目标主机不可达）报文。

PING 命令在小型网络中的应用效果较好，但由于需要对目标主机进行逐一探测，因此在大型网络中的应用效率较低。另外，当目标主机开启了防火墙（操作系统自带的防火墙功

图 5-4 使用 PING 命令来探测目标主机是否处于运行状态

能)或目标主机前端设置了防火墙,并启用了对 ICMP 报文的过滤策略时,PING 命令将失去功能。

(2) 基于 TCP 的主机扫描方法。TCP(Transmission Control Protocol,传输控制协议)是一种面向连接的、可靠的、基于字节流的传输层通信协议。每一个 TCP 通信都需要有连接建立、数据传输和连接释放这 3 个过程,其目的是让通信的双方都知道彼此的存在,并通过双方协商来确定具体的通信参数(如缓存大小、连接表中的项目、最大窗口值等)。

TCP 连接的建立采用 C/S 模式,且要通过三次握手过程,具体实现过程如图 5-5 所示。其中,Client 主动打开连接,希望与 Server 建立 TCP 连接,所以在此过程中 Server 是被动打开连接,并处于"监听"(LISTEN)状态,等待 Client 的连接请求。同时,Client 的客户进程和 Server 的服务器进程分别创建了 TCB(Transmission Control Block,传输控制模块),用于存储本次连接中的一些重要信息,如 TCP 连接表、指向发送和接收缓存的指针、指向重传队列的指针、当前的发送和接收序号等。在此基础上,Client 和 Server 将开始以下的三次握手过程。

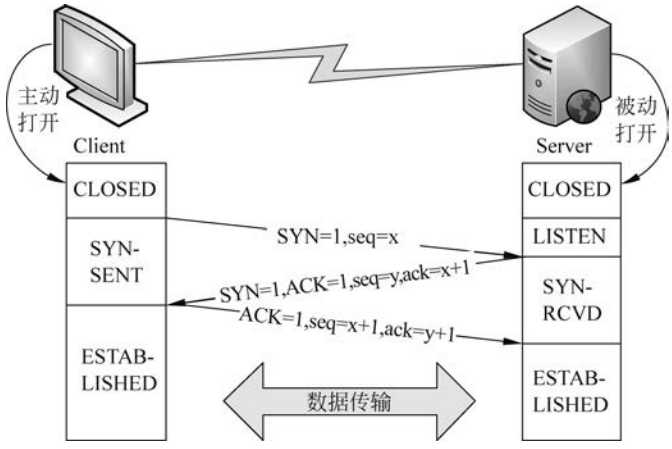

图 5-5 TCP 连接的三次握手过程

① 第一次握手:Client 向 Server 发送一个连接请求报文,该报文头部的"同步"(SYN)字段设置为 1,即 SYN=1,同时选择一个初始序号 seq=x。此时,TCP 客户端进程进入"同

步已发送”(SYN-SENT)状态。

② 第二次握手：Server 在接收到请求报文后，如果同意建立连接，则向 Client 返回一个确认报文。该确认报文中将头部的“确认”(ACK)和“同步”(SYN)字段都设置为 1，确认号为 ack＝x＋1，同时为自己选择一个初始序号 seq＝y。此时，TCP 服务器进程进入“同步收到”(SYN-RCVD)状态。

③ 第三次握手：Client 在接收到 Server 的应答报文后，还要向 Server 进行确认。该确认报文头部的“确认”(ACK)字段设置为 1，确认号 ack＝y＋1，而自己的序号 seq＝x＋1。此时，TCP 连接已经建立，Client 进入“连接已建立”(ESTAB-LISHED)状态。当 Server 接收到 Client 的确认报文后，也进入“连接已建立”(ESTAB-LISHED)状态。

由 TCP 连接的建立过程可知，可以使用 TCP 的 ACK 和 SYN 功能来探测主机的运行状态，即针对主机的 ACK 扫描和 SYN 扫描。

一种是 ACK 扫描。在三次握手过程中，ACK 表示 Server 对 Client 请求建立的确认，但如果 Client 根本没有进行 SYN 请求(第一次握手)，而是直接进行确认(第三次握手)，Server 就会认为出现了一个重要的错误，便向 Client 发送一个头部“复位”(RST)字段为 1 的报文，告诉 Client 必须释放本次连接，再重新建立 TCP 连接。根据该工作机制，如果攻击者向目标主机发送一个只有 ACK 的报文，当接收到目标主机一个 RST 反馈报文时，就可以确认目标主机的存在。

另一种是利用三次握手过程的针对主机的 SYN 扫描。如果目标主机处于运行状态，但主机上的服务器进程没有打开，则目标主机将返回一个 RST 报文；如果目标主机上的服务器进程处于“监听”(LISTEN)状态，则会返回一个第二次握手的 ACK/SYN 报文。不管返回哪一种报文，都可以从中判断目标主机的当前状态。

(3) 基于 UDP 的主机扫描方法。UDP(User Datagram Protocol，用户数据报协议)是一个无连接(没有提供三次握手过程)的、尽最大努力交付(不可靠)的、面向报文(保留了报文的边界)的传输层通信协议。如图 5-6 所示，UDP 报文的头部只有源端口、目的端口、长度及校验和 4 个字段，每个字段 2 字节，共 8 字节。其中，“源端口”只有在需要对方回复时才选用，不需要时全部置 0；“目的端口”供在目的主机上交付报文时使用，如果接收方的 UDP 发现收到的报文中的目的端口号不正确(不存在该端口号对应的应用进程)，就会丢弃该报文，并由 ICMP 向发送方返回一个“端口不可达”的差错报文。

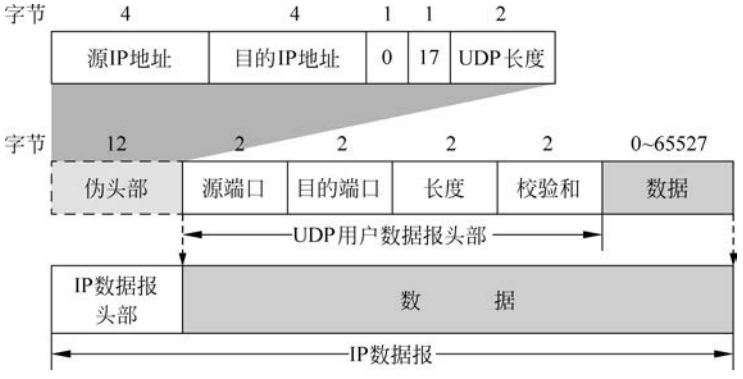

图 5-6　UDP 用户数据报的头部

基于UDP的工作原理，攻击者可以向一个开放的UDP端口发送一个带有“源端口”的报文，或者向一个未开放的UDP端口发送一个无法交付的UDP报文，根据返回的信息就可以判断目标主机的运行状态。

2. 端口扫描

端口扫描(Port Scan)是对正在处于运行状态的主机使用的TCP/UDP端口进行探测的技术。端口用于标识计算机应用层中的各个进程在与传输层交互时的层间接口地址，两台计算机间的进程在通信时，不仅要知道对方的IP地址，还要知道对方的端口号。因此可以将端口理解为进入计算机应用进程的窗口，端口在TCP和UDP中用16位表示，其值为0～65535。传输层的端口分为服务器端使用的端口号和客户端使用的端口号两大类。其中，服务器端使用的端口号又分为两类：一类称为熟知端口号(Well Known Ports)或系统端口号，其值为0～1023，可以在http://www.iana.org网站上查到；另一类称为登记端口号，其值为1024～49151，使用这类端口时需要在IANA(the Internet Assigned Number Authority，互联网数字分配机构)进行登记，以防止重用。客户端使用的端口号称为短暂端口号，其值为49152～65535，仅在客户进程运行时临时使用，通信结束后收回。

由于TCP和UDP可以使用相同的端口号(如DNS同时使用了TCP 53和UDP 53两个端口号)，因此端口扫描需要分别针对TCP和UDP的端口号进行扫描。同时由于TCP要比UDP复杂，因此TCP端口扫描也要比UDP端口扫描复杂。TCP端口扫描包括连接(connect)扫描、SYN扫描、TCP窗口扫描、FIN扫描、ACK扫描等，下面主要介绍TCP的连接扫描和SYN扫描，以及UDP端口扫描。

(1) 连接扫描。如图5-5所示，攻击者(扫描主机)通过系统调用connect()函数，可以与目标主机的每个端口尝试通过三次握手建立TCP连接，在攻击者发起连接请求(第一次握手)后，如果目标主机上对应的端口打开，则返回一个第二次握手的ACK/SYN报文，connect()调用将再发送一个ACK确认报文以完成第三次握手。如果目标端口是关闭的，那么目标主机将会直接返回一个RST报文。基于此工作原理，通过分析不同目标端口的返回报文信息，攻击者就可以判断哪些端口是开放或关闭的。该方法实现简单，但目标主机上会记录相关的尝试连接信息，容易被系统管理员或安全检测工具发现。

(2) SYN扫描。SYN扫描也称为半开连接扫描，是对连接扫描的一种改进。在连接扫描方法中，当被扫描端口打开时，目标主机会返回一个SYN/ACK报文。当攻击者收到第二次握手的SYN/ACK报文时，其实不需要进行第三次ACK握手，就已经能够判断出被扫描端口当前处于打开状态。不过，当目标主机(Server)向TCP连接请求者(Client)返回SYN/ACK报文后，目标主机将处于“半开连接”状态，等待请求者的ACK确认，以便完成第三次握手过程。此时，攻击者并没有向目标主机返回ACK确认报文，而是构造了一个RST报文，让目标主机释放该“半开连接”。

由于各类操作系统一般不会记录“半开连接”信息，因此SYN扫描的安全性要比连接扫描好。

(3) UDP端口扫描。UDP端口扫描用于探测目标主机上打开的UDP端口和网络服务。UDP端口扫描的实现原理是：首先构造并向目标主机发送一个特殊的UDP报文，如果被扫描的UDP端口关闭，将返回一个基于ICMP的“端口不可达”差错报文；如果被扫描的UDP端口处于打开状态，处于“监听”状态的UDP网络服务将响应一个特殊格式的数据

报文，并返回 UDP 数据。

UDP 端口扫描的实现原理简单，效率较高。但是，如果被探测的网络服务是一个未知的应用时，就可能无法返回 UDP 数据。

3. 系统类型探测

通过主机扫描和端口扫描，可以确定被攻击目标使用的 IP 地址及开放的端口。在此基础上，还需要对被攻击主机所使用的操作系统类型和具体的版本号及提供的网络服务进行探测，为攻击者下一步选择具体的攻击方法并实施具体的攻击做好准备。系统类型探测分为操作系统类型探测和网络服务类型探测两种类型。

