高等职业教育新目录新专标
电子与信息大类教材

Web安全与防护

王立进　朱宪花　主　编
李　臻　张宗宝　张　镇　副主编
张建标　主　审

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

Web 系统是目前最为流行的架构，由于它是黑客攻击的重要目标，因此迫切需要大量掌握 Web 安全攻防技术的人才提高其安全性。本书结合渗透测试项目实施过程，分为 Web 系统安全技术基础、信息收集与漏洞扫描、利用漏洞进行渗透测试与防范、项目验收 4 个部分，共 10 个单元，详细介绍了 Web 系统安全技术与利用漏洞进行渗透测试的方法。每个单元理论知识与实训任务相结合，较好地体现了理实一体化的教学理念。为便于学习，本书主要针对基于 PHP+MySQL 开发的 Web 系统安全攻防技术，实训内容图文并茂，易于实训任务的开展。为了使学生对 Web 安全技术融会贯通，本书讲解力求深入至 Web 系统程序代码层面。

本书体系完整，内容翔实，配套资源丰富，可供高职院校开设 Web 安全技术课程的学生使用，也可作为本科院校学生学习 Web 安全技术的入门教程，同时也可作为技术人员自学 Web 安全技术的参考书。

图书在版编目（CIP）数据

Web 安全与防护 / 王立进，朱宪花主编. —北京：电子工业出版社，2022.11

ISBN 978-7-121-43220-0

Ⅰ. ①W… Ⅱ. ①王… ②朱… Ⅲ. ①计算机网络—网络安全 Ⅳ. ①TP393.08

中国版本图书馆 CIP 数据核字（2022）第 052602 号

责任编辑：左　雅
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1092　1/16　　印张：12.75　　字数：326 千字
版　　次：2022 年 11 月第 1 版
印　　次：2022 年 11 月第 1 次印刷
定　　价：45.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254580 或 zuoya@phei.com.cn。

前　言

Web 系统是目前最为流行的架构，也是受黑客攻击最多的目标。据国家互联网应急中心（CNCERT）监测报告，在 2020 年我国境内网站被篡改的数量达到 243 709 个，提高 Web 系统的安全性刻不容缓，迫切需要大量掌握 Web 安全技术的人才提高其安全性。

本书将 Web 安全技术融入渗透测试项目实施的各个过程，既让学生掌握相关技术与技能，又让学生熟悉渗透测试项目实施的过程。全书分为 Web 系统安全技术基础、信息收集与漏洞扫描、利用漏洞进行渗透测试与防范、项目验收 4 个部分，共 10 个单元，每个单元包括相关的理论知识及实训任务。

第一部分包括 Web 系统安全形势与威胁、Web 系统架构与技术、HTTP、Web 系统控制会话技术等内容。

第二部分包括信息收集与漏洞扫描的内容，为后续的测试与防范做好准备。

第三部分是本书的重点内容，包括 SQL 注入、跨站脚本、文件上传、命令执行、文件包含、跨站请求伪造、反序列化漏洞的渗透测试与防范，共 7 个单元，根据不同的漏洞介绍相应的渗透测试技术与防范方法。

第四部分主要介绍项目验收的环节。

本书由校企双元开发，整体特点是“易教易学”，讲解力求深入至程序代码层面，无缝融入课程思政内容，融合职业技能竞赛要求的知识点与形式。

（1）本书采用理实相结合的授课方式，易于教授。每个单元的知识点都对应相应的实训任务。实训环境以开源系统为主，辅以可自行编写的小程序，采用的工具易于下载，方便实训教学的开展。

（2）本书主要针对 PHP+MySQL 开发的 Web 系统的安全技术，实训内容图文并茂，方便学生学习。有更高追求的学生可在此基础上横向扩展学习基于其他语言或数据库开发的 Web 系统的安全技术。

（3）本书侧重于 Web 系统自身的安全技术，因此力求深入到程序代码层面。漏洞形成原理、利用及防范方法都结合源代码进行讲解，使学生真正理解相关知识点，做到融会贯通。

（4）本书融合了职业技能竞赛要求的知识点与形式。在各类网络安全大赛中 Web 类型的 CTF 题目占有较大比重，本书在习题部分含有龙头企业专家编写的 CTF 题目，使学生加强对相关知识的掌握，深入理解职业技能竞赛要求的知识点与形式，有效提升学生对 Web 安全技术的兴趣。

为便于教与学，本书每单元提供若干微课视频，请扫描书中二维码观看学习。配套课

件、习题答案等请登录华信教育资源网（http://www.hxedu.com.cn）注册后免费下载。

本书由山东科技职业学院王立进、山东电子职业技术学院朱宪花担任主编，山东信息职业技术学院李臻、山东科技职业学院张宗宝、启明星辰技术总监张镇担任副主编，北京工业大学教授、博导张建标担任主审。王立进编写了单元 1 至单元 6，朱宪花编写了单元 9 及练习题目，李臻编写了单元 7，张宗宝编写了单元 8，张镇编写了单元 10 及习题中的 CTF 题目部分。本书的编写还得到 Web 前端技术国家创新团队的倾力支持；另外，在编写过程中，参考了吴翰清、张炳帅等信息安全专家及学者的著作，在此一并表示感谢。

由于信息安全攻防技术不仅涉及知识面广，而且要求深入，加之作者水平有限，时间仓促，书中难免有不足之处，欢迎各位专家、同人、读者批评指正。

编　者

2022 年 8 月

目　　录

单元 1　Web 系统安全技术基础

学习目标

通过本单元的学习，学生能够掌握 Web 系统架构、熟悉 Web 系统所采用的技术、熟悉 HTTP 相关规定、理解 Cookie 与 Session 的作用及区别、理解 Web 系统面临的威胁及威胁路径等知识。

培养学生搭建 Web 系统运行环境、安装 Web 系统、抓取分析 HTTP 数据包的技能。

培养学生发现、利用、加固 Web 系统漏洞的能力。

培养学生保障 Web 系统安全的价值观。

情境引例

根据国家互联网应急中心（CNCERT）监测报告，在 2020 年我国境内网站被篡改的数量达到 243 709 个，其中被篡改的政府网站达 1030 个。典型的事件有：

1. 境外“图兰军”黑客组织对我国网站发起攻击。境外“图兰军”黑客组织于 2019 年 12 月 22 日成立，据不完全统计，在 2020 年内攻击篡改了至少 100 个中国网站。

2. 疫情期间多个黑客组织对我国发起网络攻击。2020 年年初，在新冠肺炎疫情期间，多个国家和地区的黑客组织对我国发动网络攻击。境外“海莲花”黑客组织利用疫情话题攻击我国政府机构网站，境外“白象”黑客组织借新型肺炎对我国网络发起攻击，“绿斑”黑客团伙利用虚假“疫情统计表格”和“药方”窃取情报。

这些网络安全事件充分体现了“没有网络安全就没有国家安全”，要实现网络安全，尤其是 Web 系统的安全，迫切需要大量掌握 Web 安全技术的人才。

1.1　Web 系统安全形势与威胁

1.1.1　Web 系统安全形势

Web 系统是目前最为流行的系统架构，其广泛应用于银行服务、电子商务、购物平台、Web 网站、社交网络等多个领域。Web 系统之所以越来越流行，在于其有许多优点：

（1）Web 系统通信的核心协议是 HTTP，它是轻量级的，无需连接。其还可通过代理和其他协议传输，允许在任何网络配置下进行安全通信。

（2）Web 用户只需要有浏览器，就可访问 Web 服务器。现在浏览器功能强大，Web 应用程序利用浏览器可为用户动态生成丰富的用户界面。

（3）用于开发 Web 应用程序的核心技术和语言相对简单，还有大量开源代码和其他资源可供整合到定制的应用程序中。

由于 Web 系统应用广泛，很多应用需要接入 Internet，传统的防火墙又无法对其进行有效防护，其涉及操作系统、数据库、编程语言等多方面技术，难免会出现漏洞，因此其成为黑客攻击的主要目标，不时有 Web 系统被攻破的报道。

还有一些与 Web 系统安全相关的案例有：

（1）疑似 5.38 亿条微博用户信息泄露。有用户发现 5.38 亿条微博用户信息在暗网出售，但是不含密码，其中 1.7 亿条有账户信息，有人指出数据来源是通过脱库进行的。2020 年 3 月 20 日，《新京报》记者购买了价值 12 元的内容，获得了 201 条微博用户信息，其中包括用户身份证号、手机号等私密信息，经过 3 条账号信息的测试，2 个微博账号查询到了正确的关联手机号。

（2）多地高校数万名学生的隐私遭泄漏。2020 年 4 月，河南财经政法大学、西北工业大学明德学院、重庆大学城市科技学院等高校的数千名学生发现，自己的个人所得税 App 上有陌生公司的就职记录。很可能是学生信息被企业冒用，以达到偷税的目的。郑州西亚斯学院多名学生反映，学校近两万名学生的个人信息被泄露，以表格的形式在微信、QQ 等社交平台上流传。

（3）含有超过 34 万条数据的智慧养老服务数据库存在安全问题。据媒体 2019 年 7 月份报道，Cybernews 研究人员发现上海孝信网络的一个含有几十万用户数据的数据库存在安全问题。这个数据库中含有超过 34 万条的用户 GPS 位置信息、个人 ID、手机号、地址，用户亲属和监护人的姓名和手机号、GPS 位置、哈希口令等敏感信息。研究人员在发现不安全的数据库后于 2020 年 1 月 14 日联系了数据库所有者，孝信很快就关闭了该数据库。

总之，Web 安全事件在信息安全事件中占有较大的比重，充分说明了 Web 系统安全形势非常严峻，急需大量掌握安全技术的人才保障 Web 系统的安全。

1.1.2 Web 系统威胁分析

1. 服务器端威胁

微课 1-1 Web 服务器端威胁分析

Web 系统主要由服务器端及客户端两部分组成，这两部分都可能会成为攻击目标，但是服务器端是 Web 系统的核心，因此它是非法入侵者攻击的首要对象。

服务器端涉及操作系统、系统服务及 Web 应用三个方面，因此入侵者在渗透服务器端时可从操作系统、系统服务及 Web 应用三个层面进行攻击，如图 1-1 所示。

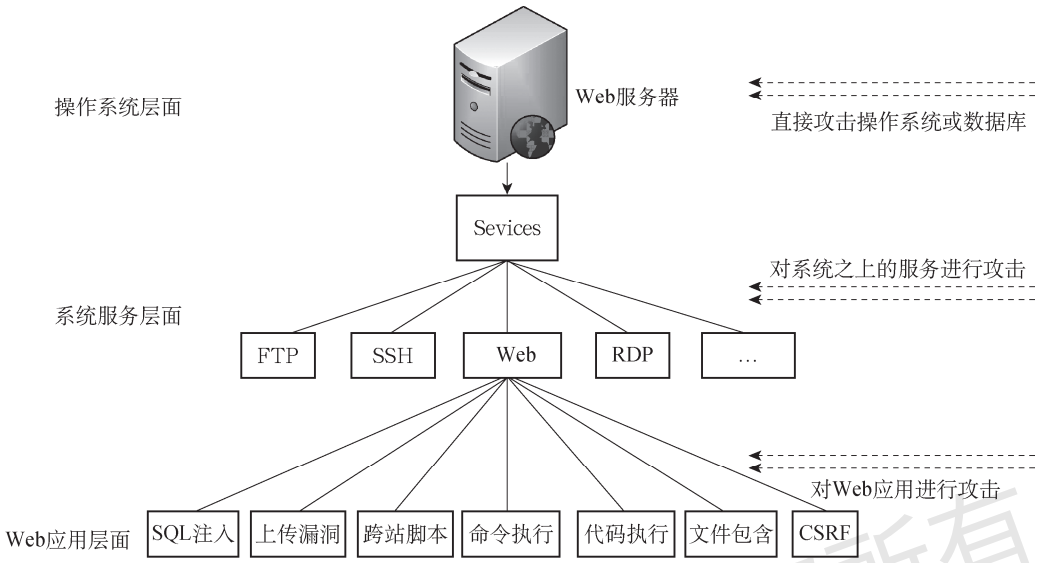

图 1-1　攻击 Web 系统路径

（1）操作系统层面。

操作系统层面的攻击主要是指利用服务器操作系统，如 Windows、Linux、UNIX 等存在的漏洞或者配置错误进行的攻击。常采用的攻击方式有暴力破解、网络监听、ARP 欺骗、缓冲区溢出等。

（2）系统服务层面。

系统服务层面的攻击是指利用操作系统之上运行的服务，如 FTP、SSH、RDP 等存在的漏洞进行攻击。攻击方法与操作系统层面的攻击大致相同。

（3）Web 应用层面。

Web 应用层面攻击是指利用应用程序在编码过程中出现的漏洞或逻辑错误对 Web 系统进行的攻击，如数据库注入、跨站脚本攻击、文件上传漏洞等。

一般情况下，Web 系统中服务器端都通过防火墙等网络安全设备进行防护，系统及相应的服务很难访问到，因此其中漏洞难以利用。而 Web 系统要对外提供服务，就必须开放 HTTP 或 HTTPS，因此 Web 应用层面就成为主要的攻击路径，对 Web 应用层面的防护也成为信息安全保障的重点。

2. 客户端安全威胁

用户在使用 Web 系统时，也可能会由于 Web 系统错误受到攻击，如跨站脚本攻击、跨站请求伪造等，导致用户的利益受到损失，影响 Web 系统拥有者的声誉，因此 Web 系统也必须重视对客户端的防护。

1.1.3　OWASP 十大 Web 系统安全漏洞

微课 1-2　OWASP 十大 Web 系统安全漏洞

开放式 Web 应用程序安全项目（Open Web Application Security Project，OWASP）关注 Web 应用程序的安全，定期更新 Web 系统的

“十大漏洞”，以提高大家对 Web 安全的意识。2017 年公布的十大 Web 系统的安全漏洞及说明如表 1-1 所示。

表 1-1　OWASP 十大 Web 系统安全漏洞

漏洞名称	漏洞说明	漏洞影响
A1-注入	未经过滤的数据作为命令或查询的一部分发送到解析器进行解释时，会产生诸如 SQL 注入、OS 注入和 LDAP 注入的缺陷	攻击者利用该漏洞可以输入恶意数据，能够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
A2-失效的身份认证和会话管理	应用程序的身份认证和会话管理功能不完善导致的缺陷	攻击者能够利用该漏洞破译密码、密钥或会话令牌，或者利用其他开发缺陷来暂时性或永久性冒充其他用户的身份
A3-敏感信息泄露	Web 应用程序和 API 没有对敏感数据正确保护，如密码、信用卡卡号、医疗记录、个人信息等	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
A4-XML 外部实体（XXE）	有些较早的或配置错误的 XML 处理器引用了 XML 文件中的外部实体引起的缺陷	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
A5-失效的访问控制	未对通过身份验证的用户实施恰当的访问控制引起的缺陷	攻击者可以利用这些缺陷访问未经授权的功能或数据，例如：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
A6-安全配置错误	由不安全的默认配置、不完整的临时配置、开源云存储、错误的 HTTP 标头配置及包含敏感信息的详细错误信息所造成的缺陷	攻击者可以利用安全配置错误对系统进行攻击，对系统造成极大威胁
A7-跨站脚本（XSS）	当应用程序输出的 HTML 页面包含不受信任的、未经恰当验证或转义的数据，如 JavaScript 命令，就会在浏览器执行非法命令时出现 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
A8-不安全的反序列化	反序列化是由保存的文本格式或字节流格式还原成对象的过程，如果应用代码允许接受不可信的序列化数据，在进行反序列化操作时，可能会产生反序列化漏洞	攻击者可以利用该漏洞来执行拒绝服务攻击、访问控制攻击和远程命令执行攻击
A9-使用含有已知漏洞的组件	组件（例如：库、框架和其他软件模块）拥有和应用程序相同的权限，因此组件存在漏洞导致 Web 应用程序存在缺陷	该漏洞会造成严重的数据丢失或服务器接管。另外，可能会破坏应用程序防御，造成各种攻击并产生严重影响
A10-不足的日志记录和监控	日志记录和监控不足导致的缺陷	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

1.1.4　Web 系统渗透测试常用工具

渗透测试就是模拟黑客的漏洞挖掘及利用手法，在客户的授权下，非破坏性的攻击性测试，并根据测试结果提供整改建议。针对 Web 系统渗透测试，有时只需要使用标准的浏

览器即可实施，但绝大多数要求使用一些其他工具。这些工具主要包括三类：一是 Web 浏览器类，二是漏洞扫描类，三是漏洞利用类，当然有些工具包括多种功能，集成测试套件有 Burp Suite、WebScarab 等。

- Web 浏览器类工具常用 Firefox、IE 等。
- 漏洞扫描类工具包括常用的端口扫描器 Nmap、漏洞扫描器 Nessus、Web 漏洞扫描器 WAVS、AppScan 等。
- 漏洞利用类工具包括 SQLMap、Metaspolit、Hydra 等。

1.2　Web 系统架构与技术

Web 系统是目前最为流行的系统架构，其广泛应用于银行服务、电子商务、购物平台、Web 网站、社交网络等多个领域。Web 系统之所以越来越流行，在于其有许多优点：

- Web 系统通信的核心协议是 HTTP 协议，它是轻量级的，无需连接。其提供了对通信错误的容错性。HTTP 还可通过代理和其他协议传输，允许在任何网络配置下进行安全通信。
- Web 用户只需要有浏览器，就可访问 Web 服务器。Web 应用程序为浏览器动态生成用户界面。界面变化只需在服务器上执行一次，就可立即生效。现在浏览器功能强大，可构建丰富并且令人满意的用户界面。
- 用于开发 Web 应用程序的核心技术和语言相对简单，还有大量开源代码和其他资源可供整合到定制的应用程序中。

1.2.1　Web 系统架构

Web 系统采用 B/S 架构，即提供服务的一端为服务器端（Server），而客户端采用浏览器（Browser）进行访问，其采用 HTTP 或 HTTPS 协议进行信息交互。Web 系统架构如图 1-2 所示。

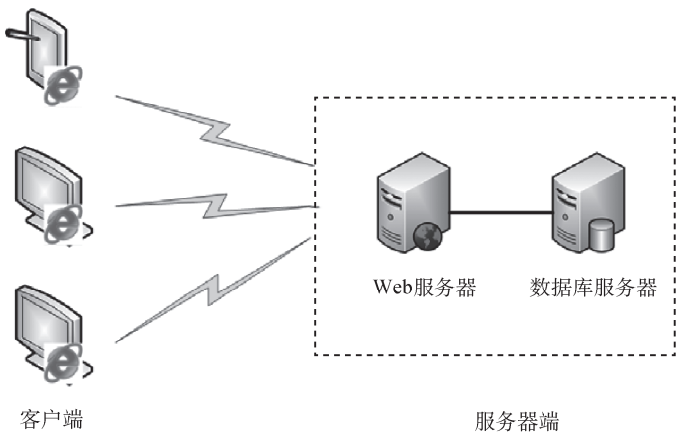

图 1-2　Web 系统架构

Web 系统主要由服务器端与客户端两部分组成。

服务器端是 Web 系统的核心，主要向用户提供动态生成的内容，当用户请求一个资源时，服务器动态建立响应，每个用户都会收到满足其特定需求的内容。其一般由 Web 服务器（包括 Web 容器及 Web 应用程序）+数据库服务器两部分组成。

客户端采用浏览器访问服务器端，即向服务器提出请求。常见的浏览器有 IE（Internet Explorer）、Safari、Firefox、Opera、Chrome 等。访问服务器非常简单，在浏览器的地址栏中输入 URL，只要 Web 服务器提供相应的服务即可进行访问。URL 的一般语法格式为：

```
protocol :// hostname[:port] / path / [;parameters][?query]
```

其中，

protocol：协议，常用的协议是 HTTP、HTTPS。

hostname：主机地址，可以是域名，也可以是 IP 地址。

port：端口，HTTP 协议默认端口是 80 端口，如果省略不写就表示 80 端口。

path：路径，网络资源在服务器中的指定路径。

parameter：参数，如果要向服务器传入参数，则在这里输入。

query：查询字符串，如果需要从服务器中查询内容，则在这里输入。

1.2.2 服务器端技术

微课 1-3 Web 服务器端技术

服务器端综合运用 Web 容器、Web 应用程序编程语言（脚本）、数据库及其他后端组件等。

1. Web 容器

Web 容器就是一种服务程序，处理从客户端收到的请求，并与 Web 应用程序交互，并把结果反馈给客户端。Web 容器给处于其中的应用程序组件提供环境，使其直接跟容器中的环境变量交互，不必关注其他系统问题。常用的 Web 容器有 Apache、IIS、Tomcat、Weblogic 等。

2. Web 应用程序编程语言

常用的 Web 应用程序编程语言有 PHP、Java、ASP.NET 等。

PHP 最初是 Personal Home Page（代表个人主页），后来更新为 Hypertext Preprocessor（超文本预处理器）。现在 PHP 已经发展成为一个功能强大、应用广泛的开放源代码的多用途脚本语言，它可嵌入 HTML 脚本中，尤其适合用于 Web 开发。PHP 脚本通过 PHP 标签括起来，常用的标签格式为<?php……?>。PHP 常常与其他免费的技术融合，如所谓的 LAMP 组合（Linux、Apache、MySQL 和 PHP）。

Java 平台企业版（J2EE）已经成为事实上的大型企业常使用的标准应用程序，它应用多层与负载平衡架构，非常适合于模块化开发与代码重用。Java 平台可在 Windows、Linux 与 Solaris 操作系统上运行。

ASP.NET 是 Microsoft 公司开发的一种 Web 应用程序框架，ASP.NET 应用程序可用任何.NET 语言（如 C#或 VB.NET）编写。

3. 数据库

数据库是一种专门存储管理数据资源的系统，数据有多种形式，如文字、数码、符号、图形、图像及声音等。在 Web 系统中，应用服务器在数据库中存储、读取数据。常见的数据库有 Oracle、MSSQL、MySQL。

Oracle 数据库是甲骨文公司的一款关系数据库管理系统，目前仍在数据库市场上占有主要份额。

MSSQL 是指微软公司的 SQL Server 数据库服务器，它是一个数据库平台，提供数据库从服务器到终端的完整的解决方案，其中数据库服务器部分是一个数据库管理系统，用于建立、使用和维护数据库。

MySQL 数据库是一个多用户、多线程的 SQL 数据库，是一个客户端/服务器结构的应用，它由一个服务器守护程序 mysqld 和很多不同的客户程序和库组成。

微课 1-4　Web 客户端技术

1.2.3　客户端技术

服务器端应用程序接收用户的输入与操作，并向用户返回其结果，它必须提供一个客户端用户界面。由于所有 Web 应用程序都通过 Web 浏览器进行访问，因此这些界面共享一个技术核心，其常用的技术包括 HTML、JavaScript 及厚客户端组件。

1. HTML

HTML 的英文全称是 Hyper Text Markup Language，即超文本标记语言，是一种标识性的语言。它是建立 Web 界面的核心技术，包括一系列标签，通过这些标签可以将网络上的文档格式统一。HTML 文本是由 HTML 命令组成的描述性文本，HTML 命令可以说明文字、图形、动画、声音、表格、链接等。其中超链接和表单是 HTML 的重要内容。

（1）超链接。

客户端与服务器之间的大量通信都由用户单击超链接驱动。Web 应用程序中的链接通常包含预先设定的请求参数，这些数据项不需要用户输入，而是由服务器将其插入用户单击的超链接的目标 URL 中，以这种方式提交。例如，Web 应用程序中可能会显示一系列新闻报道链接，其形式如下：

```
<a href="/news/showStory?newsid=26789156&lang=en">come on!</a>
```

当用户单击链接时，浏览器会提出以下请求：

```
GET /news/showStory? newsid=26789156&lang=en HTTP/1.1
```

服务器收到查询字符串中的两个参数（newsid 和 lang），并根据它们的值决定给用户返回什么内容。

（2）表单。

虽然基于超链接的方法负责客户端与服务器之间的绝大多数通信，但许多 Web 应用程序还是需要采用更灵活的形式收集输入，并接收用户输入。HTML 表单是常见的整片机制，允许用户通过浏览器提交任意输入。以下是一个典型的 HTML 表单。

```
<form action="check.php" method="POST">
      用户名：<input type="text" name="username" /></br>
      密 码：<input type="password" name="password" /></br>
      <input type="submit" name="submit" value="提交">
</form>
```

当用户在表单中单击“提交”按钮时，浏览器将提出如下请求：

```
POST check.php  HTTP/1.1
HOST: library.edu.cn
Content-Type: application/x-www-form-urlencoded
Content-Length:32

Username=admin&password=stone69&submit=提交
```

因为 form 标签中指定了 POST 方法，浏览器就使用这个方法提交表单，并将表单的数据存入请求的消息主体中。

2. JavaScript

JavaScript 是一种相对简单但功能强大的编程语言，其可使许多应用程序不仅使用客户端提交用户数据与操作，还可执行实际的数据处理。这样一是可以改善应用程序的性能，因为这样可在客户组件上彻底执行某些任务，不需要在服务器间来回发送和接收请求与响应；二是提高了可用性，因为这样可根据用户操作动态更新用户界面，而不需要加载服务器传送的全新的 HTML 页面。JavaScript 常用于执行以下任务：

- 在向服务器提交前确认用户输入的数据是否有效，避免因数据包含错误而提交不必要的请求。
- 根据用户操作动态修改用户界面，例如，执行下拉菜单和其他类似于非 Web 界面的控制。
- 查询并更新浏览器内的文档对象模型（Document Object Model，DOM），控制浏览器行为。

Ajax（或称为异步 JavaScript 和 XML）技术是 JavaScript 用法上的重大改进，可从 HTML 页面发布动态 HTTP 请求，与服务器交换数据并相应更新当前的 Web 页面，根本不需要加载一个新页面，增强了用户体验。

3. 厚客户端组件

为了改善 JavaScript 的功能，一些 Web 应用程序通过采用厚客户技术，使用定制的二进制代码从各方面扩展浏览器的内置功能。这些组件可配置为字节码，由适当的浏览器插件执行；或者可在客户端计算机上安装本地可执行程序。常用的厚客户端组件包括 Java applet、ActiveX 控件等。

1.2.4 实训：安装 DVWA 系统

实训目的

通过实训达到如下目的：

1. 认识 Web 系统架构。
2. 清楚 Web 系统所采用的技术。
3. 为后续单元的实训任务奠定基础。

实训原理

DVWA（Damn Vulnerable Web Application）是一个用来进行安全脆弱性鉴定的 PHP+MySQL Web 应用，旨在为安全专业人员测试自己的专业技能和工具提供合法的环境，更好地理解 Web 应用漏洞利用与安全防范的过程。DVWA 版本较多，本书使用 DVWA-master 版本，其主要模块如下：

- Brute Force（暴力破解）。
- Command Injection（命令行注入）。
- CSRF（跨站请求伪造）。
- File Inclusion（文件包含）。
- File Upload（文件上传）。
- SQL Injection（SQL 注入）。
- SQL Injection（Blind）（SQL 盲注）。
- XSS（Reflected）（反射型跨站脚本）。
- XSS（Stored）（存储型跨站脚本）。
- XSS（DOM）（DOM 型跨站脚本）。

同时每个模块的代码都有 4 种安全等级：Low、Medium、High、Impossible。通过不同难度的测试并参考代码变化可帮助使用者更快地理解漏洞的原理与防范方法。

务必注意：由于 DVWA 存在大量漏洞，因此不能将 DVWA 作为一种 Web 服务接入互联网，否则会给所在网络带来严重的安全隐患。

DVWA 系统是 PHP+MySQL 构成的 Web 系统，需要安装 Apache、MySQL、PHP 等相应软件作为运行环境。为简化实训，我们采用 XAMPP 集成软件包，其包括 Apache、MySQL、PHP、Perl 等应用，可以在 Windows、Linux、Solaris、Mac OS X 等多种操作系统下安装使用，支持英文、简体中文等多种语言，其可以为架构为 PHP+MySQL 的 Web 系统提供运行环境。

实训步骤

步骤 1：安装系统运行环境 XAMPP

1. 下载 XAMPP 安装包

在浏览器中打开 ApacheFriends 官网下载 XAMPP 最新安装包。

2. XAMPP 程序安装

双击安装程序，然后一直单击“Next”按钮，有空的地方全部打钩就可完成安装。在

选择安装目录时，不要选择系统盘。安装完成后，单击“Finish”按钮，出现 XAMPP 控制面板，如图 1-3 所示。

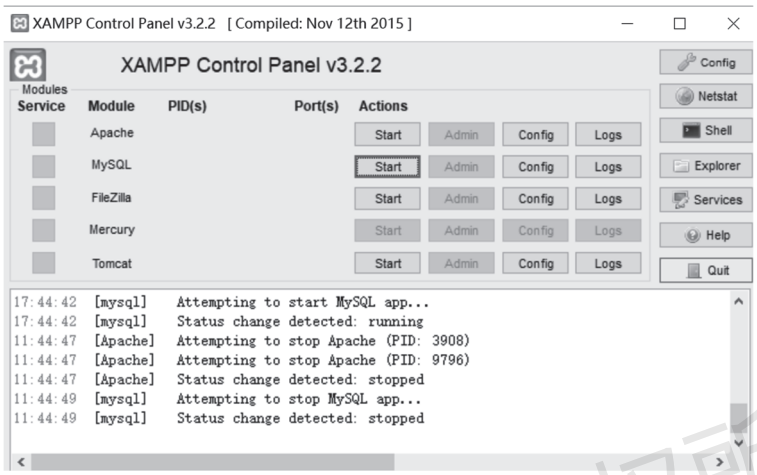

图 1-3　XAMPP 控制面板

3. 打开控制面板与启动应用

找到安装路径，此处的安装路径是 D:\XAMPP 如图 1-4 所示。

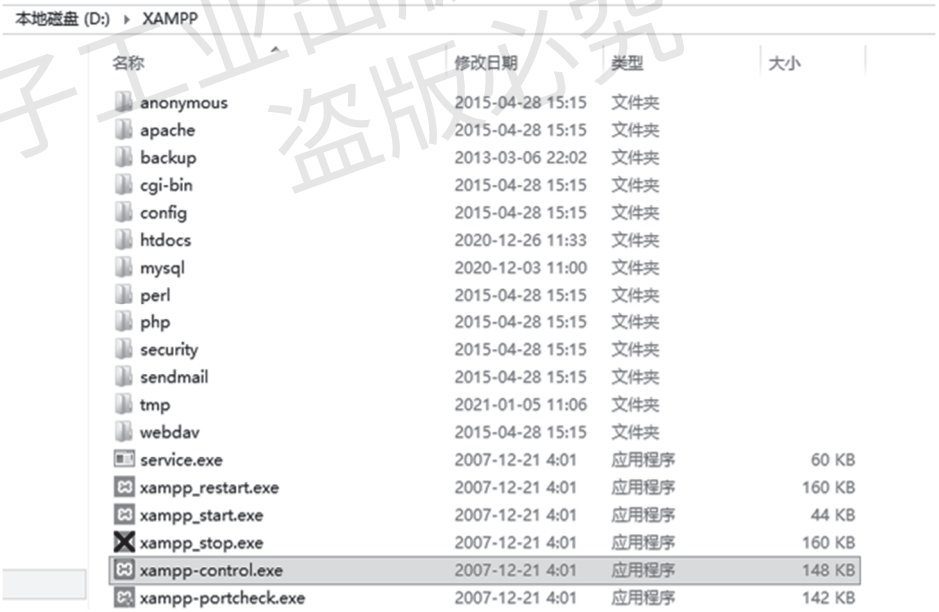

图 1-4　XAMPP 安装路径

双击该路径下的 xampp-control.exe 就可以打开控制面板。

XAMPP 有 Apache、MySQL、FileZilla 等应用，只需要单击某个应用对应的 Actions 栏中的“Start”按钮即可启动该应用。一般情况只需启动 Apache、MySQL 即可。

Apache 经常会遇到因为端口占用导致无法启动的问题。Apache 提供 HTTP 和 HTTPS 服务，它们默认对应的端口分别是 80 和 443，如果这两个端口被占用就无法启动 Apache。

解决方法一是修改 Apache 使用的端口，二是关闭占用端口的程序。修改 Apache 使用的端口的方法易于操作，因此更推荐此方法。

根据是 80 端口被占用，还是 443 端口被占用做相应配置。如果是 80 端口被占用，在控制面板中单击 Apache 对应的“Config”按钮，选择“httpd.conf”选项，将原先的“Listen 80”中的 80 修改成未被占用的端口，如 8000；如果是 443 端口被占用，在控制面板中单击 Apache 对应的“Config”按钮，选择“httpd-ssl”选项，将原先的“Listen 443”中的 443 修改成未被占用的端口。

4. 确定 XAMPP 网站根目录

Web 应用程序放置到该目录或其子目录下才能通过浏览器进行正常访问。XAMPP 默认为安装目录下的 htdocs 目录，如我们的安装目录是 D:\XAMPP，则其默认网站根目录为：D:\XAMPP\htdocs，可以通过修改 Apache 配置文件修改网站根目录，此处不多赘述。

Web 应用程序放置到根目录下，即可通过浏览器正常访问。

步骤 2：安装 DVWA 系统

1. 复制 DVWA 程序到 XAMPP 根目录

将 DVWA 压缩包解压，重命名为 DVWA，然后复制到 XAMPP\htdocs 目录下，如图 1-5 所示。

图 1-5　DVWA 系统安装路径

2. 配置 DVWA 系统

（1）在 DVWA 目录下，打开 config 目录，将其中的/config.inc.php.dist 文件名改为/config.inc.php，并用记事本等程序打开 config.inc.php 文件。

（2）修改连接 MySQL 数据库的密码。$_DVWA['db_password'] = 'p@ssw0rd'行中“=”后对应的内容为数据库的密码，在此处修改为 MySQL 数据库的真实密码。由于在 XAMPP 中 MySQL 数据库的默认密码为空，如果没有修改，就将该行修改为$_DVWA ['db_password'] = ' '。

3. 创建 DVWA 数据库

打开浏览器，在 URL 地址栏中输入：127.0.0.1/dvwa/setup.php，将出现如图 1-6 所示界面，单击该界面下的“Create/Reset Database”按钮，将完成 DVWA 数据库的创建。

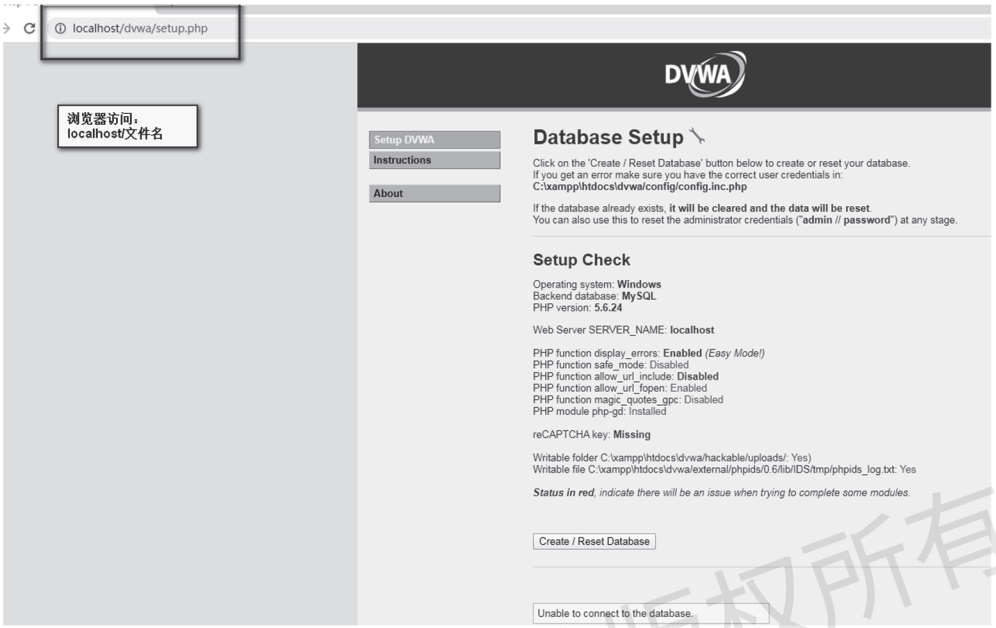

图 1-6　DVWA 系统安装界面

4. 访问 DVWA 系统

在浏览器的 URL 地址栏中输入：127.0.0.1/dvwa，登录 DVWA（默认账号：admin；默认密码：password），在登录界面输入用户名和密码，成功登录，如图 1-7 所示，至此 DVWA 系统安装成功。

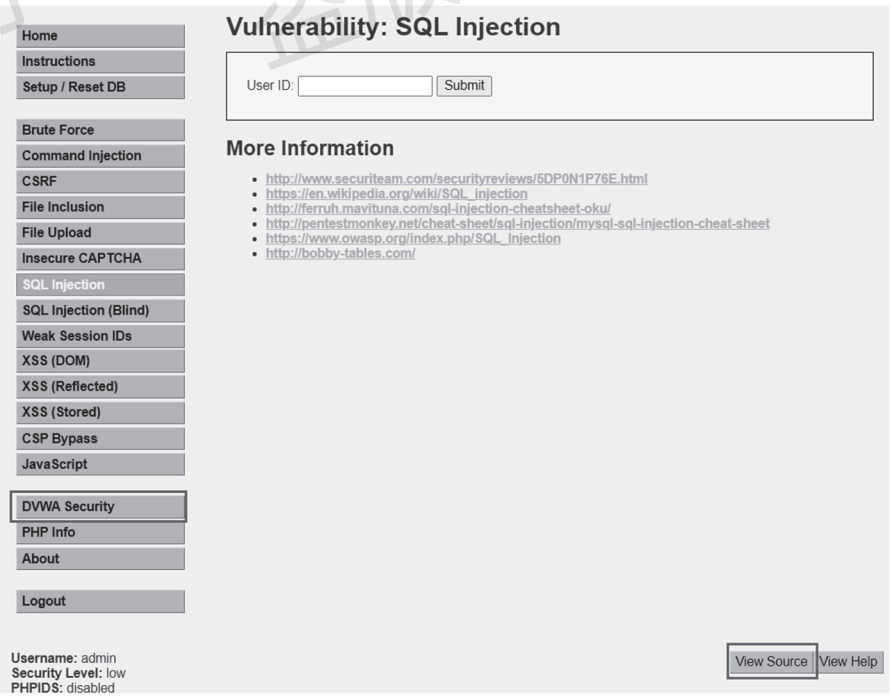

图 1-7　DVWA 系统界面

通过界面可以看到左侧导航栏列举了 Web 系统常见的 SQL Injection、XSS、File Upload 等漏洞，只要点击相应的导航菜单，就可进入相应的漏洞测试界面。

系统为每种漏洞设置了 Low、Medium、High、Impossible 4 种安全级别，通过“DVWA Security”按钮可以设置安全级别。建议在理解漏洞的原理测试时，将 DVWA Security 设置为 Low 级别，依次调高安全级别，参考源代码可更好地理解漏洞的防范方法。通过右下角的“View Source”按钮可以看到对应级别漏洞的源代码。

实训总结

1. Web 系统由服务器端与客户端两部分组成，客户端通过浏览器访问服务器。DVWA 系统的 Web 容器是 Apache，系统是通过 PHP 编写的，数据库采用 MySQL 数据库，客户端技术主要包括 HTML 及 JavaScript。

2. XAMPP 是包括 Apache、MySQL、PHP、Perl 等的应用集成软件包，可为 Web 系统提供运行环境。

3. DVWA 是一个包括 SQL Injection、XSS、File Upload 等漏洞的 Web 系统，可帮助使用者更好地理解 Web 应用漏洞利用与安全防范的过程。

1.3　HTTP

HTTP（Hyper Text Transfer Protocol，超文本传输协议）是一种用于分布式、协作式和超媒体信息系统的应用层协议。HTTP 是万维网数据通信的基础。

HTTP 是由蒂姆·伯纳斯·李于 1989 年在欧洲核子研究组织（CERN）所发起的。HTTP 的标准制定由 W3C（World Wide Web Consortium，万维网联盟）和 IETF（Internet Engineering Task Force，互联网工程任务组）进行协调，最终发布了一系列的 RFC，其中最著名的是 1999 年 6 月公布的 RFC 2616，定义了 HTTP 协议中现今广泛使用的一个版本——HTTP 1.1。2014 年 12 月，互联网工程任务组的 httpbis（Hyper Text Transfer Protocol Bis）工作小组将 HTTP/2 标准提议递交至 IESG 进行讨论，于 2015 年 2 月 17 日被批准。HTTP/2 标准于 2015 年 5 月以 RFC 7540 正式发表，取代 HTTP 1.1 成为 HTTP 的最新标准。

1.3.1　HTTP 工作原理

HTTP 定义 Web 客户端如何从 Web 服务器请求 Web 页面，以及服务器如何把 Web 页面传送给客户端。HTTP 协议采用了请求/响应模型，如图 1-8 所示。请求必须从客户端发出，最后服务器端响应该请求并返回应答。

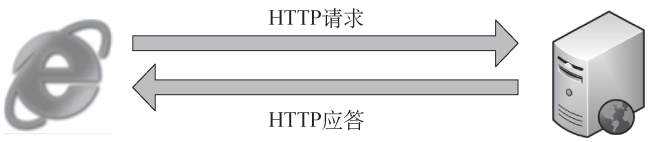

图 1-8　HTTP 请求应答模型

客户端向服务器发送一个请求报文，请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。服务器以一个状态行作为响应，响应的内容包括协议的版本、成功或错误代码、服务器信息、响应头部和响应数据。

HTTP 请求/响应的具体过程如下：

1. 客户端发送 HTTP 请求

在浏览器地址栏键入 URL，如 http://www.phei.com.cn/，其将与 Web 服务器的 HTTP 端口（默认为 80）建立一个 TCP 套接字。按下回车键后，发送 HTTP 请求。通过 TCP 套接字，客户端向 Web 服务器发送一个文本的请求报文。

2. 服务器接受请求并返回 HTTP 响应

Web 服务器解析请求，定位请求资源。服务器将资源副本写到 TCP 套接字，由客户端读取。

3. 客户端浏览器解析 HTML 内容

客户端浏览器首先解析状态行，查看表明请求是否成功的状态代码。然后解析每一个响应头，响应头告知以下为若干字节的 HTML 文档和文档的字符集。客户端浏览器读取响应数据 HTML，根据 HTML 的语法对其进行格式化，并在浏览器窗口中显示。

4. 释放 TCP 连接

若 connection 模式为 close，则服务器主动关闭 TCP 连接，客户端被动关闭连接，释放 TCP 连接；若 connection 模式为 keepalive，则该连接会保持一段时间，在该时间内可以继续接收请求。

1.3.2 HTTP 请求

微课 1-5 HTTP 请求

1. HTTP 请求报文

HTTP 请求报文由三部分组成，分别是请求行、请求头（消息头）、请求正文。以下是一个典型的 HTTP 请求：

```
   POST /dvwa/login.php HTTP/1.1       //请求行
   Host: localhost:8000                //请求头
   User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:43.0) Gecko/20100101
Firefox/43.0
   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
   Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
   Accept-Encoding: gzip, deflate
   Referer: http://localhost:8000/dvwa/login.php
   Cookie: security=high; PHPSESSID=c2370ae5baba7a3a9beed580b0262c46
   Connection: close
   Content-Type: application/x-www-form-urlencoded
   Content-Length: 45
```

```
                                                    //空白行，代表请求头结束
username=admin&password=Ilsec@web&Login=Login       //请求正文
```

（1）HTTP 请求的第一行为请求行，其由三个被空格分开的项目组成。第一个项目是说明 HTTP 方法的动词，最常用的方法是 GET。后面跟着请求的 URI 和协议的版本，格式如下：Method Request-URI HTTP-Version。其中 Method 表示请求方法，Request-URI 是一个统一资源标识符，HTTP-Version 表示请求的 HTTP 版本。

（2）第二行至空白行为 HTTP 的请求头（也称为消息头）。请求头说明了客户端向服务器端传递请求的附加信息和客户端自身的情况。常用的请求头如下：

● Host 主要用于指定被访问资源的 Internet 主机和端口号，如 Host:www.sdzy.com.cn。

● User-Agent 用于告诉服务器客户端的操作系统、浏览器和其他属性。

● Accept 用于告诉服务器客户端希望接收哪些 MIME 类型的消息，如 Accept:text/html，表示客户端希望接收 html 文本。

● Accept-Language 用于告诉服务器客户端希望接收的语言类型。

● Accept-Encoding 用于告诉服务器客户端希望接收哪些内容的编码。

● Referer 用于指示提出当前请求的原始 URL，也就是说，用户是从什么地方来到本页面的，如 Referer: http://localhost:8000/dvwa/login.php，代表用户从 login.php 来到当前页面。

● Cookie 用于向服务器提交它以前发布的 Cookie，它是存储在客户端的一段文本，常用来表示请求者身份。

● Content-Type 用于规定消息主体的介质类型。

● Content-Length 用于规定消息主体的字节长度，以字节方式存储的十进制数字表示。

● If-Modified-Since 用于说明浏览器最后一次收到被请求的资源的时间。如果那以前资源没有发生变化，服务器就会发出一个带状态码 304 的响应，指示客户使用资源的缓存副本。

（3）HTTP 请求的最后一行为请求正文，请求正文是可选的，它最常出现在 POST 请求方法中。

2. HTTP 请求方法

HTTP 1.1 中共定义了八种方法（也叫“动作”），来以不同方式操作指定的资源，其中 GET 和 POST 方法最为常见。

（1）GET 方法的作用在于向服务器获取资源。它以 URL 查询字符串的形式向被请求的资源发送请求。如果请求的资源为动态脚本，那么返回的文本是 Web 容器解析后的 HTML 源代码。通过 GET 方法向服务器端传递的参数会显示在屏幕上，并被记录在浏览器的历史记录和 Web 服务器的访问日志中，因此，请勿使用 GET 方法传送任何敏感信息。

（2）POST 方法与 GET 方法类似，但最大的区别在于，GET 方法没有请求正文，而 POST 方法有请求正文。POST 方法多用于向服务器发送大量的数据。上传文件、提交用户信息等需要传递大量数据的应用，通常都会使用 POST 方法。

（3）HEAD 方法与 GET 方法一样，都是向服务器发出指定资源的请求。只不过服务器不能在响应里返回消息主体。此方法常被用来测试超文本链接的有效性、可访问性和最近的改变。

（4）PUT 方法用于请求服务器把请求中的实体存储在请求资源下，如果请求资源存在，那么将会用此请求中的数据替换原先的数据，作为指定资源的最新修改版。如果请求资源不存在，则会创建这个资源，且把请求正文的内容作为资源内容。渗透测试人员可通过上传一段脚本，并在服务器上执行该脚本来攻击应用程序。

（5）DELETE 方法用于请求服务器删除 Request-URI 所标识的资源。服务器应该关闭此方法，因为客户端可以进行删除文件操作。

（6）TRACE 方法主要用于测试或诊断，其可回显服务器收到的请求，即服务器在响应主体中返回其收到的请求消息的具体内容。这种方法可用于检测客户与服务器之间是否存在任何操纵请求的代理服务器。

（7）OPTIONS 方法可使服务器传回该资源所支持的所有 HTTP 请求方法。用“*”来代替资源名称，向 Web 服务器发送 OPTIONS 请求，服务器通常返回一个包含 Allow 响应头的响应，并在其中列出所有有效的方法。

（8）CONNECT 方法可用于动态切换到隧道的代理。

1.3.3 HTTP 响应

微课 1-6 HTTP 响应

在接收到请求消息后，服务器会根据请求返回一个 HTTP 响应消息。

1. HTTP 响应报文

HTTP 响应报文也是由三个部分组成的，分别是响应行、响应头（消息头）、响应正文（消息主体）。以下是一个典型的 HTTP 响应：

```
    HTTP/1.1 200 OK                                          //响应行
    Date: Sat, 19 Dec 2020 10:39:01 GMT                      //响应头
    Server: Apache/2.2.9 (Win32) DAV/2 mod_ssl/2.2.9 OpenSSL/0.9.8h mod_
autoindex_color PHP/5.2.6
    X-Powered-By: PHP/5.2.6
    Expires: Tue, 23 Jun 2009 12:00:00 GMT
    Cache-Control: no-cache, must-revalidate
    Pragma: no-cache
    Vary: Accept-Encoding
    Content-Length: 4705
    Connection: close
    Content-Type: text/html;charset=utf-8
                                                             //空白行，代表响应头结束
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.
org/TR/xhtml1/DTD/xhtml1-strict.dtd">                        //响应正文
    <html xmlns="http://www.w3.org/1999/xhtml">
```

```
……
</html>
```

（1）HTTP 响应的第一行为响应行，依次是当前 HTTP 版本号、3 位数字组成的状态代码，以及描述状态的短语，彼此由空格分隔。响应行格式：HTTP-Version Status-Code Reason-Phrase。其中，HTTP-Version 表示服务器 HTTP 的版本，Status-Code 表示服务器发回的响应状态代码，Reason-Phrase 表示状态代码的文本描述。

（2）第二行至末尾的空白行为响应头，响应头说明了关于服务器的信息和对 Request-URI 所标识的资源进行下一步访问的信息。常用的响应头如下：

● Server 响应头提供所使用的 Web 服务器软件的相关信息。有时还包括其他信息，如所安装模块和服务器操作系统，但其中包含的信息可能不准确。

● Set-Cookie 响应头用来向浏览器发送一个 Cookie（即执行 Cookie 命令，在客户端上保存 Cookie），它将在以后向服务器发送的请求中由 Cookie 响应头返回给服务器。

● Pragma 响应头指示浏览器不要将响应保存在缓存中。Expires 响应头指出响应内容已经过期，因此不应保存在缓存中。当返回动态内容时常常会发送这些指令，以确保浏览器随时获得最新的内容。

● Content-Type 响应头表示这个消息主体中包括 html 文档及所有编码方式，如 Content-Type: text/html;charset=utf-8。

● Content-Length 响应头说明了消息主体的字节长度。

● Cache-Control 响应头用于向浏览器传送缓存指令。

● Location 响应头用于重定向接受者到一个新的位置（包含以 3 开头的响应码）。Location 响应头常用在更换域名的时候。

● WWW-Authenticate 响应头用在 401（未授权的）响应消息中。当客户端收到 401 响应消息，并发送 Authorization 报头域请求服务器对其进行验证时，服务器端响应报头就包含该响应头。如 WWW-Authenticate:Basic realm="Basic Auth Test!"，可以看出服务器对请求资源采用的是基本验证机制。

（3）响应正文就是服务器返回的资源内容。

2. HTTP 状态码

每条 HTTP 响应都必须在第一行中包含一个状态码，说明请求的结果。状态代码由三位数字组成，第一个数字定义了响应的类别，共有五类：

（1）1xx：信息提示。表示请求已接收，继续处理。其范围为 100～101。

（2）2xx：成功。表示服务器成功地处理了请求。其范围为 200～206。

（3）3xx：重定向。用于告诉浏览器客户端，它们访问的资源将重新对新资源发起请求，这时浏览器将重新对资源发起请求。其范围为 300～305。

（4）4xx：客户端错误。请求有语法错误或请求无法实现，或者请求一个不存在的 URL。其范围为 400～415。

（5）5xx：服务器端错误。服务器未能实现合法的请求，可能是 Web 服务器运行出错或者网站无法正常工作了。其范围为 500～505。

在 Web 安全测试中，常用的状态码如下：

- 200 OK。表示请求被成功提交，且响应主体中包含请求结果。
- 201 Created。PUT 请求的响应返回这个状态码，表示请求被成功提交。
- 301 Moved Permanently。指示浏览器永久重定向到另外一个在 Location 消息头中指定的 URL，以后客户应使用新的 URL 替换原始 URL。
- 302 Found。指示浏览器暂时重定向到另外一个在 Location 消息头中指定的 URL，客户应在随后的请求中恢复使用原始的 URL。
- 304 Not Modified。指示浏览器使用缓存中保存的被请求资源的副本。服务器使用 If-Modified-Since 与 If-None-Match 消息头确定客户是否拥有最新版本的资源。
- 400 Bad Request。表示客户端提交了一个无效的 HTTP 请求，即客户端请求有语法错误，不能被服务器所理解。
- 401 Unauthorized。表示服务器在许可请求前要求 HTTP 验证，需要和 WWW-Authenticate 报头域一起使用。
- 403 Forbidden。表示服务器收到请求，但是拒绝提供服务。
- 404 Not Found。表示被请求资源不存在，例如输入了错误的 URL。
- 405 Method Not Allowed。表示指定的 URL 不支持请求中使用的方法。如果试图在不支持 PUT 方法的地方使用该方法，就会收到本状态码。
- 413 Request Entity Too Large。表示请求主体过长，服务器无法处理。如果在本地代码中探查缓冲器溢出漏洞并就此提交超长的字符串，就可能会收到本状态码。
- 414 Request URI Too Large。表示请求中的 URL 过长，服务器无法处理。
- 500 Internal Server Error。表示服务器在执行请求时遇到错误。当提交无法预料的输入、在应用程序处理过程中造成无法处理的错误时，通常会收到本状态码。
- 503 Server Unavailable。表示虽然服务器运转正常，但 Web 应用程序无法做出响应。

1.3.4 HTTPS

HTTP 是非面向连接的协议，且通信使用明文，请求和响应不会对通信方进行确认，无法保护数据的机密性与完整性。为了满足机密性与完整性的要求，HTTPS 应运而生。HTTPS 是身披 SSL 外壳的 HTTP。HTTPS 是一种通过计算机网络进行安全通信的传输协议，经由 HTTP 进行通信，利用 SSL/TLS 建立全信道，加密数据包。HTTPS 使用的主要目的是提供对网站服务器的身份认证，同时保护交换数据的隐私与完整性。HTTPS 默认采用 443 端口，具有如下特点。

- 内容加密：采用混合加密技术，中间者无法直接查看明文内容。
- 验证身份：通过证书认证客户端访问的是自己的服务器。
- 保护数据完整性：防止传输的内容被中间人冒充或者篡改。

1.3.5　实训：抓取并分析 HTTP 数据包

实训目的

1. 安装 Burp Suite 工具。
2. 掌握 Burp Suite 工具代理的使用。
3. 能利用 Burp Suite 工具抓取 HTTP 数据包，并分析其报头组成及相关响应头的含义。

实训原理

Burp Suite 是用于攻击 Web 应用程序的集成平台，包含了许多工具。Burp Suite 为这些工具设计了许多接口，以加快攻击应用程序的过程。所有工具都共享一个请求，并能处理对应的 HTTP 消息、持久性、认证、代理、日志、警报。其主要功能模块包括 Proxy、Spider、Scanner、Intruder 等，其中 Proxy 是一个拦截 HTTP/HTTPS 的代理服务器，作为一个在浏览器和目标应用程序之间的中间人，允许拦截、查看、修改在两个方向上的原始数据流。在本实训中，我们主要利用 Proxy 实现对数据包的抓取。

实训步骤

步骤 1：安装与启动 Burp Suite

Burp Suite 有 Professional 和 Community 两个版本，Community 是免费的版本，能满足基本需要。其是一个跨平台的软件，有 Windows、Linux、Mac OS X 等多个版本。本实训使用 burpsuite_community_windows-x64，在 64 位 Windows 操作系统下安装使用。

1. 安装 Burp Suite

安装过程非常简单，只要双击安装程序，会出现如图 1-9 所示的安装向导。

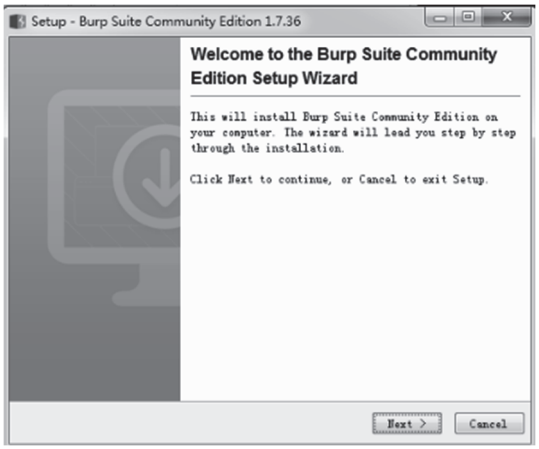

图 1-9　Burp Suite 安装向导

单击“Next”按钮，并且在接下来的窗口中一直单击“Next”按钮就可以完成安装。

2. 启动 Burp Suite

启动 Burp Suite 非常简单，双击其快捷方式就可以进入启动向导，单击“Next”按钮，进入启动界面，在启动界面单击“StartBurp”按钮之后，就正式启动并进入 Burp Suite 应用程序界面，如图 1-10 所示。

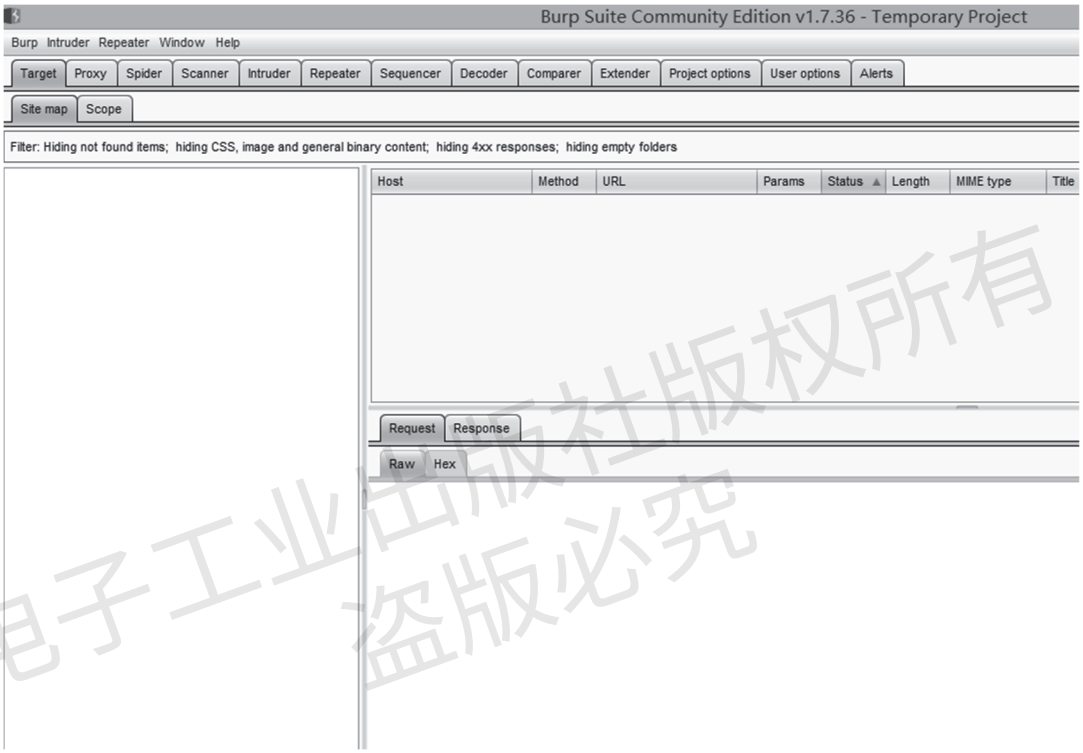

图 1-10　Burp Suite 应用程序界面

单击各个选项卡就会进入相应的功能模块，如接下来要使用 Proxy（代理）功能模块，则单击“Proxy”选项卡即可。

● Proxy 代理设置

Proxy 代理的工作原理是：当客户在浏览器中设置好 Proxy Server 后，客户使用浏览器访问所有 WWW 站点的请求都不会直接发给目的主机，而是先发给代理服务器，代理服务器接受了客户的请求以后，由代理服务器向目的主机发出请求，并接受目的主机的数据，存于代理服务器的硬盘中，然后再由代理服务器将客户要求的数据发给客户。因此需要在客户端浏览器和代理服务器上进行设置才能真正使代理服务器起作用。

步骤 2：在 Burp Suite 上配置网络代理

（1）启动 Burp Suite 之后，选择“Proxy”选项卡，然后选择相应界面下“Options”选项卡，进入网络代理配置界面，如图 1-11 所示。

图 1-11　Burp Suite Proxy Listeners 配置

（2）进入网络配置界面之后，在“Proxy Listeners”选项下，单击“Add”按钮，进入“Add a new proxy listener”界面，如图 1-12 所示。

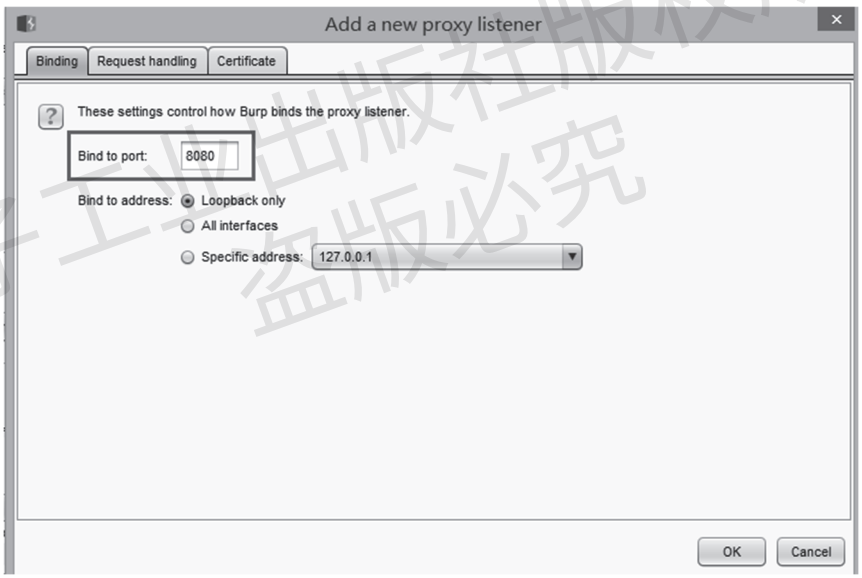

图 1-12　Burp Suite Proxy Listeners 配置服务端口

在此界面下，只要在“Bind to port”处指定端口，此处用 8080 端口，其他采用默认设置即可，单击“OK”按钮即完成添加 proxy listener 的任务。

（3）根据需要修改网络代理配置。如果要修改配置，只要单击“Edit”按钮即可进入与添加 proxy listener 类似的界面，然后进行修改即可。

步骤 3：在客户浏览器上设置代理（以 Firefox 浏览器为例）

打开 Firefox 浏览器，依次选择“工具”→“选项”→“高级”→“网络”→“设置”命令，进入代理设置界面，如图 1-13 所示。

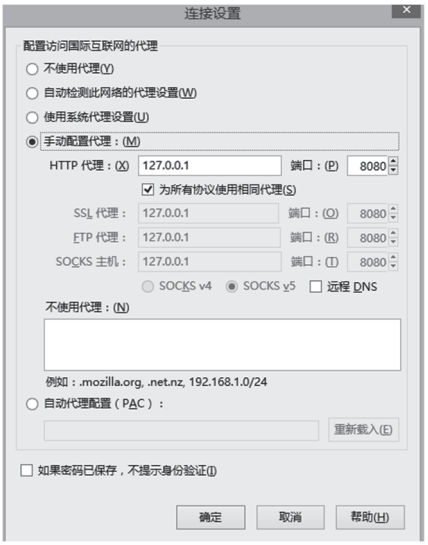

图 1-13　Firefox 浏览器代理配置

选择“手动配置代理”，在“HTTP 代理”框中输入 127.0.0.1。端口号要与 Burp Suite 代理中设置的端口号一致，此处为 8080，单击“确定”按钮，即可完成浏览器的代理设置。以后利用该浏览器上网时，会先发给代理服务器。

- **抓取与分析 HTTP 数据报头**

步骤 4：查看浏览器访问状态

在浏览器的地址栏中输入网站地址，并按回车键，此时浏览器显示“正在连接”，如图 1-14 所示。

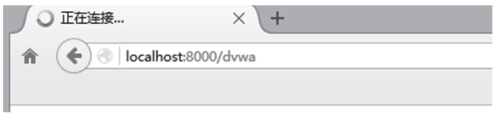

图 1-14　浏览器配置代理后的访问状态

步骤 5：Burp Suite 拦截数据包

在 Burp Suite 的“Proxy”→“Intercept”选项卡下会有请求数据包出现，如图 1-15 所示。

图 1-15　Burp Suite 拦截数据包

单击“Forward”按钮，数据包将被发送到 Web 服务器（单击“Drop”按钮，数据包将被丢弃），根据需要，继续单击“Forward”按钮，直到浏览器端显示正常的网站页面。

步骤 6：查看分析请求包、响应包数据结构

单击 Burp Suite 的“Proxy”→“HTTP history”选项卡，有如图 1-16 所示类似界面。上部显示的是历史浏览记录。如果选择了记录项，相关请求和响应就会在下部的窗口中显示。

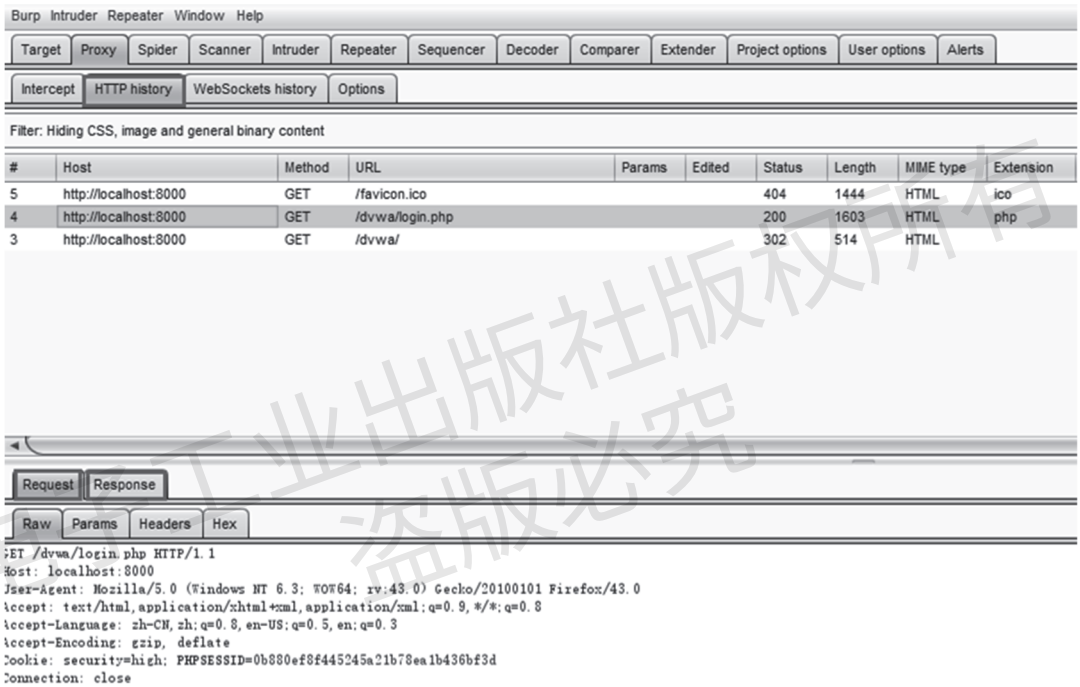

图 1-16　Burp Suite HTTP 历史记录

（1）分析 HTTP 请求报头。单击下部的“Requst”选项卡，可以看到 HTTP 请求报文，如图 1-17 所示。分析 HTTP 请求报文的组成，以及 GET 方法和 HOST、Cookie 等响应头的含义。

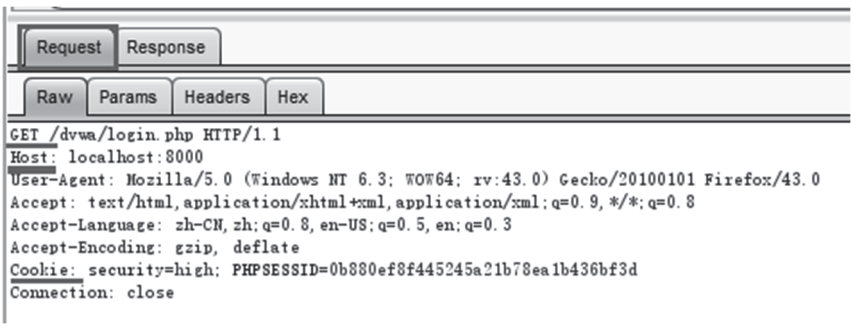

图 1-17　HTTP 请求报文

（2）分析 HTTP 响应报头。单击下部的“Response”选项卡，就会看到 HTTP 响应报文，如图 1-18 所示。分析 HTTP 响应报文的组成，以及状态码、Content-Length 等响应头的含义。

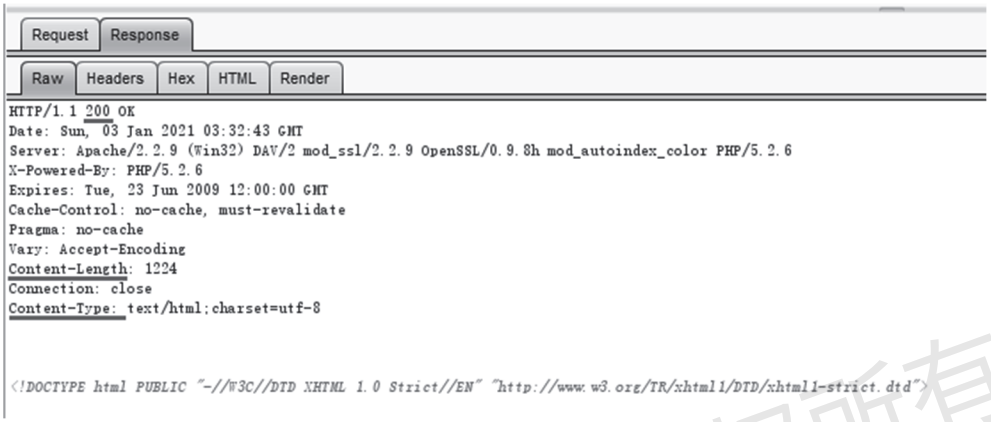

图 1-18　HTTP 响应报文

实训总结

通过实验可以看到：

1. HTTP 遵循请求应答模型，由客户端发起请求，服务器接收到请求之后进行应答。
2. HTTP 请求报文由请求行、请求头（消息头）、请求正文三个部分组成。
3. HTTP 响应报文由响应行、响应头（消息头）、响应正文（消息主体）三个部分组成。
4. Burp Suite 的代理相当于浏览器和目标应用程序之间的中间人，允许拦截、查看、修改在两个方向上的原始数据流。

1.4　Web 系统控制会话技术

服务器与浏览器之间的通信是基于 HTTP 的，是一个包含多次请求和响应的过程。由于 HTTP 协议本身没有状态，虽能够实现网页的访问，但并不能区分访问者的身份，因此需要通过会话控制，实现对访问者的信息进行跟踪、对访问者的状态进行记录。有 Cookie 和 Session 两种技术实现会话控制。

1.4.1　Cookie

Cookie 是 Web 服务器保存在客户端的一系列文本信息，即服务器把每个用户的数据以 Cookie 的形式写给用户各自的浏览器。当用户使用浏览器再去访问服务器中的 Web 资源时，无须用户采取任何措施，随后的请求就会带着各自的 Cookie 数据去访问服务器。这样，Web 服务程序处理的就是各自的用户数据，从而实现对特定对象的追踪。

如前所述，服务器使用 Set-Cookie 响应头发布 Cookie：

```
Set-Cookie: user[xm]=admin; expires=Sun, 27-Dec-2020 09:33:16 GMT
Set-Cookie: user[num]=1; expires=Sun, 27-Dec-2020 09:33:16 GMT
Set-Cookie: user[expire]=7; expires=Sun, 27-Dec-2020 09:33:16 GMT
```

然后，用户的浏览器自动将下面的消息头添加到随后返回给同一服务器的请求中：

```
Cookie: PHPSESSID=c2370ae5baba7a3a9beed580b0262c46
```

Cookie 一般以键/值对的形式出现。可以在服务器响应中使用几个 Set-Cookie 响应头发布多个消息，并可在同一个 Cookie 响应头中用分号分隔不同的 Cookie，将它们全部返回给服务器。

除 Cookie 的实际值外，Set-Cookie 响应头还可包含以下可选属性，用以处理控制浏览器处理 Cookie 的方式。

- expire。用于设置 Cookie 的有效期，是一个 UNIX 时间戳，单位为秒。如果没有设定这个属性，Cookie 仅保存默认的存储时间。
- path。用于指定 Cookie 的有效 URL 路径，表示该路径下的网页或者程序可以有权限进行 Cookie 的存取。
- domain。用于指定 Cookie 的有效域。这个域必须和收到的 Cookie 的域相同，或者是它的父域。
- secure。指定 Cookie 是否通过安全的 HTTPS 连接传送，值为 0 表示 HTTP 和 HTTPS 都可以安全传送，值为 1 则表示只在 HTTPS 连接上有效。
- HttpOnly。如果设置这个属性，则无法通过客户端 JavaScript 直接访问 Cookie，但并非所有的浏览器都支持这一限制。

1.4.2　Session

Session 的中文是“会话”的意思，代表了服务器与客户端之间的“会话”，意思是服务器与客户端在不断地交流。如果不使用 Session，则客户端的每一次请求都是独立存在的，当服务器完成某次用户的请求后，服务器将不能再继续保持与该用户浏览器的连接；当用户在网站的多个页面间切换时，页面之间无法传递用户的相关信息。从多站的角度看，用户每一次新的请求都是独立存在的。引入了 Session 概念，只要把用户的信息存储在 Session 变量中，其信息就不会丢失，而是在整个会话过程中一直存下去。

Session 将用户的信息存储在服务器端，以类似散列表的方式保存信息，其通过 SessionID 判断是否已经创建 Session，创建 Session 后，将 SessionID 返回客户端保存。

1.4.3　Cookie 与 Session 的比较

Cookie 与 Session 都能存储和跟踪特定用户的信息，但二者既相互联系，也存在很多不同之处。

- Session 是在服务器端保存用户信息，Cookie 是在客户端保存用户信息。

- Cookie 的数据大小是有限制的，每个 Cookie 文件不超过 4KB，每个站点最多只能设置 20 个 Cookie。
- Session 仍然要通过 Cookie 实现，因为用户的 SessionID 必须保存在会话 Cookie 中。
- Session 中保存的是对象，Cookie 保存的是字符串。
- Session 随会话结束而关闭，Cookie 可以长期保存在客户端。
- Cookie 可能会泄露隐私，通常用于保存不重要的用户信息。

1.4.4 实训：利用 Cookie 冒充他人登录系统

实训目的

1. 理解 Cookie 的作用。
2. 掌握利用 Cookie 的方法。

实训原理

服务器把每个用户的数据以 Cookie 的形式写给用户各自的浏览器。当用户使用浏览器再去访问服务器中的 Web 资源时，其随后请求就会带着各自的 Cookie 数据去访问服务器，通过这种措施，Web 服务程序处理的就是各自的用户数据，从而实现对特定对象的追踪。

在本实训中，我们使用 Firefox 和 Chrome 两个浏览器，虽然它们在同一台计算机上，且访问相同的地址，但它们属于不同的应用，它们之间的 Cookie 是不能互访的，因此可以模拟窃取 Cookie 冒充会话。

实训步骤

步骤 1：清除 Firefox 中的针对 DVWA 系统的 Cookie

（1）在 Firefox 浏览器工具栏找到并选择“工具”→“选项”→“隐私与安全”命令。

（2）选择“移除特定网的 cookie”选项，然后选择 DVWA 系统，即可删除 DVWA 系统对应的 Cookie。

步骤 2：观察 HTTP 请求与响应报文

使用 Firefox 浏览器，通过 Burp Suite 代理登录 DVWA 系统，观察 HTTP 请求与响应报文。

首次登录的请求与响应报文如图 1-19 所示。可以看到，在 HTTP 响应报文中，服务器给客户端发送了 Set-Cookie 命令，通过浏览器在客户端处记录了客户的 Cookie 值。

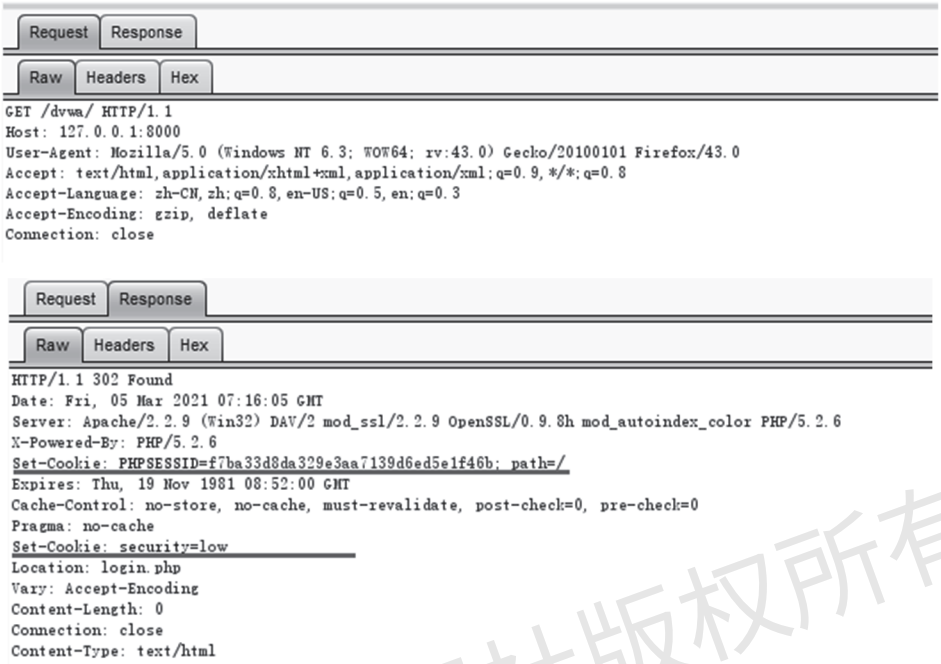

图 1-19　DVWA 登录的请求及响应报文

随后的 HTTP 请求及响应报文如图 1-20 所示。

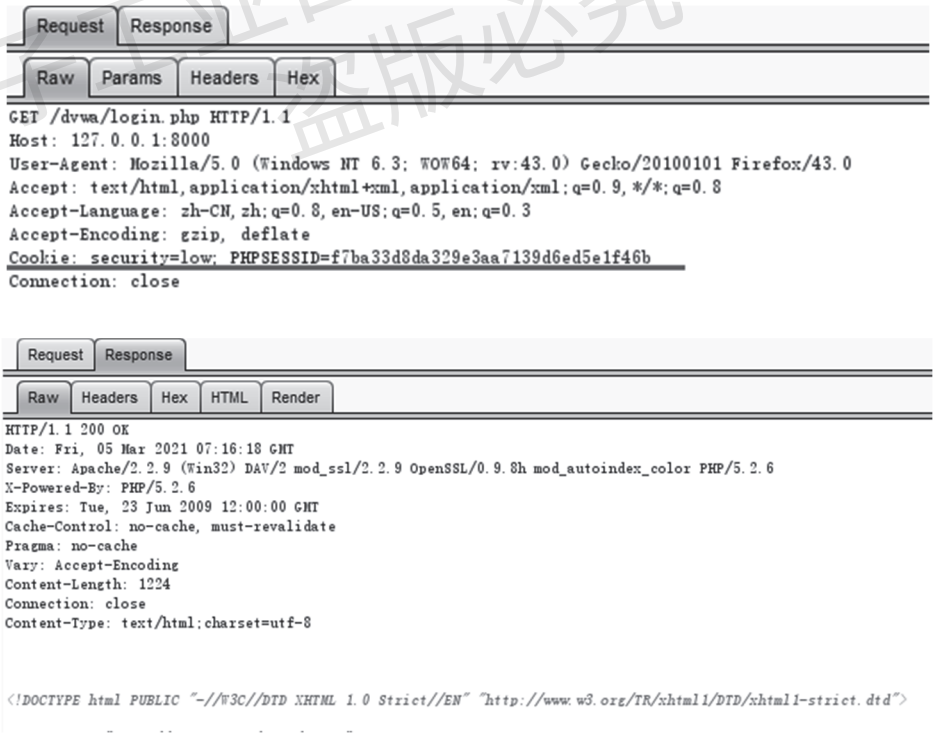

图 1-20　DVWA 登录之后的请求及响应报文

在随后的请求报文中，都自动通过 Cookie 参数把服务器写到客户端处的值传递给服务器。

步骤 3：修改请求报文的 Cookie 值，验证其作用

登录之后，单击左边导航栏的其中一个菜单。在 Burp Suite 中看到请求报文，如图 1-21 所示。

图 1-21　DVWA 系统的 sqli 模块的请求报文

修改 Cookie 值，把 PHPSESSID 的最后一位修改为其他值，如 0。单击“Forward”按钮，在浏览器端一直显示“正在连接”，如图 1-22 所示。意味着修改 Cookie 值之后，不能成功登录，说明 Cookie 可用来进行身份验证。

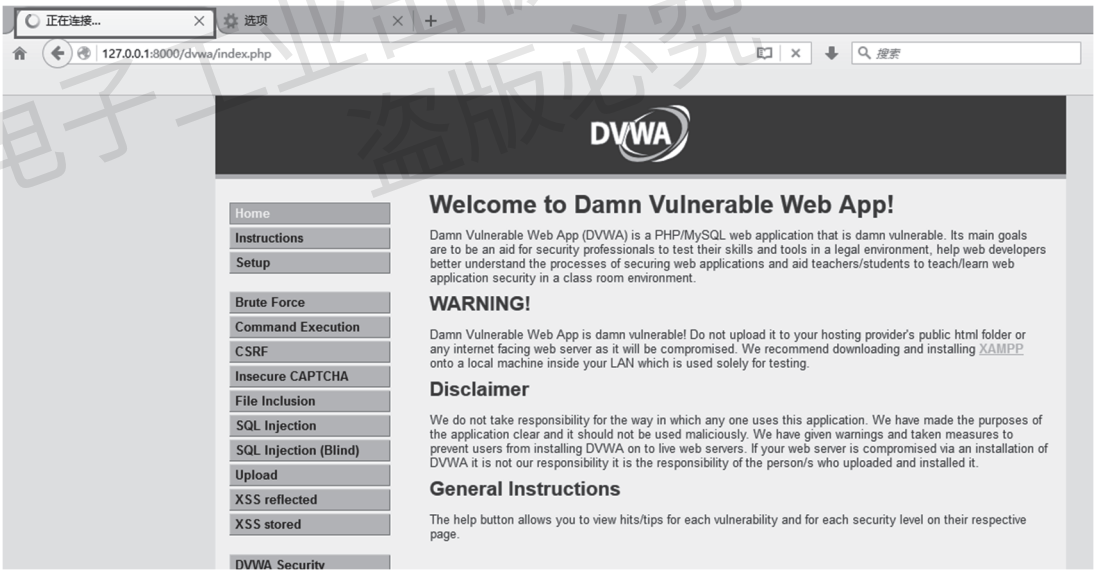

图 1-22　修改 Cookie 值之后访问 DVWA 系统的状态

步骤 4：模拟窃取 Cookie 冒充登录

通过 Chrome 浏览器登录 DVWA 系统，查看 Cookie 值，复制该 Cookie 的 PHPSESSID 值，如图 1-23 所示。

图 1-23　复制 Chrome 浏览器访问 DVWA 系统的 Cookie 值

步骤 5：修改 Firefox 浏览器请求报文中的 Cookie 值为 Chrome 浏览器中的 Cookie 值

再在 Firefox 浏览器中单击其中一个菜单，在 Burp Suite 中将会出现请求包，将其中的 PHPSESSID 值更换为在 Chrome 浏览器中复制的值，如图 1-24 所示。

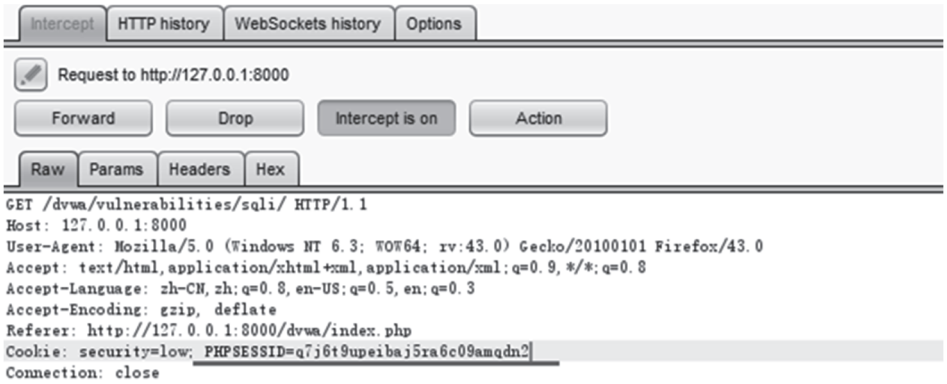

图 1-24　修改 Firefox 浏览器访问 DVWA 系统的 Cookie 值

再单击“Forward”按钮，将成功登录系统，即实现了冒用系统通过 Chrome 浏览器在客户端中写入 Cookie 值。

实训总结

通过实验可以看到：

1. 通过 Cookie 实现对特定对象的追踪，可用于身份验证。
2. 可利用获取的 Cookie 实现身份假冒登录系统。

练习题

一、填空题

1. Web 系统通信的核心协议是（　　），它是轻量级的，无需连接，即非面向连接的协议。

2. Web 系统采用（　　）架构，即提供服务的一端为服务器端，而客户端采用浏览器进行访问。

3. Web 系统主要由（　　）与客户端两部分组成。

4. HTTP 采用（　　）模型进行交互。

5. 状态码（　　）代表客户端请求成功，是最常见的状态。

6. HTTP 请求由请求行、（　　）、请求正文三个部分组成。

7. HTTP 响应由响应行、（　　）、响应正文（消息主体）三个部分组成。

8. 入侵者在渗透服务器端时可从操作系统、系统服务及（　　）三个层面进行攻击。

9. 如果 XAMPP 平台的安装目录是“D:\XAMPP”，则其默认网站根目录为（　　）。

二、选择题

1. HTTP 默认的端口是（　　）。

A. 80　　B. 443　　C. 23　　D. 8000

2. 以下属于 Web 客户端技术的是（　　）。

A. Ngix　　B. 数据库　　C. PHP　　D. JavaScript

3. 以下哪种不是关系型数据库（　　）。

A. SQL Server　　B. Linux　　C. Oracle　　D. MySQL

4. 以下哪个不是 Web 容器（　　）。

A. IIS　　B. Apache　　C. Oracle　　D. Ngix

5. 以下哪个不是 HTTP 请求包含的部分（　　）。

A. 请求行　　B. 请求头　　C. 请求正文　　D. 请求字节

6. 以下哪个不是 HTTP 响应的组成部分（　　）。

A. 响应行　　B. 响应方法　　C. 响应正文　　D. 响应头

7. 客户端浏览器与 Web 服务器进行交互时采用（　　）协议。

A. FTP　　B. TELNET　　C. HTTP　　D. RDP

8. HTTP 响应的响应行包括 HTTP 版本、（　　）及消息 OK。

A. 状态码　　B. 响应方法　　C. 响应正文　　D. 响应长度

9. 以下哪个不是 HTTP 的请求方法（　　）。

A. TELNET　　B. GET　　C. POST　　D. PUT

10. Burp Suite 的哪个模块可以拦截、查看、修改 HTTP 交互的数据报文（　　）。

A. Scanner　　B. Spider　　C. Proxy　　D. Intruder

三、简答题

1. Web 系统服务器端的主要作用是什么？
2. 简要介绍 Web 服务器所采用的技术。
3. 简要解释 HTTP。
4. HTTP 与 HTTPS 主要有哪些区别？
5. 简述 HTTP 状态码的五种类别。
6. 为什么有时候登录网站时看到 404 错误？
7. GET 方法与 POST 方法有哪些区别？
8. Session 与 Cookie 有哪些异同？
9. 简要介绍代理服务器的工作原理。
10. 结合 OWASP 十大 Web 应用漏洞，简要介绍自己对 Web 应用漏洞的理解。

四、CTF 练习

将源程序中 CTF1.zip 文件拷贝到 XAMPP 的 htdocs 文件夹，并解压到该文件夹中的 CTF1 文件夹。

1. 访问 http://127.0.0.1/ctf1/index.html，夺取 flag。
2. 访问 http://127.0.0.1/ctf1/index1.html，根据提示，夺取 flag。