

<<黑客攻防技术宝典（第2版）>>

图书基本信息

书名：<<黑客攻防技术宝典（第2版）>>

13位ISBN编号：9787115283924

10位ISBN编号：7115283923

出版时间：2012-7

出版时间：人民邮电出版社

作者：[英] Dafydd Stuttard,[英] Marcus Pinto

页数：626

字数：957000

译者：石华耀,傅志红

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<黑客攻防技术宝典 (第2版) >>

内容概要

《黑客攻防技术宝典：Web实战篇(第2版)》是探索和研究Web应用程序安全漏洞的实践指南。作者利用大量的实际案例和示例代码，详细介绍了各类Web应用程序的弱点，并深入阐述了如何针对Web应用程序进行具体的渗透测试。

《黑客攻防技术宝典：Web实战篇(第2版)》从介绍当前Web应用程序安全概况开始，重点讨论渗透测试时使用的详细步骤和技巧，最后总结书中涵盖的主题。每章后还附有习题，便于读者巩固所学内容。

第2版新增了Web应用程序安全领域近年来的发展变化新情况，并以尝试访问的链接形式提供了几百个互动式“漏洞实验室”，便于读者迅速掌握各种攻防知识与技能。

《黑客攻防技术宝典：Web实战篇(第2版)》适合各层次计算机安全人士和Web开发与管理领域的技术人员阅读。

作者简介

Dafydd Stuttard 世界知名安全顾问、作家、软件开发人士。
牛津大学博士，MDSec公司联合创始人，尤其擅长Web应用程序和编译软件的渗透测试。
Dafydd以网名PortSwigger蜚声安全界，是众所周知的Web应用程序集成攻击平台Burp Suite的开发者。

Marcus Pinto 资深渗透测试专家，剑桥大学硕士，MDSec公司联合创始人。
Marcus为全球金融、政府、电信、博彩、零售等行业顶尖组织和机构提供Web应用程序渗透测试和安全防御的咨询与培训。

书籍目录

目 录

第1章 Web应用程序安全与风险	1
1.1 Web应用程序的发展历程	1
1.1.1 Web应用程序的常见功能	3
1.1.2 Web应用程序的优点	4
1.2 Web应用程序安全	4
1.2.1 “本站点是安全的”	5
1.2.2 核心安全问题：用户可提交任意输入	6
1.2.3 关键问题因素	7
1.2.4 新的安全边界	8
1.2.5 Web应用程序安全的未来	10
1.3 小结	10
第2章 核心防御机制	12
2.1 处理用户访问	12
2.1.1 身份验证	13
2.1.2 会话管理	13
2.1.3 访问控制	14
2.2 处理用户输入	15
2.2.1 输入的多样性	15
2.2.2 输入处理方法	16
2.2.3 边界确认	18
2.2.4 多步确认与规范化	20
2.3 处理攻击者	21
2.3.1 处理错误	21
2.3.2 维护审计日志	22
2.3.3 向管理员发出警报	23
2.3.4 应对攻击	24
2.4 管理应用程序	25
2.5 小结	26
2.6 问题	26
第3章 Web应用程序技术	27
3.1 HTTP	27
3.1.1 HTTP请求	27
3.1.2 HTTP响应	28
3.1.3 HTTP方法	29
3.1.4 URL	30
3.1.5 REST	31
3.1.6 HTTP消息头	31
3.1.7 cookie	33
3.1.8 状态码	33
3.1.9 HTTPS	34
3.1.10 HTTP代理	35
3.1.11 HTTP身份验证	35
3.2 Web功能	36
3.2.1 服务器端功能	36

- 3.2.2 客户端功能 40
- 3.2.3 状态与会话 46
- 3.3 编码方案 47
 - 3.3.1 URL编码 47
 - 3.3.2 Unicode编码 48
 - 3.3.3 HTML编码 48
 - 3.3.4 Base64编码 49
 - 3.3.5 十六进制编码 49
 - 3.3.6 远程和序列化框架 49
- 3.4 下一步 50
- 3.5 问题 50
- 第4章 解析应用程序 51
 - 4.1 枚举内容与功能 51
 - 4.1.1 Web抓取 51
 - 4.1.2 用户指定的抓取 54
 - 4.1.3 发现隐藏的内容 56
 - 4.1.4 应用程序页面与功能路径 67
 - 4.1.5 发现隐藏的参数 69
 - 4.2 分析应用程序 69
 - 4.2.1 确定用户输入入口点 70
 - 4.2.2 确定服务器端技术 72
 - 4.2.3 确定服务器端功能 76
 - 4.2.4 解析受攻击面 79
 - 4.2.5 解析Extreme Internet Shopping应用程序 80
 - 4.3 小结 81
 - 4.4 问题 82
- 第5章 避开客户端控件 83
 - 5.1 通过客户端传送数据 83
 - 5.1.1 隐藏表单字段 84
 - 5.1.2 HTTP cookie 86
 - 5.1.3 URL参数 86
 - 5.1.4 Referer消息头 86
 - 5.1.5 模糊数据 88
 - 5.1.6 ASP.NET ViewState 89
 - 5.2 收集用户数据：HTML表单 91
 - 5.2.1 长度限制 91
 - 5.2.2 基于脚本的确认 93
 - 5.2.3 禁用的元素 94
 - 5.3 收集用户数据：浏览器扩展 95
 - 5.3.1 常见的浏览器扩展技术 96
 - 5.3.2 攻击浏览器扩展的方法 97
 - 5.3.3 拦截浏览器扩展的流量 97
 - 5.3.4 反编译浏览器扩展 100
 - 5.3.5 附加调试器 109
 - 5.3.6 本地客户端组件 111
 - 5.4 安全处理客户端数据 112
 - 5.4.1 通过客户端传送数据 112

- 5.4.2 确认客户端生成的数据 112
- 5.4.3 日志与警报 113
- 5.5 小结 114
- 5.6 问题 114
- 第6章 攻击验证机制 115
 - 6.1 验证技术 115
 - 6.2 验证机制设计缺陷 116
 - 6.2.1 密码保密性不强 116
 - 6.2.2 蛮力攻击登录 117
 - 6.2.3 详细的失败消息 120
 - 6.2.4 证书传输易受攻击 122
 - 6.2.5 密码修改功能 124
 - 6.2.6 忘记密码功能 125
 - 6.2.7 “记住我”功能 127
 - 6.2.8 用户伪装功能 129
 - 6.2.9 证书确认不完善 131
 - 6.2.10 非唯一性用户名 131
 - 6.2.11 可预测的用户名 132
 - 6.2.12 可预测的初始密码 133
 - 6.2.13 证书分配不安全 133
 - 6.3 验证机制执行缺陷 134
 - 6.3.1 故障开放登录机制 134
 - 6.3.2 多阶段登录机制中的缺陷 135
 - 6.3.3 不安全的证书存储 138
 - 6.4 保障验证机制的安全 139
 - 6.4.1 使用可靠的证书 140
 - 6.4.2 安全处理证书 140
 - 6.4.3 正确确认证书 141
 - 6.4.4 防止信息泄露 142
 - 6.4.5 防止蛮力攻击 143
 - 6.4.6 防止滥用密码修改功能 144
 - 6.4.7 防止滥用账户恢复功能 145
 - 6.4.8 日志、监控与通知 146
 - 6.5 小结 146
 - 6.6 问题 147
- 第7章 攻击会话管理 148
 - 7.1 状态要求 148
 - 7.2 会话令牌生成过程中的薄弱环节 151
 - 7.2.1 令牌有一定含义 152
 - 7.2.2 令牌可预测 153
 - 7.2.3 加密令牌 162
 - 7.3 会话令牌处理中的薄弱环节 170
 - 7.3.1 在网络上泄露令牌 170
 - 7.3.2 在日志中泄露令牌 173
 - 7.3.3 令牌—会话映射易受攻击 175
 - 7.3.4 会话终止易受攻击 176
 - 7.3.5 客户端暴露在令牌劫持风险之中 177

- 7.3.6 宽泛的cookie范围 178
- 7.4 保障会话管理的安全 180
 - 7.4.1 生成强大的令牌 181
 - 7.4.2 在整个生命周期保障令牌的安全 182
 - 7.4.3 日志、监控与警报 184
- 7.5 小结 185
- 7.6 问题 185
- 第8章 攻击访问控制 187
 - 8.1 常见漏洞 187
 - 8.1.1 完全不受保护的功能 188
 - 8.1.2 基于标识符的功能 190
 - 8.1.3 多阶段功能 191
 - 8.1.4 静态文件 191
 - 8.1.5 平台配置错误 192
 - 8.1.6 访问控制方法不安全 192
 - 8.2 攻击访问控制 193
 - 8.2.1 使用不同用户账户进行测试 194
 - 8.2.2 测试多阶段过程 197
 - 8.2.3 通过有限访问权限进行测试 199
 - 8.2.4 测试“直接访问方法” 201
 - 8.2.5 测试对静态资源的控制 202
 - 8.2.6 测试对HTTP方法实施的限制 202
 - 8.3 保障访问控制的安全 203
 - 8.4 小结 206
 - 8.5 问题 207
- 第9章 攻击数据存储区 208
 - 9.1 注入解释型语言 208
 - 9.2 注入SQL 210
 - 9.2.1 利用一个基本的漏洞 211
 - 9.2.2 注入不同的语句类型 213
 - 9.2.3 查明SQL注入漏洞 216
 - 9.2.4 “指纹”识别数据库 219
 - 9.2.5 UNION操作符 220
 - 9.2.6 提取有用的数据 224
 - 9.2.7 使用UNION提取数据 224
 - 9.2.8 避开过滤 226
 - 9.2.9 二阶SQL注入 227
 - 9.2.10 高级利用 229
 - 9.2.11 SQL注入之外：扩大数据库攻击范围 236
 - 9.2.12 使用SQL注入工具 238
 - 9.2.13 SQL语法与错误参考 241
 - 9.2.14 防止SQL注入 246
 - 9.3 注入NoSQL 249
 - 9.4 注入XPath 250
 - 9.4.1 破坏应用程序逻辑 251
 - 9.4.2 谨慎XPath注入 252
 - 9.4.3 盲目XPath注入 252

- 9.4.4 查找XPath注入漏洞 253
- 9.4.5 防止XPath注入 254
- 9.5 注入LDAP 254
 - 9.5.1 利用LDAP注入 255
 - 9.5.2 查找LDAP注入漏洞 257
 - 9.5.3 防止LDAP注入 258
- 9.6 小结 258
- 9.7 问题 258
- 第10章 测试后端组件 260
 - 10.1 注入操作系统命令 260
 - 10.1.1 例1:通过Perl注入 261
 - 10.1.2 例2:通过ASP注入 262
 - 10.1.3 通过动态执行注入 264
 - 10.1.4 查找OS命令注入漏洞 264
 - 10.1.5 查找动态执行漏洞 267
 - 10.1.6 防止OS命令注入 268
 - 10.1.7 防止脚本注入漏洞 268
 - 10.2 操作文件路径 268
 - 10.2.1 路径遍历漏洞 269
 - 10.2.2 文件包含漏洞 278
 - 10.3 注入XML解释器 279
 - 10.3.1 注入XML外部实体 279
 - 10.3.2 注入SOAP 281
 - 10.3.3 查找并利用SOAP注入 283
 - 10.3.4 防止SOAP注入 284
 - 10.4 注入后端HTTP请求 284
 - 10.4.1 服务器端HTTP重定向 285
 - 10.4.2 HTTP参数注入 287
 - 10.5 注入电子邮件 290
 - 10.5.1 操纵电子邮件标头 290
 - 10.5.2 SMTP命令注入 291
 - 10.5.3 查找SMTP注入漏洞 292
 - 10.5.4 防止SMTP注入 293
 - 10.6 小结 294
 - 10.7 问题 294
- 第11章 攻击应用程序逻辑 296
 - 11.1 逻辑缺陷的本质 296
 - 11.2 现实中的逻辑缺陷 297
 - 11.2.1 例1:征求提示 297
 - 11.2.2 例2:欺骗密码修改功能 298
 - 11.2.3 例3:直接结算 299
 - 11.2.4 例4:修改保险单 300
 - 11.2.5 例5:入侵银行 302
 - 11.2.6 例6:规避交易限制 303
 - 11.2.7 例7:获得大幅折扣 305
 - 11.2.8 例8:避免转义 305
 - 11.2.9 例9:避开输入确认 306

<<黑客攻防技术宝典 (第2版) >>

- 11.2.10 例10：滥用搜索功能 308
- 11.2.11 例11：利用调试消息 310
- 11.2.12 例12：与登录机制竞赛 311
- 11.3 避免逻辑缺陷 312
- 11.4 小结 313
- 11.5 问题 314
- 第12章 攻击其他用户 315
 - 12.1 XSS的分类 316
 - 12.1.1 反射型XSS漏洞 316
 - 12.1.2 保存型XSS漏洞 320
 - 12.1.3 基于DOM的XSS漏洞 322
 - 12.2 进行中的XSS攻击 323
 - 12.2.1 真实XSS攻击 323
 - 12.2.2 XSS攻击有效载荷 324
 - 12.2.3 XSS攻击的传送机制 327
 - 12.3 查找并利用XSS漏洞 329
 - 12.3.1 查找并利用反射型XSS漏洞 331
 - 12.3.2 查找并利用保存型XSS漏洞 352
 - 12.3.3 查找并利用基于DOM的XSS漏洞 357
 - 12.4 防止XSS攻击 360
 - 12.4.1 防止反射型与保存型XSS漏洞 360
 - 12.4.2 防止基于DOM的XSS漏洞 364
 - 12.5 小结 365
 - 12.6 问题 365
- 第13章 攻击用户：其他技巧 366
 - 13.1 诱使用户执行操作 366
 - 13.1.1 请求伪造 366
 - 13.1.2 UI伪装 374
 - 13.2 跨域捕获数据 377
 - 13.2.1 通过注入HTML捕获数据 377
 - 13.2.2 通过注入CSS捕获数据 378
 - 13.2.3 JavaScript劫持 380
 - 13.3 同源策略深入讨论 384
 - 13.3.1 同源策略与浏览器扩展 384
 - 13.3.2 同源策略与HTML5 386
 - 13.3.3 通过代理服务应用程序跨域 388
 - 13.4 其他客户端注入攻击 389
 - 13.4.1 HTTP消息头注入 389
 - 13.4.2 cookie注入 393
 - 13.4.3 开放式重定向漏洞 396
 - 13.4.4 客户端SQL注入 402
 - 13.4.5 客户端HTTP参数污染 402
 - 13.5 本地隐私攻击 403
 - 13.5.1 持久性cookie 404
 - 13.5.2 缓存Web内容 404
 - 13.5.3 浏览历史记录 405
 - 13.5.4 自动完成 406

<<黑客攻防技术宝典 (第2版) >>

- 13.5.5 Flash本地共享对象 406
- 13.5.6 Silverlight独立存储 406
- 13.5.7 Internet Explorer userData 407
- 13.5.8 HTML5本地存储机制 407
- 13.5.9 防止本地隐私攻击 407
- 13.6 攻击ActiveX控件 408
 - 13.6.1 查找ActiveX漏洞 409
 - 13.6.2 防止ActiveX漏洞 410
- 13.7 攻击浏览器 411
 - 13.7.1 记录键击 411
 - 13.7.2 窃取浏览器历史记录与搜索查询 412
 - 13.7.3 枚举当前使用的应用程序 412
 - 13.7.4 端口扫描 412
 - 13.7.5 攻击其他网络主机 413
 - 13.7.6 利用非HTTP服务 413
 - 13.7.7 利用浏览器漏洞 414
 - 13.7.8 DNS重新绑定 414
 - 13.7.9 浏览器利用框架 415
 - 13.7.10 中间人攻击 416
- 13.8 小结 418
- 13.9 问题 418
- 第14章 定制攻击自动化 419
 - 14.1 应用定制自动化攻击 419
 - 14.2 枚举有效的标识符 420
 - 14.2.1 基本步骤 420
 - 14.2.2 探测“触点” 421
 - 14.2.3 编写攻击脚本 422
 - 14.2.4 JAttack 423
 - 14.3 获取有用的数据 428
 - 14.4 常见漏洞模糊测试 431
 - 14.5 整合全部功能：Burp Intruder 434
 - 14.6 实施自动化的限制 442
 - 14.6.1 会话处理机制 443
 - 14.6.2 CAPTCHA控件 448
 - 14.7 小结 451
 - 14.8 问题 451
- 第15章 利用信息泄露 453
 - 15.1 利用错误消息 453
 - 15.1.1 错误消息脚本 453
 - 15.1.2 栈追踪 454
 - 15.1.3 详尽的调试消息 455
 - 15.1.4 服务器与数据库消息 456
 - 15.1.5 使用公共信息 458
 - 15.1.6 制造详尽的错误消息 459
 - 15.2 收集公布的信息 460
 - 15.3 使用推论 461
 - 15.4 防止信息泄露 462

<<黑客攻防技术宝典 (第2版) >>

- 15.4.1 使用常规错误消息 462
- 15.4.2 保护敏感信息 462
- 15.4.3 尽量减少客户端信息泄露 463
- 15.5 小结 463
- 15.6 问题 463
- 第16章 攻击本地编译型应用程序 466
 - 16.1 缓冲区溢出漏洞 467
 - 16.1.1 栈溢出 467
 - 16.1.2 堆溢出 467
 - 16.1.3 “一位偏移”漏洞 468
 - 16.1.4 查找缓冲区溢出漏洞 470
 - 16.2 整数漏洞 472
 - 16.2.1 整数溢出 472
 - 16.2.2 符号错误 472
 - 16.2.3 查找整数漏洞 473
 - 16.3 格式化字符串漏洞 474
 - 16.4 小结 475
 - 16.5 问题 475
- 第17章 攻击应用程序架构 477
 - 17.1 分层架构 477
 - 17.1.1 攻击分层架构 478
 - 17.1.2 保障分层架构的安全 482
 - 17.2 共享主机与应用程序服务提供商 483
 - 17.2.1 虚拟主机 484
 - 17.2.2 共享的应用程序服务 484
 - 17.2.3 攻击共享环境 485
 - 17.2.4 保障共享环境的安全 490
 - 17.3 小结 491
 - 17.4 问题 491
- 第18章 攻击Web服务器 493
 - 18.1 Web服务器配置缺陷 493
 - 18.1.1 默认证书 493
 - 18.1.2 默认内容 494
 - 18.1.3 目录列表 499
 - 18.1.4 WebDAV方法 500
 - 18.1.5 Web服务器作为代理服务器 503
 - 18.1.6 虚拟主机配置缺陷 504
 - 18.1.7 保障Web服务器配置的安全 504
 - 18.2 易受攻击的服务器软件 505
 - 18.2.1 应用程序框架缺陷 505
 - 18.2.2 内存管理漏洞 507
 - 18.2.3 编码与规范化漏洞 508
 - 18.2.4 查找Web服务器漏洞 512
 - 18.2.5 保障Web服务器软件的安全 513
 - 18.3 Web应用程序防火墙 514
 - 18.4 小结 515
 - 18.5 问题 516

- 第19章 查找源代码中的漏洞 517
 - 19.1 代码审查方法 517
 - 19.1.1 “黑盒”测试与“白盒”测试 517
 - 19.1.2 代码审查方法 518
 - 19.2 常见漏洞签名 519
 - 19.2.1 跨站点脚本 519
 - 19.2.2 SQL注入 520
 - 19.2.3 路径遍历 520
 - 19.2.4 任意重定向 521
 - 19.2.5 OS命令注入 522
 - 19.2.6 后门密码 522
 - 19.2.7 本地代码漏洞 522
 - 19.2.8 源代码注释 524
 - 19.3 Java平台 524
 - 19.3.1 确定用户提交的数据 524
 - 19.3.2 会话交互 525
 - 19.3.3 潜在危险的API 526
 - 19.3.4 配置Java环境 528
 - 19.4 ASP.NET 529
 - 19.4.1 确定用户提交的数据 529
 - 19.4.2 会话交互 530
 - 19.4.3 潜在危险的API 531
 - 19.4.4 配置ASP.NET环境 533
 - 19.5 PHP 534
 - 19.5.1 确定用户提交的数据 534
 - 19.5.2 会话交互 536
 - 19.5.3 潜在危险的API 536
 - 19.5.4 配置PHP环境 540
 - 19.6 Perl 542
 - 19.6.1 确定用户提交的数据 542
 - 19.6.2 会话交互 543
 - 19.6.3 潜在危险的API 543
 - 19.6.4 配置Perl环境 544
 - 19.7 JavaScript 545
 - 19.8 数据库代码组件 546
 - 19.8.1 SQL注入 546
 - 19.8.2 调用危险的函数 547
 - 19.9 代码浏览工具 547
 - 19.10 小结 548
 - 19.11 问题 548
- 第20章 Web应用程序黑客工具包 550
 - 20.1 Web浏览器 550
 - 20.1.1 Internet Explorer 550
 - 20.1.2 Firefox 551
 - 20.1.3 Chrome 552
 - 20.2 集成测试套件 552
 - 20.2.1 工作原理 553

- 20.2.2 测试工作流程 566
- 20.2.3 拦截代理服务器替代工具 568
- 20.3 独立漏洞扫描器 570
 - 20.3.1 扫描器探测到的漏洞 570
 - 20.3.2 扫描器的内在限制 571
 - 20.3.3 扫描器面临的技术挑战 572
 - 20.3.4 当前产品 574
 - 20.3.5 使用漏洞扫描器 576
- 20.4 其他工具 577
 - 20.4.1 Wikto/Nikto 577
 - 20.4.2 Firebug 577
 - 20.4.3 Hydra 578
 - 20.4.4 定制脚本 578
- 20.5 小结 581
- 第21章 Web应用程序渗透测试方法论 582
 - 21.1 解析应用程序内容 584
 - 21.1.1 搜索可见的内容 584
 - 21.1.2 浏览公共资源 585
 - 21.1.3 发现隐藏的内容 586
 - 21.1.4 查找默认的内容 586
 - 21.1.5 枚举标识符指定的功能 586
 - 21.1.6 调试参数 587
 - 21.2 分析应用程序 587
 - 21.2.1 确定功能 587
 - 21.2.2 确定数据进入点 587
 - 21.2.3 确定所使用的技术 588
 - 21.2.4 解析受攻击面 588
 - 21.3 测试客户端控件 588
 - 21.3.1 通过客户端传送数据 589
 - 21.3.2 客户端输入控件 589
 - 21.3.3 测试浏览器扩展组件 590
 - 21.4 测试验证机制 592
 - 21.4.1 了解验证机制 592
 - 21.4.2 测试密码强度 593
 - 21.4.3 测试用户名枚举 593
 - 21.4.4 测试密码猜测的适应性 593
 - 21.4.5 测试账户恢复功能 594
 - 21.4.6 测试“记住我”功能 594
 - 21.4.7 测试伪装功能 594
 - 21.4.8 测试用户名唯一性 595
 - 21.4.9 测试证书的可预测性 595
 - 21.4.10 检测不安全的证书传输 595
 - 21.4.11 检测不安全的证书分配 596
 - 21.4.12 测试不安全的存储 596
 - 21.4.13 测试逻辑缺陷 596
 - 21.4.14 利用漏洞获取未授权访问 597
 - 21.5 测试会话管理机制 598

- 21.5.1 了解会话管理机制 598
- 21.5.2 测试令牌的含义 599
- 21.5.3 测试令牌的可预测性 599
- 21.5.4 检查不安全的令牌传输 600
- 21.5.5 检查在日志中泄露的令牌 600
- 21.5.6 测试令牌?会话映射 601
- 21.5.7 测试会话终止 601
- 21.5.8 测试会话固定 602
- 21.5.9 检查CSRF 602
- 21.5.10 检查cookie范围 602
- 21.6 测试访问控件 603
 - 21.6.1 了解访问控制要求 603
 - 21.6.2 使用多个账户测试 604
 - 21.6.3 使用有限的权限测试 604
 - 21.6.4 测试不安全的访问控制方法 605
- 21.7 测试基于输入的漏洞 605
 - 21.7.1 模糊测试所有请求参数 605
 - 21.7.2 测试SQL注入 607
 - 21.7.3 测试XSS和其他响应注入 609
 - 21.7.4 测试OS命令注入 611
 - 21.7.5 测试路径遍历 612
 - 21.7.6 测试脚本注入 613
 - 21.7.7 测试文件包含 613
- 21.8 测试特殊功能方面的输入漏洞 613
 - 21.8.1 测试SMTP注入 614
 - 21.8.2 测试本地代码漏洞 614
 - 21.8.3 测试SOAP注入 616
 - 21.8.4 测试LDAP注入 616
 - 21.8.5 测试XPath注入 617
 - 21.8.6 测试后端请求注入 617
 - 21.8.7 测试XXE注入 617
- 21.9 测试逻辑缺陷 618
 - 21.9.1 确定关键的受攻击面 618
 - 21.9.2 测试多阶段过程 618
 - 21.9.3 测试不完整的输入 619
 - 21.9.4 测试信任边界 619
 - 21.9.5 测试交易逻辑 619
- 21.10 测试共享主机漏洞 620
 - 21.10.1 测试共享基础架构之间的隔离 620
 - 21.10.2 测试使用ASP主机的应用程序之间的隔离 620
- 21.11 测试Web服务器漏洞 621
 - 21.11.1 测试默认证书 621
 - 21.11.2 测试默认内容 621
 - 21.11.3 测试危险的HTTP方法 622
 - 21.11.4 测试代理功能 622
 - 21.11.5 测试虚拟主机配置不当 622
 - 21.11.6 测试Web服务器软件漏洞 622

- 21.11.7 测试Web应用程序防火墙 623
- 21.12 其他检查 623
 - 21.12.1 测试基于DOM的攻击 624
 - 21.12.2 测试本地隐私漏洞 624
 - 21.12.3 测试脆弱的SSL加密算法 625
 - 21.12.4 检查同源策略配置 625
- 21.13 检查信息泄露 625

章节摘录

版权页：插图：10.2.2文件包含漏洞 许多脚本语言支持使用包含文件（include file）。这种功能允许开发者把可重复使用的代码插入到单个的文件中，并在需要时将它们包含在特殊功能的代码文件中。

然后，包含文件中的代码被解释，就好像它插入到包含指令的位置一样。

1. 远程文件包含 PHP语言特别容易出现文件包含漏洞，因为它的包含函数接受远程文件路径。

这种缺陷已经成为PHP应用程序中大量漏洞的根源。

以一个向不同位置的人们传送各种内容的应用程序为例。

用户选择他们的位置后，这个信息通过一个请求参数传送给服务器，代码如下：`https://wahh—app.com/main.php?Country=US` 应用程序通过以下方式处理Country参数：这使执行环境加载位于Web服务器文件系统中的US.php文件。

然后，这个文件的内容被复制至Umain.php文件中，并得以执行。

攻击者能够以各种方式利用这种行为，最严重的情况是指定一个外部URL作为包含文件的位置。

PHP包含函数接受这个位置作为输入，接着，执行环境将获取指定的文件并执行其内容。

因此，攻击者能够构建一个包含任意复杂内容的恶意脚本，将其寄存在他控制的web服务器上，并通过易受攻击的应用程序函数调用它然后执行。

例如：2. 本地文件包含 有时，应用程序根据用户可控制的数据加载包含文件，但这时不可能给位于外部服务器上的文件指定URL。

例如，如果用户可控制的数据被提交给ASP函数Server.Execute，那么攻击者就可以执行任意一段ASP脚本，只要这段脚本属于调用这个函数的相同应用程序。

在这种情况下，攻击者仍然可以利用应用程序的行为执行未授权操作。

□在服务器上可能有一些通过正常途径无法访问的文件，例如，任何访问路径 / admin的请求都会被应用程序实施的访问控制阻止。

如果能够将敏感功能包含在一个授权访问的页面中，那么就可以访问那个功能。

□服务器上的一些静态资源也受到同样的保护，无法直接访问。

如果能够将这些文件动态包含在其他应用程序页面中，那么执行环境就会将静态资源的内容复制到它的响应中。

3. 查找文件包含漏洞 任何用户提交的数据项都可能引起文件包含漏洞。

它们经常出现在指定一种语言或一个位置的请求参数中，也常常发生在以参数形式传送服务器端文件名的情况下。

<<黑客攻防技术宝典（第2版）>>

媒体关注与评论

关于黑客攻防技术，没有一本书能比这本书讲解得更为透彻和全面！

——Jason Haddix，惠普公司渗透测试总监 如果你对Web应用程序安全感兴趣，我强烈推荐本书，它实为Web安全人士必读之作。

——Robert Wesley McGrew，McGrew安全公司研究人员 第1版本来就是Web安全领域的扛鼎之作，第2版可谓经典之上的完善，绝对值得拥有！

——Daniel Miessler，安全顾问

编辑推荐

安全技术宝典全新升级 深入剖析，实战演练，使你如饮醍醐

<<黑客攻防技术宝典（第2版）>>

名人推荐

“关于黑客攻防技术，没有一本书能比这本书讲解得更为透彻和全面！”

”——Jason Haddix，惠普公司渗透测试总监 “如果你对Web应用程序安全感兴趣，我强烈推荐本书，它实为Web安全人士必读之作。

”——Robert Wesley McGrew，McGrew安全公司研究人员 “第1版本来就是Web安全领域的扛鼎之作，第2版可谓经典之上的完善，绝对值得拥有！”

”——Daniel Miessler，安全顾问

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>