（1）操作系统类型探测。操作系统类型探测（OS Identification）是通过采取一定的技术手段，通过网络远程探测目标主机上安装的操作系统类型及其版本号的方法。在确定了操作系统的类型和具体版本号后，可以为进一步发现安全漏洞和渗透攻击提供条件。

协议栈指纹分析（Stack Fingerprinting）是一种主流的操作系统类型探测手段，其实现原理是在不同类型和不同版本的操作系统中，网络协议栈的实现方法存在着一些细微的区别，这些细微区别就构成了该版本操作系统的指纹信息。通过创建完整的操作系统协议栈指纹信息库，将探测或网络嗅探所得到的指纹信息在数据库中进行比对，就可以精确地确定目标主机上操作系统的类型和版本号。

（2）网络服务类型探测。网络服务类型探测（Service Identification）的目的是确定目标主机上打开的端口及该端口上绑定的网络应用服务类型及版本号。通过网络服务类型探测，可以进一步确定目标主机上运行的网络服务及服务进程对应的端口。

操作系统类型探测主要依赖于 TCP/IP 协议栈的指纹信息，它涉及网络层、传输层、应用层等各层的信息，而网络服务类型探测主要依赖于网络服务在应用层协议实现所包含的特殊指纹信息。例如，同样是在应用层提供 HTTP 服务的 Apache 和 IIS，两者在实现 HTTP 规范时的具体细节上存在一些差异，根据这些差异就可以辨别出目标主机的 TCP 80 端口上运行的 HTTP 服务是通过 Apache 实现的，还是通过 IIS 实现的。

nmap 提供了强大的系统类型探测功能，可以精确地探测出目标主机的操作系统类型和版本号，以及网络服务类型和版本号。图 5-7 是利用 nmap 工具对一台内部主机进行探测后显示的信息，显示的内容包括主机名、开放的端口及版本号、操作系统类型及版本号等信息。

```
nmap -sV -T4 -O -F --version-light 172.17.200.27
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-01 16:10 ?D1ú±ê×?ê±??
Nmap scan report for www.jspi.cn (172.17.200.27)
Host is up (0.012s latency).
Not shown: 93 closed ports
PORT     STATE    SERVICE      VERSION
22/tcp   open     ssh          OpenSSH 4.3 (protocol 2.0)
80/tcp   open     http         Apache Tomcat/Coyote JSP engine 1.1
111/tcp  open     rpcbind
135/tcp  filtered msrpc
139/tcp  filtered netbios-ssn
445/tcp  filtered microsoft-ds
8009/tcp open     ajp13        Apache Jserv (Protocol v1.3)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.27

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 22.09 seconds
```

图 5-7　nmap 工具对目标主机系统类型进行探测的结果

5.2.4 漏洞扫描

漏洞扫描是一种通过扫描方式发现目标网络或特定主机上存在已知安全漏洞的技术手段。漏洞扫描是网络扫描的最后一个环节，也是整个攻击过程中最关键的一个环节。漏洞扫描的目的是探测并发现目标网络中特定主机上运行的操作系统、网络服务与应用程序中存在的安全漏洞，并根据已确定的漏洞来选择相应的渗透工具，实施对目标主机的渗透攻击。

1. 漏洞扫描的原理

漏洞扫描除用于网络攻击外，还用于对网络的安全防御。系统管理员通过对网络漏洞的系统扫描，全面地了解网络的安全现状，并对发现的安全漏洞及时安装补丁程序，提升网络的安全性。

漏洞扫描技术的工作原理是基于目标对象（操作系统、网络服务、应用程序等）的特征码来实现的。例如，对于同一个类型和版本号的操作系统来说，针对某一安全漏洞，对于某些网络请求的应答，安装安全补丁前后会存在一些细微的差异，这些差异便构成了针对特定安全漏洞的特征码（指纹信息）。漏洞扫描技术正是利用了这些特征码来识别目标对象是否存在特定的安全漏洞。

2. 漏洞扫描器

网络漏洞扫描器对目标系统进行漏洞检测时，首先探测目标网络中的存活主机，再对存活主机进行端口扫描，确定系统已打开的端口，同时根据协议栈指纹技术识别出主机的操作系统类型。然后，扫描器对开放的端口进行网络服务类型的识别，确定其提供的网络服务。漏洞扫描器根据目标系统的操作系统平台和提供的网络服务，调用漏洞资料库（一般该资料库需要与业界标准的 CVE 保持兼容）中已知的各种漏洞进行逐一检测，通过对探测响应数据包的分析判断是否存在漏洞。

现有的网络漏洞扫描器主要是利用特征匹配的原理来识别各种已知的漏洞。扫描器发送含有某一漏洞特征探测码的数据包，根据返回数据包中是否含有该漏洞的响应特征码来判断是否存在漏洞。因此，只要在研究了各种漏洞且知道不同漏洞的探测特征码和响应特征码后，就可以利用软件来实现对各种已知漏洞的模拟。漏洞扫描器一般由以下几部分组成。

（1）安全漏洞数据库。该数据库一般与 CVE（Common Vulnerabilities and Exposures，通用漏洞披露目录）保持兼容，主要包含安全漏洞的具体信息、漏洞扫描评估的脚本、安全漏洞危害评分（一般采用 CVSS 通用漏洞分组评价体系标准）等信息，新的安全漏洞被公开后，数据库需要及时更新。其中，CVSS（Common Vulnerability Scoring System，通用漏洞评分系统）是一个开放的并且能够被产品厂商免费采用的标准。

（2）扫描引擎模块。作为漏洞扫描器的核心部件，扫描引擎模块可以根据用户在配置控制台上所设定的扫描目标和扫描方法，对用来扫描网络的请求数据包进行装配与发送，并将从目标主机接收到的应答包与漏洞数据库中的漏洞特征码进行比对，以判断目标主机上是否存在这些安全漏洞。为了提高效率，扫描引擎模块一般提供了主机扫描、端口扫描、操

作系统扫描、网络服务探测等功能，供具体扫描时根据需要选用。

（3）用户配置控制台。供用户进行扫描设置的操作窗口，要扫描的目标系统、要检测的具体漏洞等信息都可以通过配置控制台来设置。

（4）扫描进程控制模块。为了让使用者更加直观地了解扫描过程的进展情况，漏洞扫描器都会提供扫描进程控制模块，以显示当前扫描进程任务的进展情况。

（5）结果存储与报告生成模块。根据漏洞扫描结果自动生成扫描报告，告知用户从哪些目标系统上发现了哪些安全漏洞。

5.2.5 网络查点

通过网络踩点，攻击者可以确定目标对象的网络位置及个人或组织的相关信息；通过网络扫描，攻击者可以发现目标网络中处于运行状态的主机、主机上运行的操作系统及网络服务的类型与版本号，并通过漏洞扫描发现目标对象存在的安全漏洞。这些信息为实施网络渗透攻击提供了依据。但是，对于一次具体的攻击过程来说，在实施渗透攻击之前，还需要系统地分析已经发现的弱点，对于已经识别出来的系统安全漏洞和服务还要进行更充分和有针对性的探测，以便从中找到可以攻击的入口。在攻击过程中，还需要提供哪些数据和辅助工具，也需要事先做好准备。为此，在网络踩点和网络扫描之后，具体实施渗透攻击之前，将该阶段所做的准备工作称为网络查点（enumeration）。

网络踩点一般是通过一些常用的技术和工具，在目标网络外围进行的信息收集行为。而网络查点主要对目标系统进行的主动连接与查询。从攻击者的角度来看，网络查点要比网络踩点的入侵程度深，而且网络查点行为可能会记入系统日志，并触发入侵检测系统的报警。网络扫描侧重于攻击者在较大范围内搜索、发现和确定攻击目标，而网络查点则是有针对性地收集攻击目标的详细信息。例如，用户账户、网络服务程序类型和版本号、错误或不当的系统配置等，这些看似并不重要的信息，一旦被恶意利用将会成为攻击者的利器。在得到了用户账户名后可以通过口令破解来获得对应的密码，在确定了网络服务程序类型和版本号后就可以进一步发现安全漏洞并准备渗透工具，错误或不当的配置为恶意程序上传提供了有效途径。

针对一些网络协议的工作原理，利用相关的工具远程探测并获取信息是网络查点常用的方法。HTTP、FTP、SMTP、POP、SNMP 等广泛使用的网络协议，在使用相关的工具时都会不同程度地暴露出一些有关服务器端的配置信息。例如，Telnet 是大部分操作系统和网络设备都支持的远程登录与管理工具，除此之外，还可以用来连接任意采用明文传输协议的网络服务。图 5-8 是利用 Windows 操作系统提供的 Telnet 工具尝试连接 www.sina.com.cn 80 时返回的信息，从中可以看到服务器类型“Server：nginxsServer：nginxs”，即 www.sina.com.cn 的 Web 网站使用的是 nginxs 服务器。

NetBIOS 是局域网中使用的一个非路由协议，其服务包括名字服务（UDP 137）、会话服务（TCP 139/445）和数据报服务（UDP 138）。在由 Windows 操作系统环境所组成的局域网中，BetBIOS 名字服务提供了计算机名与 IP 地址之间的名字解析（类似于互联网中的 DNS）。Windows 操作系统提供的 nbtstat 工具可以连接到指定的计算机远程查看其

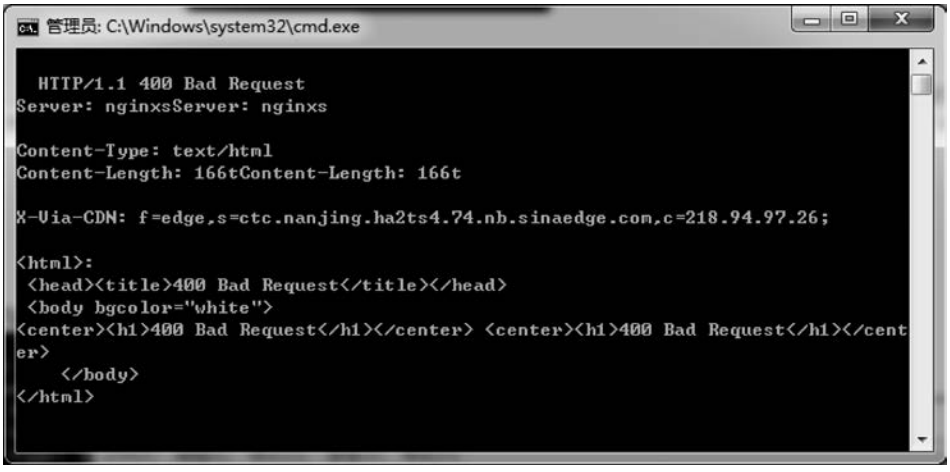

图 5-8 使用 Telnet 工具查看 www.sina.com.cn 80 的 Web 服务器

NetBIOS 名称表,包括计算机名、当前登录用户和运行的网络服务(针对 Windows Server 版本)、网卡的 MAC 地址等,如图 5-9 所示。

图 5-9 使用 nbtstat 工具查看 BetBIOS 远程计算机名称表

Windows 环境中的 SMB(Server Message Block,服务器信息块)是 NetBIOS 会话服务使用的用于进行文件与打印共享的协议。当用户在局域网中设置打印机或目录共享时要用到 SMB 协议,而早期 Windows 版本中的 SMB 协议在使用中很不安全,尤其是未经认证的用户也可以访问操作系统的默认隐蔽共享(包括逻辑盘符 C$、D$等,系统目录 winnt 或 windows 共享 admin$、进程间通信 IPC 共享 IPC$等),这些默认共享为网络查点提供了非常丰富的信息,所以 SMB 会话服务存在较大的安全隐患。

5.2.6 针对 Web 服务器信息收集的防范方法

针对 Web 服务器的网络踩点、网络扫描和网络查点等信息收集攻击,可通过以下方法进行必要的防范。

1. 针对网络踩点的防范方法

由于网络踩点攻击的目的是利用各类已有的工具和平台，从中收集与攻击目标相关的信息，然后对这些信息进行系统分析，从而确定攻击目标的网络位置、单位或个人信息等内容，为进一步实施网络攻击提供基础信息服务。因此，针对网络踩点的有效防范措施是加强单位或个人隐私的保护。将凡是不想让别人知道的信息都作为个人隐私进行有效保护，有些信息在单独出现时可能不会对隐私构成威胁，但是出现在不同时间和位置的多条信息在进行了有效关联后，有可能会将某人或单位的隐私暴露出来。

大数据技术的成熟和广泛应用，结合网络爬虫技术，使网络踩点攻击的实施越来越容易，而防御越来越困难。对于用户来说，应养成良好的上网习惯，不轻易在网上保留与个人或单位相关的信息，如果发现应尽快清除。随着网络实名制的推行及一些在线信息查询系统的使用，对于用户信息的保护提出了新的要求，从防范网络踩点攻击的角度，在不违反相关管理制定的前提下，应尽可能少地提供用户的真实信息。

2. 针对网络扫描的防范方法

针对网络扫描攻击中的主机扫描，可使用位于主机前端（如网关处）的网络入侵检测系统或运行在主机上的监测扫描软件来探测网络中发生的攻击行为，然后过滤掉正在发生的攻击报文，如许多个人防火墙软件都提供了对 ICMP、TCP、UDP 等报文的扫描和检测功能。目前，针对主机扫描的防范较为困难，因为大量的扫描报文可以绕过防火墙等安全设备，但通过网络安全产品制定严格的安全访问策略是防范主机扫描攻击的一种有效方法。以防火墙应用为例，网络管理员可以对进入网络的 ICMP 报文进行细致的评估，然后通过对防火墙上的 ACL（Access Control List，防火控制列表）的配置，决定哪些 ICMP 报文可以进入哪些网段，如只允许 ICMP Echo Reply（回送应答）、Echo Request Timed Out（请求超时）、Destination Host Unreachable（目标主机不可达）等响应报文进入防火墙的 DMZ（Demilitarized Zone，非军事区）网段并到达特定的主机，而其他的 ICMP 报文将被拦截。

针对 TCP 的主机和端口扫描，最有效的防范方法是在防火墙等安全设备或主机上关闭不需要的端口。系统管理员可以在网络中部署端口扫描监测工具来主动探测网络中正在发生的针对端口或系统服务的攻击行为，再有针对性地进行系统加固，防范针对主机端口和系统类型探查的攻击。

针对漏洞扫描攻击，最直接的防范方法是在安全漏洞和不安全配置被攻击者发现之前进行封堵。系统管理员可以通过漏洞扫描工具定期扫描网络中的主机，发现安全漏洞后及时安装补丁程序，如果发现不安全配置或安全弱点可及时进行调整。

3. 针对网络查点的防范方法

针对网络查点的攻击行为，可以通过以下方法进行防范。

（1）关闭不需要的网络服务。一些缺乏安全意识和经验的网络管理员在部署网络应用系统时多采用系统默认的安装和配置方法，一些不需要的网络服务会自动安装和启用，留下了安全隐患。为此，系统管理员应坚持系统服务的最少开放原则，在不影响应用功能的前提下，关闭不需要的网络服务，防止不必要的网络访问。

（2）加强网络服务的安全配置。在确定了系统中运行的网络服务后，需要对其安全性

进行配置。例如，针对 FTP 服务，应根据不同用户类型限制其访问方式，防止某些用户无限制地上传文件；禁用系统默认口令，避免使用弱口令；禁用一些很可能造成用户信息泄露的服务功能等。

(3) 使用安全性高的网络协议。在不影响应用功能的前提下，尽可能使用安全性较高的网络协议。例如，使用安全加密的 SSH(Secure Shell)协议来替代使用明文传输的 Telnet 协议实现对系统的远程配置，针对 FTP 服务可以在众多的软件中选用安全性较高的软件等。

5.3 Web 数据的攻防

Web 数据主要包括保存在后台数据库中的数据和 Web 客户端以表单形式提交的数据，这些数据存在着被窃取、篡改及非法注入的威胁。

5.3.1 针对敏感数据的攻防

针对敏感数据的攻击是近年来最常见的、最具影响力的一种攻击方式。在针对敏感数据的攻击中，攻击者不是直接攻击密码，而是在传输过程中或从客户端(如 Web 浏览器)窃取密钥、发起中间人攻击，或者从服务器端窃取明文数据。

1. 敏感数据的定义

个人信息是用来直接或间接识别自然人情况的数据资料，它被认为具有特殊风险，从而通常受到特殊保护。将个人信息划分为个人一般信息与敏感信息(特殊信息、特殊数据)，对个人敏感信息应坚持“使用限制原则”和“安全保护原则”。其中，使用限制原则是指个人数据不应该被披露和公开使用，除非在数据主体同意及法律授权情况下才能公开；安全保护原则是指个人资料应采取合理的安全保护措施，以防止资料的丢失、非法接触、毁损、利用、修改和泄露等危险的发生。

在 Web 应用环境中，由于系统配置上的不当，或者使用者缺乏必要的安全意识，导致敏感信息出现在 Web 站点上。通常情况下，这些信息包括大量的个人敏感信息(Personal Identifiable Information，PII)，如个人的身份证号码、联系方式、银行信用卡账号、医疗记录等。另外，还包括员工通讯录、重要会议记录、重要技术资料等涉及单位的敏感信息，甚至涉及国家或企业秘密的信息等。

针对敏感数据，最常见的漏洞是不对敏感数据进行加密处理。在数据加密过程中，最常见的问题是不安全的密钥生成和管理，以及使用弱加密算法、弱协议和弱密码。特别是使用弱的 Hash 函数来保护密码。例如，一个应用程序使用自动加密系统加密信用卡信息，并存储在数据库中。但是，当数据被检索时会被自动解密，这就使得 SQL 注入漏洞能够以明文形式获得所有信用卡的信息。又如，密码数据库使用未加入盐值(salt)的 Hash 算法或弱 Hash 算法来存储每个用户的密码。一个文件上传漏洞使攻击者能够获取密码文件，所以这些未加入盐值(salt)的 Hash 密码就可以通过彩虹表来暴力破解。

2. 敏感数据的保护方式

对于互联网应用中产生和使用的海量数据，首先需要确认哪些属于敏感数据应进行加

密保护。对于已经确定的敏感数据，还需要针对不同的保护要求和应用环境来确认使用的保护方式，主要包括以下几点。

（1）在数据传输过程中是否使用明文传输，这与选择所使用的传输协议相关，如HTTP、SMTP、FTP等。

（2）当数据被长期存储时，无论存储在哪里，它们是否都被加密，是否包含备份数据。

（3）无论默认条件还是在源代码中，是否还在使用一些陈旧的或脆弱的加密算法。

（4）是否使用默认加密密钥，是否生成或重复使用脆弱的加密密钥，是否缺少恰当的密钥管理或密钥回转。

（5）是否强制加密敏感数据，如用户代理（如Web浏览器）指令和传输协议是否被加密。

（6）用户代理（如应用程序、邮件客户端等）是否未验证服务器端证书的有效性。

在确定了敏感数据的保护要求后，可以采取以下方式进行保护。

（1）对系统处理、存储或传输的数据分类，根据分类进行访问控制。

（2）熟悉与敏感数据保护相关的法律和条例，并根据每项法规要求保护敏感数据。

（3）对于不需要保存的重要敏感数据，应当尽快清除。

（4）确保存储的所有敏感数据被加密。

（5）确保使用了最新的、强大的加密算法或密码、参数、协议和密匙，并且能够安全规范地对密钥进行管理。

（6）确保传输过程中的数据被加密，如采用TLS/SSL。确保数据加密被强制执行，如使用HTTP严格安全传输（HTTP Strict Transport Security，HSTS）协议。

（7）禁止缓存包含敏感数据的响应。

3. Web服务器端敏感数据的泄露和防御方式

在Web应用环境中，敏感信息一般可以通过目录遍历和错误的配置等途径或方式被泄露，可以有针对性地采取相应的防御方式。

（1）目录遍历。由于发布Web网站的Web服务器配置不当，可能导致存放Web网站信息的目录存在遍历漏洞。目录遍历（路径遍历）是由于Web服务器或者Web应用程序对用户输入的文件名称的安全性验证不足而导致的一种安全漏洞，使得攻击者通过利用一些特殊字符就可以绕过服务器的安全限制，访问任意的文件（可以是Web根目录以外的文件），甚至执行系统命令。

目录遍历漏洞被利用的原理是程序在实现上没有充分过滤用户输入的类似于“../”的目录跳转符，导致恶意用户可以通过提交目录跳转来遍历服务器上的任意文件。例如，正常读取文件的URL为“http://www.abc.com/text.jsp?file=xyz.html”，而恶意URL输入可以设置为“http://www.abc.com/text.jsp? file=../../Windows.system.ini”。

对上面遍历攻击的防范方法有以下几种。

① 对用户的输入进行验证，特别是路径替代字符“../”。

② 尽可能采用白名单的形式来验证所有的用户输入。

③ 合理配置Web服务器的目录权限。

④ 程序出错时，不要显示内部相关细节。

（2）上传目录配置错误。Web网站一般都需要通过FTP方式远程上传数据或更新程

序，由于配置上的不当，攻击者可以查看、修改或删除存放 Web 网站所在目录中的文件。攻击者也可以在发起攻击之前远程探测 FTP 服务器软件的类型和版本号，为进一步确定安全漏洞并实施渗透攻击收集基础信息。例如，当用户连接管理 Web 网站目录的 FTP 服务器时，出现了如下提示信息。

```
ftp172.16.32.105
Connected to 172.16.32.105
220 - Serv - U FTP Server v15.1.6 for WinSock ready...
220 S TEAM
```

从这条连接信息中，攻击就可以清楚地知道 FTP 服务器使用的是 Serv-U v15.1.6 软件。

在使用 FTP 服务来远程管理 Web 网站目录时，出于安全考虑，需要注意以下几个方面的问题。

① 未经授权的用户禁止在 Web 服务器上进行任何 FTP 操作，包括查看 Web 网站目录下的信息。

② FTP 用户只允许访问经系统管理员授权的目录和文件。

③ 未经允许，FTP 用户不能在服务器上创建文件或目录。

④ 提供翔实的 FTP 用户访问日志信息。

为了防止部分 Web 网站管理人员没有对安全问题引起足够的重视，可以从系统配置上强化安全管理。以 Linux 中的 FTP 服务为例，可以从以下几个方面强化安全设置。

① 除匿名用户之外的其他所有 FTP 用户账户必须添加在“/etc/passwd”文件中，并且口令不能为空。在没有正确输入用户名和口令的情况下，服务器拒绝访问。

② “/etc/FTPusers”是 FTP 守护进程 FTPd 使用的一个文件，如果将某一用户的账号添加在该文件中，在使用对应的账号登录时，FTP 服务器将拒绝。该功能相当于建立用户访问的黑名单，可以将存在安全风险的登录账号添加在该文件进行管理。

③ 当以“anonymous”或“FTP”作为用户名、以用户的互联网电子邮件地址作为保密字进行匿名登录时，FTP 服务器的“/etc/passwd”文件中需要添加“FTP”用户账号；否则 FTP 服务器不接受匿名 FTP 连接。出于安全考虑，可以禁用匿名 FTP 登录。

(3) 泄露敏感数据。不管是出于技术或非技术原因，由于缺乏安全意识，在 Web 网站上可能会出现有关个人隐私、企业商业秘密，甚至是国家机密。对于可能出现在 Web 网站上的敏感信息，可以采用技术手段在发布之前进行过滤，也可以通过安全管理制度对需要在 Web 网站上发布的信息进行审核。尤其是对于提供留言、回复或发帖等功能的 Web 网站，一定要通过技术手段进行不良信息的过滤，并通过审核制度严把不良、不实甚至是违法信息的传播，更要从技术上防止 Web 网站被攻击者控制后上传一些不良或非法信息。

5.3.2 网站篡改

网站篡改(Website Defacement)是一类出现较早且经常发生的网络攻击形式，是一类不以谋求经济利益为目的网络攻击行为。

网站篡改是指攻击者在成功入侵网站发布服务器并控制了对 Web 网站目录的操作权限后，用预先编写好的带有攻击意图的页面替换掉原网站页面（一般为主页面），从而实现攻击者预谋的一种恶意攻击行为。

近年来，随着信息化进程的快速推进，政府机关、学校、企业等单位几乎都在互联网上创建了自己的门户网站，用于宣传或重要信息的发布。单位的门户网站是向用户提供各类信息的重要渠道和窗口，也成为黑客入侵的主要目标之一。网站存在的安全漏洞很容易被不法分子利用，进行页面篡改以达到传播反动、淫秽色情等内容的目的，影响用户正常获取信息的方式，并对网站拥有者造成很大的负面影响。

网站主页被直接篡改是最常见的一种针对网络篡改的攻击方式。网站主页被直接篡改，需要利用网站存在的漏洞侵入网站空间，然后对网站主页文件进行修改或替换。近年来，一些重要网站被入侵的事件频发，网站主页被篡改成赌博、色情和政治敏感内容等，不但损害了网站拥有者的形象，而且给用户使用带来了极大的不便，甚至是失去了用户的信任。例如，2013 年 5 月 12 日，一个只有 16 岁的江西高中生在入侵了兰州大学宣传部网站后，在主页上挂了十几条以“尊敬的兰大领导老师你们好”为标题的短文，强调向往兰州大学的信息安全专业，恳求学校给自己一个学习机会，为中国网络贡献自己的力量。又如，2013 年 5 月 24 日，有网友在微博称在埃及卢克索神庙的浮雕上看到“……到此一游”几个字，并称“我们试图用纸巾擦掉这羞耻，但很难擦干净，这是三千五百年前的文物呀”。因为此事，该生所在学校的网站于当月 26 日被黑。打开该学校网站后，最先显示的是“……到此一游”的弹出窗口，单击“确定”按钮后才能显示正常内容。

诸如此类安全事件，近年来发生的很多。由此说明，即使是学校和政府机关等重要网站的安全形势也非常严峻，网站安全建设亟待加强。网站被篡改一般可以分为以下 3 种类型。

① 来自境外黑客的以各种政治和宗教为目的的篡改攻击，主要针对对象是政府和知名高校的网站。攻击成功后，黑客会在被攻下的网站上留下一些宣传不同宗教信仰和政治立场的文字和图片。

② 攻击者以炫耀技术为目的，攻击成功后一般会留下一些调侃的文字或图片。

③ 以经济利益为目的，主要针对的是政府和知名高校网站，其目的是通过加入黑链以获得较高的搜索引擎权重。

针对网站被直接篡改的安全问题，只能加强对网站空间的安全管理，包括对网站服务器、网站远程管理账号、网站开发程序的安全管理等。

5.4 Web 应用程序的攻防

一个 Web 应用程序是由完成特定任务的各种 Web 组件（web components）构成的，在实际应用中，Web 应用程序是由多个 Servlet（server applet）、JSP 页面、HTML 文件及图像文件等组成的，所有这些组件相互协调为用户提供一组完整的服务。

5.4.1 Web 应用程序安全威胁

由于开发一个 Web 应用程序时，需要涉及需求分析、设计、选择架构、开发和发布等环

节，每一个环节考虑不周，都可能留下安全隐患。例如，在选择开发架构时，可以在 ASP.NET、PHP、Python、Ruby on Rails 等方案中选择，每一种架构不存在绝对的安全或不安全，关键是在满足功能需求的前提下，开发人员要能够熟悉其安全上的薄弱环节并进行加固。

Web 应用程序是目前 Web 服务中安全性最为脆弱的一个组成部分。相比于底层的操作系统、主流应用软件和网络服务，由于 Web 应用程序的开发门槛相对较低，大量 Web 网站的编码质量相对不高，而且在正式上线发布之前并未进行安全性测试，这就导致许多 Web 应用程序中存在不同程度的安全隐患。同时，Web 应用程序的复杂性和实现方式上的多样性也是导致安全问题频发的关键因素。

针对复杂的 Web 应用程序安全问题，由安全专家、行业顾问和诸多组织的代表组成的国际团体 WASC(Web Application Security Consortium，Web 应用安全联盟)于 2010 年发布的 *WASC Threat Classification 2.0*(WASC 威胁分类 2.0)报告中，将 Web 应用所受到的威胁、攻击进行了说明，并归纳成具有共同特征的分类，如表 5-1 所示。

表 5-1 WASC 团队报告的安全威胁分类

攻 击 类 型	脆弱性说明
功能滥用	利用 Web 站点自身的特性和功能来使用、蒙骗或阻挠访问控制机制的一种攻击方法
暴力攻击	使用穷举测试来猜测个人的用户名、密码、信用卡账号或密钥的过程
缓冲区溢出	通过覆盖部分内存来改变应用程序流的攻击方式
内容电子欺骗	用于骗取用户相信 Web 站点上出现的某些内容合法且不是来自外部源的一种攻击方法
凭证/会话预测	是一种操纵或假冒 Web 站点用户的方法。推断或猜测识别特定会话或用户的唯一值，以完成攻击行为
跨站脚本	跨站脚本(XSS)是强制 Web 站点回传攻击者提供的可执行代码(加载在用户浏览器中)的一种攻击方法。受跨站脚本影响的用户账户可能会受操纵(cookie 盗用)，其浏览器可能会重定向到其他位置，或者可能显示用户正在访问的 Web 站点所提供的欺骗性内容
拒绝服务	拒绝服务(DoS)是旨在阻止 Web 站点为正常用户活动提供服务的一种攻击方法
目录索引	自动目录列表/索引是一项服务器功能，在没有普通基本文件(index.html/home.html/default.htm)的情况下，该功能会列出所请求目录中的所有文件
格式化字符串攻击	会使用字符串格式化库功能来访问其他内存空间，以改变应用程序流
信息泄露	指 Web 站点显示可能会协助攻击者攻击系统的敏感数据(如开发者注释或错误消息)
抗自动化能力不足	指 Web 站点准许攻击者将应当仅手动执行的过程自动化
认证不充分	指 Web 站点准许攻击者访问敏感内容或功能而未对其访问许可权进行适当认证
权限不足	指 Web 站点准许访问应当需要提高访问控制限制的敏感内容或功能
不充分处理验证	指 Web 站点准许攻击者绕过或回避计划的应用程序流量控制
会话有效期不足	指 Web 站点准许攻击者复用旧会话凭证或会话标识以进行授权。会话有效期不足将增加 Web 站点受攻击(窃取或假冒其他用户)的可能性
LDAP 注入	是一种攻击方法，它通过用户提供的输入来构造 LDAP(Lightweight Directory Access Protocol，轻量级目录访问协议)控制语句，进而攻击 Web 站点

续表

攻击类型	脆弱性说明
操作系统命令	是一种攻击方法，它通过操纵应用程序输入来执行操作系统命令，进而攻击 Web 站点
路径遍历	路径遍历攻击方法会强制访问可能位于文档根目录外的文件、目录和命令
可预测资源位置	是用于显示隐藏 Web 站点内容和功能的一种攻击方法。该攻击通过强行搜索，以查找不打算供公共查看的内容。临时文件、备份文件、配置文件和样本文件这些示例都是潜在的剩余文件
会话固定	该攻击方法会强制赋予用户的会话标识一个确定值
SQL 注入	是一种攻击方法，它通过用户提供的输入来构造 SQL(Structured Query Language，结构化查询语言)语句，进而攻击 Web 站点
SSI 注入	SSI(Server Side Includes，服务器端包含)注入是一种服务器端攻击方法，该方法允许攻击者将代码发送到随后将由 Web 服务器在本地执行的应用程序
弱密码恢复验证	指 Web 站点准许攻击者非法获取、更改或恢复其他用户的密码
XPath 注入	是一种攻击方法，它通过用户提供的输入来构造 XPath 查询，进而攻击 Web 站点

另一个 Web 应用安全领域知名的研究团队 OWASP(Open Web Application Security Project，开放 Web 应用程序安全项目)在对普遍和流行的安全隐患进行长期跟踪研究中，于 2017 年公布了 OWASP Top 10 2017，其中有 10 项最严重的 Web 应用程序安全风险，如表 5-2 所示(排在表格最前面的安全风险等级最高)。

表 5-2　OWASP Top 10 2017 中确定的 10 项最严重的 Web 应用程序安全风险

安全风险	说　明
注入	将不受信任的数据作为命令或查询的一部分发送到解析器时，会产生如 SQL 注入、NoSQL 注入、OS 注入和 LDAP 注入的注入缺陷。攻击者的恶意数据可以诱使解析器在没有适当授权的情况下执行非预期命令或访问数据
失效的身份认证	通常通过错误使用应用程序的身份认证和会话管理功能，攻击者能够破译密码、密钥或会话令牌，或者利用其他开发缺陷来冒充其他用户的身份
敏感信息泄露	攻击者可以通过窃取或修改未加密的数据来实施信用卡诈骗、身份盗窃或其他犯罪行为。未加密的敏感数据容易受到破坏，因此需要对敏感数据加密，这些数据包括传输过程中的数据、存储的数据及浏览器的交互数据
XML 外部实体(XXE)	攻击者可以利用外部实体窃取使用 URI 文件处理器的内部文件和共享文件、监听内部扫描端口、执行远程代码和实施拒绝服务攻击
失效的访问控制	攻击者可以利用安全缺陷，使用未经授权的应用功能或访问未经授权的数据，如访问其他用户的账户、查看敏感文件、修改其他用户的数据、更改访问权限等
安全配置错误	安全配置错误是最常见的安全问题，这通常是由于不安全的默认配置、不完整的临时配置、开源云存储、错误的 HTTP 头部配置及包含敏感信息的详细错误信息所造成的。因此，不仅需要对所有的操作系统、框架、库和应用程序进行安全配置，而且必须及时修补存在的安全漏洞，并及时升级程序
跨站脚本(XSS)	当应用程序的新网页中包含不受信任的、未经恰当验证或转义的数据时，或者使用可以创建 HTML 或 JavaScript 的浏览器 API 更新现有的网页时，就会出现 XSS 缺陷。XSS 让攻击者能够在受害者的浏览器中执行脚本，并劫持用户会话、破坏网站或将用户重定向到恶意站点

续表

安全风险	说明
不安全的反序列化	把对象转换为字节序列的过程称为对象的序列化，把字节序列恢复为对象的过程称为对象的反序列化。不安全的反序列化会导致远程代码执行。即使反序列化缺陷不会导致远程代码执行，攻击者也可以利用它们来执行攻击，包括重播攻击、注入攻击和特权升级攻击
使用含有已知漏洞的组件	组件(如库、框架和其他软件模块)拥有和应用程序相同的权限。如果应用程序中含有已知漏洞的组件被攻击者利用，可能会造成严重的数据丢失或服务器接管。同时，使用含有已知漏洞的组件的应用程序和 API 可能会破坏应用程序防御，造成各种攻击并产生严重影响
不足的日志记录和监控	不足的日志记录和监控，以及事件响应缺失或无效的集成，使攻击者能够进一步攻击系统、保持持续性或转向更多系统，以及篡改、提取或销毁数据

5.4.2 SQL 注入漏洞


视频讲解

动态网页技术在丰富了 Web 页面表现形式和应用功能的同时，因其后台数据库在技术自身和具体应用中存在的一些不足，为 Web 网站的安全带来了一些隐患。SQL 注入(SQL Injection)便是近年来最受关注的一类针对 Web 站点的网络攻击类型。

1. 注入攻击的概念

几乎任何数据源都能成为注入载体，这些数据源包括环境变量、所有类型的用户和参数、外部和内部 Web 服务等。当攻击者向解释器发送恶意数据时，注入漏洞产生。目前，注入漏洞的存在非常广泛，通常存在于 SQL、LDAP、XPath(或 NoSQL)查询语句、OS 命令、XML 解析器、SMTP 报文头部、表达式语句和 ORM(Object Relational Mapping，对象关系映射)查询语句中，所以常见的注入有 SQL 注入、OS 命令注入、ORM 注入、LDAP 注入、EL(Expression Language，表达式语言)注入、OGNL(Object Graphic Navigation Language，对象图导航语言)注入。注入攻击轻则导致数据丢失、破坏或泄露给无授权方，重则导致主机被完全接管。注入主要由以下原因产生。

(1) 用户提供的数据没有经过应用程序的验证、过滤或净化。

(2) 在没有上下文感知转义的情况下，动态查询语句或非参数化的调用被用于解释器。

(3) 在 ORM 搜索参数中使用了恶意数据，这样搜索将获得包含敏感或未授权的数据。

(4) 恶意数据直接被使用或连接，如 SQL 语句或命令在动态查询语句、命令或存储过程中包含结构和恶意数据。

在众多的注入攻击中，SQL 注入是目前最常见、影响范围最广的一种攻击方式。为此，下面重点以 SQL 注入攻击为例，介绍其实现原理、方法、过程及相应的防范方法。

2. SQL 注入的概念

SQL(Structured Query Language，结构化查询语言)是一种最常使用的用于访问 Web 数据以及进行 Web 数据查询、更新和管理的数据库查询和程序设计语言，是实现 Web 客户端与数据库服务器信息交互的重要工具。

SQL 注入攻击需要具备两个前提条件：从软件系统自身来看，被攻击系统能够以“字符

串”方式接收用户输入，可以利用输入“字符串”构造的 SQL 语句来执行数据库操作；从软件开发人员来看，对于用户输入的“字符串”，虽然符合 SQL 语句的语法要求，但对其可能的执行结果未进行严格验证。当以上两个条件同时具备时，攻击者便将恶意代码注入“字符串”后，使其得以在数据库系统上执行，实现攻击目的。

SQL 注入是利用 Web 应用程序数据层存在的输入验证漏洞，将 SQL 代码插入或添加到应用程序（或用户）的输入参数中，再将这些参数传递给后台的 SQL 服务器加以解析并执行的攻击方式。如果 Web 应用程序未对动态构造的 SQL 语句所使用的参数进行正确性审查，那么攻击者很可能会修改后台 SQL 语句的构造。如果攻击者能够修改 SQL 语句，那么该语句将与该应用程序的使用者拥有相同的运行权限。

SQL 是访问 MS SQL、Oracle、MySQL 等数据库服务器的标准语言，大多数 Web 应用程序都需要与数据库进行交互，并且大多数 Web 应用程序的编程语言（如 ASP、C＃、.net、Java、PHP 等）均提供了通过编程来连接数据库并进行交互的功能。如果 Web 开发人员无法确保在通过 Web 表单、Cookie 及输入参数等方式中接收到数据并传递给 SQL 查询（该查询在数据库上执行）之前对其进行验证，那么通常会出现 SQL 注入漏洞。如果攻击者能够控制发送给 SQL 查询的输入，并且能够操纵该输入将其解析为代码而非数据，那么攻击者就很有可能在后台数据库执行该代码。

SQL 注入存在的危害主要表现在数据库信息泄露、网页篡改、网站挂马、数据库被恶意操作、服务器被远程控制等几方面。

3. SQL 注入攻击的原理

SQL 注入攻击主要是通过构建特殊的输入（这些输入往往是 SQL 语法中的一些组合）作为参数传递给 Web 应用程序，通过执行 SQL 语句而执行攻击者预期的操作。下面以一个动态 ASP 的 Web 应用系统的登录认证模块为例，具体介绍 SQL 注入攻击的实现原理。

约定：需要进行认证登录的 Web 网站的用户名和对应的登录密码保存在 MS SQL Server 数据库 UserDB 的 users 表中，对应的字段名分别为 id、username 和 password，典型的 SQL 查询语句如下。

```
SELECT * from users WHERE usename = 'wangqun' AND password = 'wq123456'
```

当该 SQL 语句提交给后台数据库执行时，如果用户“wangqun”和登录密码“wq123456”已保存在数据库 UserDB 的 users 表中，则将成功登录该 Web 系统。

但是，如果为 username 和 password 分别赋值为“' wangqun ' OR ' 1 ' = ' 1 '”和“'abc123456'OR '1'='1'”，那么将会构造一个 SQL 查询语句。

```
SELECT * from users WHERE usename = 'wangqun' OR '1' = '1' AND password = 'abc123456' OR '1 = 1'
```

由于在 SQL 中关系型运算符优先级从高到低为 NOT > AND > OR，因此上述语句等价于：

```
SELECT * from users WHERE usename = 'wangqun' OR ('1' = '1' AND password = 'abc123456') OR '1 = 1'
```

该 SQL 查询语句可以分为 3 个判断，只要有一个条件成立，就会成功执行。由于“1＝1”在逻辑上是永恒成立的，因此无论 users 表中的 username 和 password 字段是何内容，攻击者都

可以在不需要知道数据库中真实的用户名和登录密码的前提下,通过 SQL 注入攻击成功登录。

如果攻击者为 username 字段赋值“x' OR '1'='1'”,为 password 字段赋值“'wq123456';DROP TABLE users;SELECT * from admin WHERE 't'='t'”,将会构造如下的 SQL 查询语句。

```
SELECT * from users WHERE usename = 'x' OR '1' = '1' AND password = 'wq123456';
DROP TABLE users;
SELECT * from admin WHERE't' = 't'
```

当上述 SQL 查询语句提交给后台数据库时,将顺序执行 3 条不同的 SQL 操作:第 1 条是对 users 表进行查询操作;第 2 条是删除 users 表;第 3 条是查询 admin 表中的全部记录。

SQL 注入攻击过程中有关 SQL 语句的构造方法较多,在此不再一一介绍。

4. SQL 注入攻击的实现过程

SQL 注入可以出现在任何系统或用户接收数据输入的前端应用程序中。在 Web 应用环境中,Web 浏览器为前端应用程序,它负责向用户请求数据并将数据发送到远程服务器端。远程服务器使用提交的数据创建 SQL 查询。可以通过识别服务器响应中的异常来确定是否存在 SQL 注入漏洞。

(1) 寻找 SQL 注入点。在互联网上寻找和确定 SQL 注入点是进行 SQL 注入攻击的前提。最常见的 SQL 注入点的判断方法是在动态网页中寻找如下形式的链接。

```
http://Website/**.asp?xx=abc
http://Website/**.php?xx=abc
http://Website/**.jsp?xx=abc
http://Website/**.aspx?xx=abc
```

下面以 http://Website/**.asp? xx=abc 为例进行介绍。在实际应用中,参数 xx 字段的类型可能是整数型或字符串型。

当参数 xx 字段的类型为整数型时,SQL 查询语句的形式为

```
SELECT * from users WHERE xx = abc
```

这时,攻击者可以将“abc”设置为如下 3 种不同类型的字符串,并通过返回的页面信息来确定该动态网页是否存在 SQL 注入点。

① 当原来的整数型输入“abc”修改为“abc'”时,由于输入后的数据类型不符合字段类型要求,将会导致 SQL 语句错误,并返回错误提示信息。

② 当原来的整数型输入“abc”修改为“abc and 1=1”时,由于“1=1”永恒成立,不对查询条件造成任何影响,因此动态网页将返回正常页面。

③ 当原来的整数型输入“abc”修改为“abc and 1=2”时,由于“1=2”永恒不成立,将查询不到任何信息,因此将会返回一个空白或错误提示页面。

当同时满足以上 3 个条件时,可以认定该 Web 应用程序存在 SQL 注入点。

当参数 xx 字段的类型为字符串参数时,SQL 查询语句的形式为

```
SELECT * from users WHERE xx = 'abc'
```

这时，攻击者可以将参数取值“abc”设置为如下 3 种不同的字符串，并通过返回的页面信息来确定该动态网页是否存在 SQL 注入点。

① 当原来的字符串参数输入“abc”修改为“abc'”时，由于输入后的“'”（单引号）不符合字段类型要求，将会导致 SQL 语句错误，并返回错误提示信息。

② 当原来的字符串参数输入“abc”修改为“abc' and '1'='1'”时，由于“'1'='1'”永恒成立，不对查询条件造成任何影响，因此动态网页将返回正常页面。

③ 当原来的字符串参数输入“abc”修改为“abc ' and '1'='2'”时，由于“'1'='2'”永恒不成立，将查询不到任何信息，因此将会返回一个空白或错误提示页面。

当同时满足以上 3 个条件时，可以认定该 Web 应用程序存在 SQL 注入点。

（2）探测后台数据库的类型。不同 SQL 数据库软件在操作方法上存在着差异，只有了解了具体的数据库软件类型后才能有针对性地进行远程操控。一般情况下，后台数据库类型与所使用的开发语言有关，例如，ASP 和. NET 一般使用 MS SQL Server 数据库，PHP 使用 MySQL 和 PostgreSQL 数据库，而 Java 使用 Oracle 和 MySQL 数据库等。也就是说，如果知道某一 Web 应用程序是使用什么语言开发的，就可以大体确定后台数据库的类型。

另外，还可以借助数据库的一些特征来探测其类型，最常见的是根据数据库服务器的系统表进行判断。例如，Access 的系统表为 msysobjects，在 Web 环境下没有访问权限；而 MS SQL Server 的系统表是 sysobjects，且在 Web 环境下有访问权限。为此，可以输入类似以下两条语句。

```
http://Website/**.asp?xx=abc and (select count(*) from sysobjects)>0
http://Website/**.asp?xx=abc and (select count(*) from msysobjects)>0
```

当第 1 条请求 URL 运行正常，而第 2 条不正常时，可以确定后台数据库为 MS SQL Server；当两条都不正常时，可以确定后台数据库为 Access。

（3）获取管理员账户信息。在绝大多数情况下，Web 网站的发布和日常维护管理都需要以远程方式进行，管理员通过在远程登录管理界面正确输入用户账户信息后，才能对 Web 应用程序进行上传/下载文件、修改配置、浏览目录等操作。在此过程中，如何获取管理员账户信息是非常关键的。

一般情况下，Web 应用程序管理员账户信息保存在后台数据库中。如果能够通过 SQL 注入攻击获取到管理员账户，就可以实现对 Web 网站的远程控制。然后，就可以根据攻击需要，上传后门程序（如 ASP 后门），对 Web 服务器软件甚至是 Web 服务器操作系统进行操控，实施攻击行为。

5. SQL 注入攻击的防范方法

由于大多数 SQL 注入攻击都是利用 Web 应用程序中对用户输入内容没有进行严格的转义字符过滤和类型查询这一漏洞而实施，因此对 SQL 注入的防范方法主要针对用户输入内容中特殊字符及参数类型与长度的严格检查。在具体的防范中，除部署专业的 Web 应用防火墙外，在代码层可以采取以下的方法进行防范。

（1）使用参数化语句。在 Web 应用程序中利用用户输入参数来构造动态 SQL 语句

时，要注意参数类型的安全性，使用能够确保类型安全的参数化语句。在 ADO、ADO.NET 等数据库访问 API 时，可以明确输入参数的具体类型（如字符串、整数型、日期等），以保证用户输入内容符合该类型的格式，并被正确地进行转义和编码，进而避免 SQL 注入攻击的发生。

（2）输入验证。输入验证是指验证用户输入的内容，确保其符合 Web 应用程序中已确定的标准，具体可分为白名单验证和黑名单验证两种方式。

① 白名单验证。白名单验证是指 Web 应用程序只接收记录中可信的输入内容。它在接收输入并做进一步处理之前，需要验证输入是否符合期望的类型、长度（或大小）、数据范围等标准或格式。例如，要验证输入值是用户身份证号码时，需要验证的输入内容可包含字符和总长度（一般为 18 位）。

② 黑名单验证。黑名单验证是指 Web 应用程序会自动阻止已经确认并保存在记录中的恶意内容输入。在黑名单验证过程中，由于潜在的恶意内容列表较大，检索效率较低，而且黑名单的维护和更新较为困难，因此使用效果不如白名单验证好。

（3）实施最小权限原则。一旦攻击者获得执行 SQL 查询的能力，就会以一个数据库用户的身份进行查询。所以，可以为每一个数据库用户设置只能拥有完成自己的任务所必需的权限，而限制拥有超出自己操作范围的权限，以此通过实施最小权限来防止 SQL 注入攻击。如果一个数据库用户拥有很大的权限，攻击者在获取了该用户的权限后就可能删除数据表，操纵其他用户的权限，从而发起其他 SQL 注入攻击。为此，绝对不能以超级用户、其他权限较高或管理员级的用户身份访问网络应用程序的数据库，从而杜绝这种情况发生。最小权限原则的另外一个变体是区别数据库的读数据和写数据权限。具体可以设置一个拥有写数据权限的用户和另一个只有读数据权限的用户，这种角色区分可以确保在 SQL 注入攻击目标为只读用户时，攻击者无法写数据或操纵表数据。

（4）使用存储过程。存储过程（Stored Procedure）是数据库中的一个重要对象，它是存储在数据库中的一组为了完成特定功能的 SQL 语句集，经过第一次编译后再次调用时不需要再次编译，用户通过指定存储过程的名称并给出参数（如果该存储过程带有参数）来执行它。将 Web 应用程序设计成专门使用存储过程来访问数据库是一种可以防止或减轻 SQL 注入攻击的技术。因为在大多数数据库中使用存储过程时都可以在数据库层配置访问控制，这就意味着如果发现了可利用的 SQL 注入攻击，则会通过正确配置许可来保证攻击者无法访问数据库中的敏感信息。

（5）加强对 SQL 数据库服务器的安全配置。可以采用必要的方法加强对 SQL 数据库服务器的安全配置，尤其是加强 Web 应用程序与数据库之间的安全连接，以最小权限原则配置 Web 应用程序连接数据库的查询操作权限，避免敏感数据（如用户账户信息）以明文形式存储在数据库中。另外，通过设置不泄露任何有价值信息的默认出错机制（如 Web 查询结果出错、用户认证失败等）并以此来替代默认出错提示，避免为攻击者提供有用信息。

5.4.3 跨站脚本漏洞


视频讲解

跨站脚本（Cross Site Scripting，XSS）漏洞是一种经常出现在 Web 应用程序中的安全漏洞，是由于 Web 应用程序对用户的输入内容的安全验证与过滤不够严格而产生的。XSS 漏洞的最大特点是能够注入 HTML 和 JavaScript 代码到用户浏览器的网页上，从而达到劫

持用户会话的目的。

1. 跨站漏洞及跨站脚本攻击

跨站漏洞的产生是由于网站开发人员在编写网站程序时对一些变量没有做充分的过滤，直接把用户提交的数据送到 SQL 语句里执行，导致用户可以提交一些特意构造的语句，如 JavaScript、VBScript 和 ActionScript 等脚本代码。在此基础上，攻击者利用跨站漏洞输入恶意的脚本代码，当恶意代码被执行后就产生了跨站脚本（Cross Site Scripting，XSS）攻击。

跨站脚本攻击是指攻击者通过向 Web 页面中插入恶意的脚本代码，当用户打开该页面时，嵌入其中的恶意代码就会被执行，从而达到恶意攻击的目的。为了提高用户的体验、丰富网站的功能，现在许多网站采取动态网页技术，即根据用户提供的数据（通过数据库或手工输入方式），Web 应用程序会动态地显示输入内容。动态网站技术为实现 XSS 攻击提供了便利，当攻击者输入隐藏了恶意目的的代码时，攻击就会发生。攻击者会在网站页面文件中植入恶意代码，当用户打开该网页时，这些恶意代码就会注入客户端的浏览器中并执行，使用户受到攻击。

在 XSS 攻击中，攻击者利用跨站漏洞可以在网站中插入任意代码，这些代码的功能包括获取网站管理员或普通用户的 Cookie、隐蔽运行网页木马、格式化浏览者的硬盘等，只要脚本代码能够实现的功能，跨站攻击都能够实现，如窃取用户隐私、钓鱼欺骗、偷取密码、传播恶意代码等。

与前文介绍的 SQL 注入攻击不同的是：XSS 攻击的最终目标不是提供服务的 Web 应用程序，而是使用 Web 应用程序的用户。随着 Web 技术的快速发展，大量的主流浏览器及其插件普遍支持对 JavaScript、Flash Action Script、Silverlight 等客户端脚本代码的本地执行，为 XSS 攻击提供了所需要的环境。攻击者可以利用 Web 应用程序中的安全漏洞，在 Web 服务器网页中插入经过精心构造的客户端脚本代码，形成恶意攻击页面。当 Web 客户端访问这些网页时，所使用的浏览器就会自动下载并执行这些网页中的恶意客户端脚本，对其进行解析和执行，从而遭受攻击。

需要说明的是，层叠样式表（Cascading Style Sheets，CSS）网页开发技术出现于 1994 年，而在 1996 年 XSS（Cross Site Scripting，跨站脚本）出现后，为了便于区别，则将其英文缩写确定为 XSS。

2. XSS 攻击的分类

根据攻击特征和对安全漏洞利用方法的不同，可以将 XSS 攻击分为反射式 XSS 攻击、存储式 XSS 攻击和基于 DOM 的 XSS 攻击 3 种类型。

（1）反射式 XSS 攻击。反射式 XSS 攻击也称为非持久性 XSS 攻击或参数型 XSS 攻击，是一种最常见的 XSS 攻击类型，主要用于将恶意脚本附加到 URL 地址的参数中。例如：

```
http://WebSite/home.php?id=<script>alert(/xss/)</script>
```

在 Web 交互操作中，当 Web 浏览器在 HTTP 请求参数或 HTML 表单中接收到信息时，则由服务器端脚本为该用户产生一个结果页面。在此过程中，由于服务器端脚本缺乏对

请求数据的安全验证与过滤机制，就会因存在的 XSS 漏洞遭受到攻击。

反射式 XSS 攻击的实现过程中一般为：攻击者发现存在 XSS 安全漏洞网页(URL)后，根据输出点的环境构造 XSS 攻击代码并进行编码，然后通过特定手段(如发送电子邮件)发送给受害者，诱使受害者去访问一个包含恶意代码的 URL，当受害者单击这个经过专门设计的 URL 链接后，攻击代码会直接在受害者的浏览器上解析并执行。

需要说明的是，反射式 XSS 攻击一般需要欺骗用户自己去单击链接才能触发 XSS 代码(服务器中没有这样的页面和内容)，一般容易出现在搜索页面中。

(2) 存储式 XSS 攻击。存储式 XSS 漏洞是危害最为严重的 XSS 漏洞，它通常出现在一些可以将用户输入内容持久性地保存的 Web 服务器端，并在一些看似正常的页面中持续性地显示，从而能够影响所有访问这些页面的 Web 用户。因此将存储式 XSS 攻击也称为持久性 XSS 攻击，通常针对留言板、论坛、博客等 Web 应用，攻击者通过以输入留言信息的方式注入包含恶意脚本代码的内容后，当其他用户访问该网页时，站点即从 Web 服务器端读取攻击代码，然后显示在页面中，并在受害者主机上的浏览器中解析并执行恶意代码。

存储式 XSS 攻击的攻击代码持久性地保存在 Web 服务器中，不需要用户单击特定的 URL 就能够执行跨站脚本，并在用户端执行恶意代码。另外，利用存储式 XSS 漏洞可以编写危害性更大的 XSS 蠕虫，XSS 蠕虫会直接影响到网站的所有用户，当一个地方出现 XSS 漏洞时，相同站点下的所有用户都可能被攻击。

(3) 基于 DOM 的 XSS 攻击。DOM(Document Object Model，文档对象模型)是一个与平台和语言无关的接口，可以使程序和脚本动态访问和更新文档的内容、结构和样式，处理后的结果能够成为显示页面的一部分。DOM 中有很多对象，其中一些(如 URL、location、refelTer 等)是用户可以操纵的。客户端的脚本程序可以通过 DOM 动态地检查和修改页面内容，它不依赖于提交数据到服务器端，而从客户端获得 DOM 中的数据并在本地执行。期间，如果 DOM 中的数据没有经过严格验证和过滤，就会产生基于 DOM 的 XSS (DOM-based XSS)漏洞。

传统的 XSS 漏洞都存在于用来向用户提供 HTML 响应页面的 Web 服务器中，而基于 DOM 的 XSS 漏洞则发生在客户端处理内容的阶段。基于 DOM 的 XSS 攻击源于 DOM 相关的属性和方法，在实现过程中被插入用于 XSS 攻击的脚本。下面是一个典型的例子。

HTTP 请求 http://Website/welcome.html? name=wangqun 使用以下的脚本打印出登录用户 wangqun 的名称，即

```
<SCRIPT>
var pos = docmnent.URL.indexOf("name = ") + 5:
document.write (document.URL.substring(pos,document.URL.1ength));
</SCRIPT>
```

如果这个脚本用于请求 http://Website/welcome.html? name=<script>alert("XSS")</script>，就会导致 XSS 攻击的发生。当用户单击这个链接时，Web 服务器返回包含上述脚本的 HTML 静态文本，用户端浏览器把 HTML 文本解析成 DOM，DOM 中的 document 对象 URL 属性的值就是当前页面的 URL。在脚本被解析时，这个 URL 属性值的一部分被写入 HTML 文本，而这部分 HTML 文本便是 JavaScript 脚本，这使得<script>

alert("XSS")</script >成为页面最终显示的 HTML 文本，从而导致基于 DOM 的 XSS 攻击的发生。

3. XSS 攻击的防范方法

虽然 XSS 的表现形式多种多样，利用方法又灵活多变，但恶意脚本执行都是在客户端的浏览器上，危害的也是客户端的安全。可以从以下几个方面重点加强对 XSS 攻击的防范。

（1）XSS 过滤。虽然 XSS 攻击的对象是客户端，但 XSS 的本质是 Web 应用服务的漏洞，所以必须同时对 Web 服务器和客户端进行安全加固才能避免攻击的发生。XSS 过滤需要在客户端和服务器端同时进行。

由于 XSS 攻击是利用一些正常的站内交互机制来实现的，如通过发布评论、添加文章等方式来提交含有恶意 JavaScript 的内容，服务器端如果没有过滤或转义掉这些脚本，反而作为内容发布到页面上，那么当正常用户访问该页面时就会运行这些恶意攻击脚本。因此需要针对“< >”“JavaScript”等敏感字符串进行过滤，如果发现用户输入的信息中包含有可疑字符串，在保存到服务器端之前需要对其进行转义或直接禁用。

由于 XSS 攻击的目标是客户端，具体在浏览器上解析和执行，因此客户端防范 XSS 攻击的有效方法是提升浏览器的安全性。一方面，可以使用自带 XSS 过滤插件的安全浏览器，只允许受信任的网站启用 JavaScript 等脚本；另一方面，通过对浏览器的安全设置（如提高访问非受信网站时的安全等级、关闭 Cookie 功能等），尽量降低浏览器的安全风险。

（2）输入验证。输入验证就是对用户提交的信息进行有效性验证，仅接收有效的信息，阻止或忽略无效的用户输入信息。在对用户提交的信息进行有效性验证时，不仅要验证数据的类型，还要验证其格式、长度、范围和内容。

在进行输入验证时，需要对所有输入中的 script、iframe 等字样进行严格的检查。这里的输入不仅仅是用户可以直接交互的输入接口，还包括 HTTP 请求报文中的变量等。

大部分 Web 应用程序会依靠客户端来验证用户提交给服务器的数据，从而提高程序的可用性。不过，仅仅在客户端对非法输入进行验证和测试是不够的，因为客户端组件和用户输入不在服务器的控制范围内，用户能够完全控制客户端及提交的数据，从而绕过客户端的检查而将信息直接提交给服务器。为此，对客户端提交数据的安全性进行检查，还必须依靠服务器的防范措施，如 CSS 过滤。

（3）输出编码。由于大多数 Web 应用程序都会把用户输入的信息完整地输出到页面中，因此导致 XSS 漏洞的存在。为解决这一问题，当需要将一个字符串输出到 Web 网页，但又无法确定这个字符串是否包含 XSS 特殊字符时，为了确保输出内容的完整性和正确性，可以使用 HTML 编码（HTML Encode）进行处理。

HTML 编码通过用对应的 HTML 实体编号来替代字符串（如将字符串“<”替换为实体编号“&＃60;”），可使浏览器安全处理可能存在的恶意字符，将其当作 HTML 文档的内容而非结构加以处理，通过编码转义可有效防范 XSS 攻击的发生。

总体来说，相对于其他网络漏洞攻击，因跨站漏洞而引起的 XSS 攻击显得更隐蔽和难以防范。XSS 攻击过程是用户浏览器与网站服务器 Web 程序的交互过程，因此安全管理和防范也同时涉及网站和浏览器两个方面，但防范重点应放在网站程序的编写上。要求网站开发者在编写程序代码时要检测其安全性，如过滤用户提交数据中的代码，不再将数据作为

代码直接来处理，限制输入字符的类型和长度，限制用户上传 Flash 文件等；同时要求用户浏览时也应采用安全的浏览方式，如不轻易单击网站的链接、提高浏览器的安全等级等。

5.5 Web 服务器软件的攻防

IIS(Internet Information Services)和 Apache 是目前应用较为广泛的两款典型的 Web 服务器软件，除此之外，还有 IBM WebSphere、Oracle IAS、BEA WebLogic 和 Tomcat 等。每一款软件都有其应用优势和最佳应用环境，同样也都不同程度地存在着安全隐患，攻击者可以利用存在的安全漏洞对 Web 服务器实施渗透攻击或窃取敏感信息。

5.5.1 Apache 攻防

近年来，虽然 Nginx、LightHttpd 等 Web 服务器软件得到了快速发展，也逐渐得到了用户的青睐，但 Apache HTTP Server(简称 Apache)在这一领域的主导地位仍然没有被撼动。如今，互联网上大多数的 Web 应用仍然运行在 Apache Httpd(httpd 是 Apache HTTP 服务器的主程序)上。目前，虽然在互联网上存在各种类型的 Web 服务器软件，但不管采用哪种软件，基本的安全问题主要集中在 3 个方面：不必要的服务带来的安全威胁、基础安全认证机制和协议存在的安全缺陷及 Web 服务器自身存在的安全漏洞。

1. 针对 Apache 模块的攻防

一般情况下，Web 服务器的安全主要集中在两点：一是 Web 服务器自身的安全；二是 Web 服务器是否提供了可供使用的安全功能。与其他服务器软件一样，Apache 同样也因出现一些高危安全漏洞导致系统服务出现安全问题，但通过对近年来发生的大量安全漏洞的统计分析，Apache 的高危漏洞主要集中在 Apache 模块(Apache Modules)，而非 Apache 核心程序。这是因为 Apache 核心程序的设计是非常安全的，但大量的官方和非官方模块的出现，在丰富了 Apache 应用功能的同时，也带来了大量的安全隐患。尤其在安装了 Apache 后，默认安装和启动的模块中存在不少安全漏洞。

出于安全防范，首先要检查 Apache 模块的安装情况。最小权限原则同样也适用于 Apache 模块的安装，应该不安装或尽可能少地安装不必要的模块，以减少系统出错的机会。对于已经安装的模块，也要确保升级为最新版本，防止安全漏洞的出现和被利用。

2. 针对 Apache 管理员账户的攻防

首先需要说明是，以 root 或 admin 身份运行 Apache 进程是非常不安全的。这是因为 root 或 admin 是服务器管理员在管理计算机系统时使用的身份，一般具有最高的权限，可以在系统中从事管理脚本、访问配置文件、读取日志等操作。这时，如果攻击者以管理员身份登录系统，将直接获取一个最高权限的 Shell。同时，应用程序本身将具有较高权限，当应用程序存在漏洞时，将会带来安全风险，如删除本地硬盘上的重要文件、终止服务进程等，其中有些操作带来的后果是灾难性的。

正确的配置和防范方法是：为 Apache 进程的运行使用专门的用户账户(user/group)，而且这个用户账户唯一的作用是运行 Apache 进程，而不应具有 Shell 权限。

3. 正确配置 Apache 服务器

错误的或不恰当的配置是导致 Apache 服务器软件存在安全问题的一个主要原因。由于 Apache 是一个较为复杂的系统，因此下面举例来对常用的配置内容进行说明。

(1) Apache 服务器配置文件。Apache 服务器主要有 3 个配置文件，位于“/usr/local/apache/conf”目录下。这 3 个配置文件分别为 httpd.conf(主配置文件)、srm.conf(添加资源文件)和 access.conf(设置文件的访问权限)。其中，每一个配置文件都涉及大量的参数，许多参数的配置都与安全直接相关。

例如，文件 access.conf 中包含着一些指令用于控制允许哪些用户能够访问 Apache 目录。应该把 deny from all 设为初始化指令，再使用 allow from 指令打开访问权限。

```
order deny,allow
deny from all
allow from abc.net
```

通过该设置，可允许来自某个域(如 abc.net)、IP 地址或者 IP 地址段的访问。

(2) Apache 服务器的密码保护。htaccess 文件是 Apache 服务器中的一个配置文件，它负责相关目录下的网页配置。例如，通过 htaccess 文件可以帮助用户实现文件夹密码保护、用户自动重定向、自定义错误页面、改变用户的文件扩展名、限制特定 IP 地址的用户访问、只允许特定 IP 地址的用户访问、禁止目录列表，以及使用其他文件作为 index 文件等一些功能。

管理员可以通过对.htaccess 文件的配置，把某个目录的访问权限赋予某个用户。需要启用.htaccess 文件时，管理员首先要通过修改 httpd.conf 来启用 AllowOverride，并可以用 AllowOverride 限制特定命令的使用。如果需要使用.htaccess 以外的其他文件名，可以用 AccessFileName 指令来改变。

5.5.2 IIS 攻防

针对任何一台提供互联网应用服务的 Web 服务器，管理员必须建立一套完善的完全管理策略，而且必须熟悉策略中每一项设置的功能。Windows Server 中的 IIS 采用模块化设计，默认只会安装基本的功能组件，其他功能在用户需要时由系统管理员自动添加，以此减少 IIS 网站的被攻击面，减少系统管理员所要面对的不必要的安全挑战。同时，IIS 也提供了一些安全措施来强化网站的安全性。

1. 安全漏洞

虽然 IIS 提供了必需的安全管理措施，但由于 IIS 存在的各类安全漏洞及一些系统管理员缺乏必要的安全管理意识，长期以来一直是攻击者首选的攻击对象。

与其他的网络服务守护进程一样，IIS 同样也面临着缓冲区溢出、不安全代码和指针、格式化字符串等一系列攻击，这类攻击是基于数据驱动安全漏洞的远程渗透攻击，往往能够让攻击者在 Web 服务器上直接获得远程代码的执行权，并执行一些操作。IIS 6.0 之前的多个版本都存在该安全漏洞。

作为 HTTP1.1 的扩展，WebDAV（Web-based Distributed Authoring and Versioning，Web 分布式创作和版本控制）已经成为重要的 Web 通信协议。对于使用 WebDAV 的客户端可以进行如下的操作：在 WebDAV 目录中复制和移动文件、修改与某些资源相关联的属性、锁定并解锁资源以便多个用户可同时读取一个文件、搜索 WebDAV 目录中的文件的内容和属性等。由于使用 WebDAV 较为方便，因此经常用于对 IIS 网站的远程管理，如图 5-10 所示。

图 5-10　WebDAV 登录界面

然而，针对 WebDAV 的安全漏洞频繁出现。例如，WebDAV 本地提权漏洞（CVE-2016-0051）就是一个针对客户端验证输入不当而存在的特权提升漏洞，攻击者利用该安全漏洞可以使用提升的特权执行任意代码。又如，CVE-2017-7269 是 IIS 6.0 中存在的一个栈溢出漏洞，在 IIS 6.0 处理 propfind 指令时，由于对 URL 的长度没有进行有效的控制和检查，导致执行 memcpy 对虚拟路径进行构造时引发堆栈溢出，该漏洞可以导致远程代码执行。

针对 IIS 安全漏洞最有效的防范方法仍然是及时安装补丁程序，并坚持最小权限原则对系统进行管理和安全配置。例如，针对 IIS 写文件漏洞，除升级最新补丁程序外，还需要对 WebDAV 组件进行安全配置。如果没有用到 WebDAV 功能，建议直接将其禁用。如果一定要使用该组件时，可以对其权限（尤其是“写入”和“脚本资源访问”权限，如图 5-11 所示）进行严格管理。

2. IIS 的安全配置

作为一款应用广泛（尤其是中小单位广泛使用）的 Web 服务器软件，可以从以下几个方面加强对 IIS 的安全配置和管理。

（1）禁用默认网站。IIS 中的“默认网站”是系统提供的用于对 IIS 运行状态进行测试的网站，如图 5-12 所示。由于该网站仅仅是用于性能测试，没有过多的考虑其安全性，因此出于安全考虑，在 IIS 安装结束并测试运行正常后将“默认网站”禁用。更不能直接在默认网站的基础上，通过修改代码和配置来发布自己的网站。

（2）防止资源解析攻击。Web 服务器软件在处理资源请求时，根据不同 Web 客户端的

图 5-11　设置 WebDAV 文件夹的访问权限

图 5-12　查看 IIS 提供的“默认网站”

要求，针对同一资源（如 Web 页面）需要解析为不同的标准化名称，将这一过程称为资源解析。例如，HTTP 的资源请求 URL 可以用 Unicode 方式进行编码，而 Web 服务器软件在接受 Unicode 编码方式的 URL 时，就需要进行标准化的解析。但是，一些 Web 服务器软件可能在资源解析过程中缺乏对一些输入的合法性验证，从而导致目录遍历、敏感信息泄露、代码注入等攻击现象的发生。IIS 软件中的“ASP::$DATA”漏洞是一种典型的资源解析安全漏洞，它允许攻击者下载 ASP 源代码而不是把它们提交给 IIS ASP 引擎进行动态渲染。解决此安全问题的主要途径仍然是及时安装补丁程序。

（3）正确选择验证方式。IIS 网站默认允许所有用户连接（匿名验证）。如果网站只对特定的用户开放，就需要启用网站登录用户验证方式，即当用户正确输入用户名和密码后才能够访问。在 IIS 8.0（Windows Server 2012）中用来验证用户名和密码的方式主要有匿名

身份验证、基本身份验证、摘要式身份验证和 Windows 身份验证。4 种身份验证方式的比较如表 5-3 所示，管理员可以根据安全要求进行选择和配置。

表 5-3 IIS 提供的 4 种身份验证方式的比较

身份验证方式	安全等级	密码传输方式	是否能够通过防火墙或代理服务器
匿名身份验证	无		是
基本身份验证	低	明文	是
摘要式身份验证	中	Hash 处理	是
Windows 身份验证	高	Kerberos(Kerberos Ticket)或 NTLM(Hash 处理)	Kerberos：可通过代理服务器，但一般会被防火墙拦截 NTLM：无法通过代理服务器，但可开放其通过防火墙

(4) 通过 IP 地址限制连接。根据应用要求，可以允许或拒绝某台(某个 IP 地址)或某组(某段 IP 地址)特定计算机连接指定的网站，如图 5-13 所示。例如，单位内部网站可以被设置成只允许内部计算机访问，拒绝所有外部的网络连接。这时可以在 IIS 中针对特定网站进行设置，当管理员通过 IP 地址限制某台或某组客户端计算机不可以连接网站后，所有来自被限制计算机的 HTTP 连接请求都会被拒绝，增加了系统应用的安全性。

图 5-13 设置 IP 地址和域名限制

另外，IIS 还提供了“动态 IP 限制”功能，可以根据连接行为来决定是否允许客户端的连接。该功能的实现主要基于以下两个策略。

(1) 基于并发请求数量拒绝 IP 地址。如果同一个客户端的并发连接数量超过此处的设置值，就拒绝其连接。

(2) 基于一段时间内的请求数量拒绝 IP 地址。如果同一个客户端在指定时间内的连接数量超过此处的设置值，就拒绝其连接。

习题

1. 简述 C/S 结构和 B/S 结构的特点。

2. Web 应用安全涉及哪些具体内容，简述其安全防范方法。

3. 为什么在对 Web 服务器实施攻击之前需要进行信息的收集？具体应收集哪些内容？如何收集？

4. 简述网络踩点、网络扫描和网络查点的功能，并通过具体操作掌握其应用。

5. 简述 PING 命令的功能，并简述如何使用 PING 来探测一台主机的连通性。

6. 在熟悉 TCP 和 UDP 工作原理的基础上，分别简述基于 TCP 和 UDP 进行主机扫描的方法。

7. 在熟悉端口概念和应用分类的基础上，简述端口扫描的功能及常见的 TCP 和 UDP 端口扫描的实现方法。

8. 什么是系统类型探测？分别简述操作系统类型和网络服务类型探测的实现原理，并借助工具软件（如 nmap 等）进行实验测试。

9. 漏洞扫描的工作原理是什么？漏洞扫描器是如何工作的？简述漏洞扫描器各组成部分的功能。

10. 什么是网络查点？它与网络踩点、网络扫描之间存在什么关系，简述常见的网络查点方法。

11. 针对 Web 服务器的各类信息收集攻击，如何进行有效防范？

12. 互联网环境中的敏感数据主要包括哪些内容？如何加强对敏感数据的管理？

13. 什么是网站篡改？如何进行防范？

14. 什么是内容注入和 SQL 注入？简述 SQL 注入攻击的实现原理和具体过程，并简述其防范方法。

15. 什么是跨站脚本漏洞与跨站脚本攻击？XSS 攻击分为哪几类？如何有效防范 XSS 攻击？

16. 针对 Web 服务器软件的攻防，分别简述 Apache 和 IIS 服务器的安全配置方法。