

基于行为仿真的 Web 系统口令安全测试方法

白小凡^{1,2}, 王俊琦^{1,2}, 赵易如^{1,2}, 王笑克^{1,2}, 赵磊^{1,2}

¹ 武汉大学网络安全学院 武汉 中国 430072

² 武汉大学空天信息安全与可信计算教育部重点实验室 武汉 中国 430072

摘要 身份认证是保障信息系统安全的重要手段, 自 20 世纪 70 年代以来, 口令一直是最常用的身份认证方式。然而, 由于低信息熵弱口令的滥用和凭证泄露事件的频繁发生, 基于口令的身份认证面临安全挑战。攻击者能够利用暴力破解和凭证填充等方式入侵系统, 威胁信息系统的整体安全性。现有的自动化口令安全检测方法主要包括脚本集成方法、基于分析登录表单构造报文的方法和脚本录制方法。然而, 这些方法在通用性、有效性和大规模检测效率等方面存在显著局限, 难以应对现代 Web 系统中复杂多样的身份认证机制。为应对上述挑战, 本文提出了一种基于行为仿真的 Web 系统口令安全测试方法。该方法首先设计了三种登录行为模型, 即单步式登录、分段式登录和弹出式登录, 并基于这些模型提出了标签定位与登录行为链的自动化构造方法, 从而实现了基于行为仿真的凭证提交。此外, 本文提出了一种基于差分比较的报文重构策略, 通过差分分析识别加密、签名等防护机制, 并针对不同的保护策略选择相应的报文重构方法。实验结果表明, 本文提出的方法在登录页面识别准确率、报文构造通用性及大规模检测效率等方面均显著优于现有方法。其中, 登录页面识别误报率在两个数据集上分别为 0.52% 和 1.68%, 显著低于 Shepherd 方法的 30.30%。本文的研究为提升现代 Web 系统的口令安全性提供了一种高效可行的解决方案。

关键词 身份认证; 不安全口令; 登录行为模型; 行为仿真

中图分类号 TP309 DOI 号 10.19363/J.cnki.cn10-1380/tn.2026.05.01

A Behavioral Simulation Approach for Password Security Testing in Web Systems

BAI Xiaofan^{1,2}, WANG Junqi^{1,2}, ZHAO Yiru^{1,2}, WANG Xiaoke^{1,2}, ZHAO Lei^{1,2}

¹ School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China

² Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, Wuhan University, Wuhan 430072, China

Abstract Authentication is a fundamental measure to ensure the security of information systems. Since the 1970s, passwords have been the most widely used method for authentication. However, the widespread use of weak passwords with low entropy and the frequent occurrence of credential leakage incidents have introduced significant security challenges to password-based authentication. Attackers often exploit vulnerabilities through brute force attacks and credential stuffing, jeopardizing the overall security of information systems. Existing automated password security detection methods, such as script integration, login form-based message construction, and script recording, suffer from notable limitations in generality, effectiveness, and scalability. These constraints hinder their ability to address the increasingly complex and diverse authentication mechanisms prevalent in modern web systems. To overcome these challenges, this study presents a novel password security testing approach for web systems based on behavioral simulation. The method begins with the design of three distinct login behavior models: single-step login, multi-step login, and popup login. Building upon these models, an automated mechanism for label identification and login behavior chain construction is introduced, enabling a simulated and adaptive credential submission process. Additionally, a differential comparison-based message reconstruction strategy is proposed, leveraging differential analysis to identify protective mechanisms such as encryption and digital signatures. Tailored reconstruction methods are then applied based on the identified protection strategies. Experimental results provide strong evidence of the proposed approach's superiority in login page identification accuracy, message construction generality, and large-scale detection efficiency, as measured by credential submission time. Notably, the false positive rates for login page identification are 0.52% and 1.68% across two datasets, significantly outperforming the 30.30% rate achieved by the Shepherd method. This research offers an efficient and robust solution for improving password security in modern web systems.

Key words authentication; insecure passwords; login behavior model; behavioral simulation

通讯作者: 赵磊, 博士, 教授, Email: leizhao@whu.edu.cn.

本课题得到国家自然科学基金项目(No. U2436205, No. 62532008, No. 62172305) 和中国博士后科学基金会与湖北省联合资助项目(No. 2025T054HB)资助。

收稿日期: 2024-10-22; 修改日期: 2025-02-25; 定稿日期: 2026-04-09

1 引言

身份认证是保障信息系统安全的第一道防线,自 20 世纪 70 年代以来,口令一直是最主要的身份认证方法^[1]。然而,低信息熵的弱口令^[2]以及数据泄露事件中泄露的凭证等易猜解或易获取的不安全口令,使得这一防线面临严峻挑战。攻击者通过暴力破解^[3]或凭证填充^[4]等技术手段,利用系统中存在的不安全口令实施非法入侵,造成数据窃取和系统破坏等攻击事件。根据 IBM 发布的 2024 年数据泄露成本报告^[5],泄露凭证已成为最常见的初始攻击媒介,对信息系统的安全构成持续威胁。据统计^[6],仅 2024 年 4 月,针对中央企业的弱口令暴力破解就高达 935 起,严重威胁了政府部门的系统安全。因此,研究自动化口令安全检测方法,显得尤为重要。

为应对不安全口令^[7-14]带来的安全风险,研究人员提出了多种自动化口令安全检测方法,主要分为三类:脚本集成方法^[15-17]、基于分析登录表单构造报文的方法^[18-21]和脚本录制方法^[20,22]。脚本集成方法通过人工分析目标系统的登录流程,编写自动化脚本并集成到工具中以实现口令安全检测。基于分析登录表单构造报文的方法通过规则库解析登录表单,构造登录报文,并通过报文重放来检测口令。这两种方法主要适用于以静态表单实现登录的特定系统(如物联网设备管理系统、内容管理系统和论坛软件系统等),其自动化实现相对容易。然而,随着 Web 系统身份认证机制的多样化和复杂化,上述技术在通用性和有效性方面存在显著不足,难以应对动态渲染页面和具有防护机制的登录页面。脚本录制方法通过人工录制登录脚本实现自动化登录,理论上可以应对多种身份认证形式,但该方法需要为每个页面单独录制脚本,不具备大规模检测能力。

除此之外,一些针对 Web 系统登录安全机制的研究^[23-32],提出了自动化登录的概念。2020 年的 Shepherd^[27]首次实现了通用的自动化登录框架。然而,该框架对注册页面等干扰页面的区分能力较弱,存在较大的误报。此外,在大规模口令检测场景中,该框架的报文重构策略需重复执行首次登录流程,显著增加了时间成本。

综上,现有工作的主要问题包括:(1) 登录页面识别准确率难保证;(2) 报文构造通用性不足;以及(3) 大规模口令爆破时间成本高。为解决上述问题,本文分析了面向 Web 系统的口令安全检测的三个核

心步骤:登录页面识别、登录凭证提交和登录报文重构,并识别出三个关键挑战。第一,登录模型难定义。Web 系统的身份认证缺乏统一的标准,开发者可以根据需求和场景选择合适的身份认证方案,或者以自定义的方式实现身份认证。因此,难以直观定义登录模型,而注册表单等干扰页面也给登录模型的定义带来挑战。第二,凭证信息难提交。现代 Web 应用程序不再是静态 HTML 页面,开始强调与用户的动态交互。Ajax 技术^[33]和虚拟 DOM 技术^[34]的使用,使得简单的表单分析与报文构造方法难以完成凭证提交。第三,通用性和高效性难兼顾。凭证提交的适用范围决定方法的通用性;而报文重构的时间成本直接影响其效率。现有方法通常为实现通用性牺牲了部分效率,而在报文重构时沿用凭证提交方法的策略,更难平衡通用性与高效性。

针对上述挑战,本文通过分析登录流程,发现两项重要观察:(1) 用户透明的提交逻辑;和(2) 凭证信息随报文提交。并基于此提出了一种基于行为仿真的 Web 系统口令安全测试方法。具体而言,本文首先以用户视角分析登录行为,设计了单步式登录、分段式登录和弹出式登录三种登录行为模型,并基于模型完成标签定位和登录行为链构造。随后,通过仿真,实现了基于行为仿真的凭证提交方法。为进一步优化报文重构,本文提出一种基于差分比较的登录报文重构策略选择方法,通过差分报文分析识别登录过程是否存在加密、签名等防护机制,并根据分析结果动态选择基于行为重构的报文重构方法或基于内容重写的报文重构方法。此外,为提高登录页面识别的准确率,本文在标签定位、登录行为链构造和报文分析环节中,引入了三次登录页面验证。

实验结果表明,本文提出的方法在页面识别准确率、报文构造通用性以及大规模凭证提交的时间成本等方面均显著优于现有检测方法。在行业数据集^①和通过网络搜索引擎 Fofa^[35]爬取的定制数据集上的实验显示,本方法在两数据集登录页面识别中的误报率分别为 0.52% 和 1.68%,远低于 Shepherd 的 30.30%。在报文构造方面,本方法支持三种登录类型,并能够应对复杂登录和有防护登录。此外,本文提出的两种报文重构策略(行为重构,内容重写)的时间成本分别为十秒级(平均 33.97 秒)和毫秒级。值得注意的是,在行业数据集中可采用内容重写策略的无防护页面占比高达 56%,这表明本文的重构策略选择能够有效降低时间成本,相较于单一报文构造方法

① 数据集来自第一届中国研究生网络安全创新大赛

在效率上更具优势。

2 背景与动机

身份认证是保护信息系统安全的第一道防线。尽管现代认证方式不断推陈出新,涌现出诸如生物因素认证、动态口令认证以及硬件令牌认证等新方式,但这些认证方式在设备兼容性、用户体验和成本等方面存在一定的局限性。相比之下,口令作为一种经典的认证方式,具备简单易用、成本低廉(不需要其他硬件设备)以及容易更改的优势,且不会面临生物因素特征被泄露或滥用的风险。因此,口令在较长时间内仍是最流行且广泛适用的认证方式之一。然而,不安全口令的存在严重削弱了其防护能力,攻击者可以直接利用不安全口令冒名顶替进入信息系统,进行攻击和信息窃取。因此,如何在不安全口令被攻击者利用前及时检测,是保障信息系统安全的关键任务。

2.1 不安全口令

不安全口令是指在认证过程中,容易被攻击者猜测、破解或直接获取的口令。这类口令在现代信息系统中广泛存在,对系统安全构成了严重威胁。主要包括以下几类。

弱口令:弱口令是指简单常见、强度低且易被猜测的口令(如 123456、admin123 等),这些口令具有较低的信息熵容易受到暴力破解或字典攻击。

默认凭证:默认凭证是设备或应用程序出厂时配置的默认用户名和口令,且用户未对其进行更改。由于这些凭证出厂时已知,攻击者很容易通过默认凭证获取对系统的访问权限。

易猜测口令:易猜测口令是指那些遵循常见模式或与用户个人信息相关的凭证,容易通过社会工程学等手段猜解。如遵循键盘模式的凭证^[7],以及包含姓名、生日和身份证号等信息的凭证^[8]。2019 年, