基于 Web 环境的网络安全攻防技术应用研究

程占斌
中国电信股份有限公司东莞分公司，广东 东莞 523000

DOI: 10.61369/TACS.2026080003

摘　　要： 当下 Web 场景中的安全风险不断增多。不法分子借助 Web 应用存在的漏洞发起攻击，极易引发数据外泄、系统停运等问题。针对 Web 攻防相关技术展开研究，具备实际应用价值。文中梳理了 SQL 注入、跨站脚本、跨站请求伪造等主流攻击方式，同时结合实际场景，总结了输入校验、会话管控、部署 Web 应用防火墙等可行的防护手段。结合应用实践，提出安全加固策略。研究表明，构建多层防护体系能够有效提升 Web 环境安全性，降低攻击成功概率。

关 键 词： Web 安全；网络安全；攻防技术

Research on the Application of Network Security Attack and Defense Technologies Based on Web Environment

Cheng Zhanbin
China Telecom Corporation Limited Dongguan Branch, Dongguan, Guangdong 523000

Abstract: Currently, security risks in Web scenarios are constantly increasing. Malicious actors exploit vulnerabilities in Web applications to launch attacks, which can easily lead to data breaches and system downtime. Therefore, researching Web attack and defense technologies holds significant practical value. This paper reviews mainstream attack methods such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF). Combined with practical scenarios, it summarizes feasible defense measures including input validation, session management, and the deployment of Web Application Firewalls (WAF). Furthermore, based on application practices, security hardening strategies are proposed. The research indicates that building a multi-layered defense system can effectively enhance Web environment security and reduce the probability of successful attacks.

Keywords: Web security; network security; attack and defense technologies

引言

互联网技术持续发展，Web 应用如今已融入各行各业。电商平台、线上办公、社交平台等服务，均依托 Web 架构搭建运行。随之而来的是，各类针对 Web 系统的网络攻击也愈发频发。攻击手法不断翻新，防御技术面临严峻挑战。许多 Web 系统存在安全配置缺陷，容易成为攻击目标。因此，系统研究 Web 环境的攻防技术，分析攻击原理并设计有效防御方案，对于保障网络空间安全具有重要价值。

一、Web 环境安全威胁概述

（一）Web 应用发展现状与安全挑战

Web 技术由静态网页逐步发展为动态交互程序，整体功能变得愈发繁杂。各类企业 Web 系统，会存储和处理海量用户数据与交易信息。不法分子常利用 Web 漏洞发起网络入侵。OWASP 持续公布 Web 十大安全隐患，也体现出网络威胁类型的繁杂程度。安全挑战主要体现在三个方面。第一，Web 应用开发周期短，安全测试不足。第二，第三方组件和开源库引入未知漏洞。第三，用户安全意识薄弱，容易触发钓鱼攻击。Web 环境运行依托 HTTP/HTTPS 协议。协议本身的无状态特性增加了会话管理难度。攻击者通过构造畸形请求绕过服务器检测。云原生和微服务架构的普及扩大了攻击面。容器化部署中，一个容器漏洞可能影响整个集群。因此，认识 Web 环境的安全脆弱点成为研究起点。

（二）常见 Web 安全漏洞类型

Web 安全漏洞按照成因可分为输入验证类、身份认证类、授权访问类、配置管理类等。输入验证类漏洞包括 SQL 注入、命令注入、XSS。这类漏洞因为服务器未对用户输入做严格过滤。身份认证类漏洞表现为弱密码策略、会话标识可预测。授权访问类漏洞涉及越权操作，低权限用户获取高权限功能。配置管理类漏洞包含默认账户、目录列出、调试信息泄露。实际攻击案例中，SQL 注入长期占据高发位置。攻击者在查询参数中插入 SQL 语句

 Copyright © This Work is Licensed under A Commons Attribution-Non Commercial 4.0 International License.

片段，改变原始查询逻辑。XSS 漏洞允许攻击者注入恶意脚本，窃取用户 Cookie 或执行钓鱼操作。CSRF 漏洞诱导已登录用户执行非本意请求。文件上传漏洞利用不安全的文件类型检查，上传 Webshell 后门。每种漏洞类型对应不同的利用场景，需要针对性防御。

二、Web 端攻击技术分析

（一）SQL 注入攻击原理与手法

SQL 注入利用 Web 应用未对用户输入做充分转义或过滤。攻击者在表单字段、URL 参数、Cookie 中注入 SQL 代码。服务器将恶意代码拼接到 SQL 语句后执行，导致数据库信息泄露。注入手法分为联合查询注入、报错注入、布尔盲注、时间盲注。联合查询注入使用 UNION 关键字获取额外数据。攻击者先判断字段数量，再构造查询获取数据库名、表名、字段内容。报错注入利用数据库错误信息反馈。例如通过 extractvalue 函数触发 XML 解析错误，错误信息中携带查询结果。布尔盲注适用于页面不返回直接数据但显示内容变化。攻击者逐字符猜测数据，根据页面差异判断正确与否。时间盲注使用延时函数 sleep 或 benchmark，通过响应时间推断数据内容。SQL 注入的危害性极高。攻击者可以获取管理员密码、用户隐私信息，甚至执行操作系统命令。数据库权限配置不当的情况下，攻击者获得数据库服务器控制权。防止 SQL 注入的根本方法是使用参数化查询或预编译语句 [1]。

（二）跨站脚本攻击 (XSS) 分类与利用

XSS 攻击向 Web 页面注入恶意脚本代码。当其他用户访问该页面时，脚本在浏览器中执行。XSS 分为反射型、存储型、DOM 型三种。反射型 XSS 的恶意脚本附在 URL 参数中。服务器将参数内容直接返回到页面。攻击者诱骗用户点击恶意链接触发脚本。存储型 XSS 将脚本持久保存到服务器数据库或文件系统。任何访问受害页面的用户都会受到攻击。这种类型的危害范围大，传播隐蔽。

DOM 型 XSS 不经过服务器响应，完全由前端 JavaScript 操作 DOM 树引发。攻击者控制 location.hash 或 document.referrer 等输入源，再通过 eval 或 innerHTML 执行恶意代码。XSS 攻击常用伎俩包括窃取 Cookie、记录键盘输入、发起钓鱼弹窗、劫持会话。防御 XSS 需要输出编码，按照上下文对 HTML 实体、JavaScript 字符串、URL 参数分别编码。内容安全策略 (CSP) 限制脚本来源，有效降低 XSS 风险。

（三）跨站请求伪造 (CSRF) 与文件上传漏洞

CSRF 攻击依托用户已登录的浏览器会话开展。攻击者制作恶意页面引导用户点击，浏览器会附带网站 Cookie 发起请求，服务器会判定请求合法。该攻击常针对转账、改密、发布内容等功能接口。常见防护手段有增设随机 Token、校验 Referer 字段、搭配验证码完成二次核验。文件上传漏洞多因服务端检测机制不完善引发。攻击者上传 PHP、JSP 等可执行脚本，文件被解析后便会生成 Webshell 后门。借助后门，攻击者可操控服务器、调取数据。常见绕过方式包括篡改请求类型、使用双后缀、植入图片木马。防护可采用后缀白名单、文件重命名、关闭上传目录执行权限，或将文件存放至独立域名 [2]。

三、Web 安全防御技术研究

（一）输入验证与参数过滤

输入验证是 Web 安全的第一道防线。服务器端对所有用户提交的数据执行合法性检查。验证规则包括数据类型、长度、范围、格式。严格区分数据校验与数据清洗。数据校验拒绝不符合格式的输入，数据清洗转义特殊字符。参数过滤采用黑名单与白名单相结合的策略。黑名单过滤已知危险字符，如引号、分号、尖括号。白名单定义允许的字符集，拒绝任何未匹配内容。

正则表达式常用于格式验证。例如邮箱地址、手机号码、身份证号使用严格正则匹配。对于数字型参数，使用整数类型转换函数强制转换。日期参数使用日期解析函数验证有效性。防御 SQL 注入的输入验证方法包括检测注释符、联合查询关键字等号连接。但输入验证不能完全替代参数化查询，因为攻击者可能使用编码绕过。最佳实践是将输入验证作为深度防御的一环，与参数化查询共同使用 [3]。

（二）身份认证与会话管理

身份认证确认用户身份真实性。安全认证机制要求密码复杂度策略，禁止弱口令。多因素认证增加攻击难度。会话管理负责在 HTTP 无状态环境中维持登录状态。会话标识 (Session ID) 必须具有随机性和不可预测性。会话标识在传输过程中使用 HTTPS 加密，防止中间人窃取。设置合理的会话过期时间。用户关闭浏览器后会话失效，长时间无操作自动注销。

防止会话劫持的措施包括绑定客户端 IP 地址和 User-Agent。服务器检测会话标识的请求来源是否与登录时一致。登录成功后重新生成 Session ID，避免会话固定攻击。敏感操作如修改密码、支付转账要求重新验证身份。使用 JSON Web Token(JWT) 时注意签名算法强度，设置较短的过期时间。禁用 Cookie 的 HttpOnly 属性防止 XSS 窃取，启用 Secure 属性保证 Cookie 仅通过 HTTPS 传输。

（三）Web 应用防火墙 (WAF) 部署

WAF 部署在客户端与 Web 服务器之间，能够分析 HTTP 流量，识别并拦截攻击数据。它主要分为云、软件、硬件三类。云 WAF 借助 DNS 引流，操作与防护流程简便。软件 WAF 可嵌入服务器模块，ModSecurity 为常用实例。硬件 WAF 设于网络入口，运算性能突出。该设备可防御 SQL 注入、XSS、命令注入、路径遍历等主流攻击。

规则匹配易产生误报与漏报。误报会拦截正常访问，漏报则放任攻击流量通行。结合业务场景调整规则敏感度，可完成规则优化。部分 WAF 运用语义分析解析 HTTP 参数，有效减少误报。WAF 对越权访问、验证码暴力破解等业务逻辑漏洞防护能力有限，需搭配其他安全手段使用。漏洞修复期间，其虚拟补丁还可实现临时防护 [4]。

Copyright © This Work is Licensed under A Commons Attribution-Non Commercial 4.0 International License.

四、攻防技术应用实践

（一）安全测试工具应用

安全测试能够排查 Web 系统存在的各类漏洞。常用测试工具主要分为静态分析与动态扫描两类。静态工具可检测代码里的危险函数与不当配置，动态工具则模拟攻击发起探测请求。常用的开源工具包括 OWASP ZAP、Burp Suite Community、SQLmap、Nmap。专业商业工具有 AppScan、WebInspect。测试过程遵循信息收集、漏洞扫描、验证利用、报告生成四个阶段。

SQLmap 自动化检测和利用 SQL 注入漏洞。该工具支持多种数据库类型，提供布尔盲注、时间盲注、联合查询等多种技术。安全测试人员使用 SQLmap 获取数据库架构、导出数据、尝试获取 shell。XSS 测试工具如 XSStrike 检测反射型和 DOM 型 XSS，使用模糊测试生成有效载荷。Burp Suite 作为抓包代理，拦截和修改 HTTP 请求。测试人员手动构造攻击参数，分析服务器响应差异。工具扫描结果需要人工验证，排除误报。

（二）漏洞修复与安全加固策略

发现漏洞后及时修复。SQL 注入漏洞的修复方案是将动态 SQL 改为参数化查询。Java 使用 PreparedStatement，PHP 使用 PDO 绑定参数，Python 使用参数化查询。对于无法修改代码的场景，使用存储过程或输入验证白名单作为补偿措施。XSS 漏洞修复采用输出编码函数。不同上下文使用不同编码函数，HTML 实体编码用于标签之间，JavaScript 编码用于脚本字符串。文件上传漏洞修复要求配置文件存储目录不可执行，使用随机重命名保留原始扩展名白名单 [5]。

安全加固涵盖 Web 服务器和操作系统层面。移除不必要的 HTTP 方法，禁用 TRACE、OPTIONS。修改服务器默认响应头，隐藏版本信息。配置目录权限最小化原则，Web 根目录禁止写入。操作系统层面关闭无关端口和服务，定期更新安全补丁。数据库服务器使用专用低权限账户连接，分离 Web 与数据库服务器。备份重要数据，制定应急响应预案。安全加固是一个持续过程，每次版本更新后重新评估风险。

（三）综合防护体系构建

单点防御无法应对复杂攻击。综合防护体系采用分层防御模型。第一层是网络边界防御，包括防火墙和入侵检测系统。第二层是 Web 应用防御，部署 WAF 和 Web 服务器安全模块。第三层是应用代码防御，执行安全编码规范和代码审查。第四层是数据层防御，实施数据加密和审计日志。每层之间协同工作，一层防线被突破后其他层继续防护。防护体系中，安全开发生命周期 (SDL) 起到源头控制作用。需求阶段进行威胁建模，设计阶段评审安全架构，编码阶段使用静态分析工具，测试阶段开展渗透测试，运维阶段监控异常行为。DevSecOps 理念将安全融入 CI/CD 流水线。自动化安全扫描在每次代码提交时触发。运行时应用自我保护技术 (RASP) 嵌入应用程序内部，实时检测攻击行为并阻断。构建安全态势感知平台，收集日志和告警数据，利用机器学习识别未知威胁。定期开展安全演练，检验防护体系有效性。

五、结束语

Web 环境下的网络安全攻防技术研究需要持续深入。该文系统分析 SQL 注入、XSS、CSRF、文件上传等攻击手法，探讨输入验证、会话管理、WAF 等防御措施。攻击技术不断演进，防御体系也需同步升级。实践证明，单一防御手段存在局限性，构建多层综合防护体系才能有效应对威胁。未来研究可关注人工智能赋能安全检测、零信任架构在 Web 环境的应用、自动化攻防对抗等方向。只有保持攻防技术动态平衡，才能保障 Web 应用安全运行。

参考文献

[1] 付啸鹏，赵鹏，晁文杰，等 . 基于 Web 环境的网络安全攻防技术应用研究 [J]. 网络安全技术与应用 ,2026(2):1-3.

[2] 刘光，黄伟平，黎德靖，等 . Web 环境下网络安全攻防技术研究 [J]. 中国新通信 ,2024,26(13):44-46.

[3] 吕东晨 . Web 环境下网络安全攻防技术的应用 [J]. 中国宽带 ,2024,20(1):10-12.

[4] 黄哲韬 . 分析互联网企业 Web 系统漏洞对网络空间安全影响 [J]. 网络安全技术与应用 ,2023(5):10-12.

[5] 牛鸿伟 . 基于 Python 的 SQL 注入攻击防御的策略研究 [J]. 现代信息科技 ,2021,5(21):146-148.

 Copyright © This Work is Licensed under A Commons Attribution-Non Commercial 4.0 International License.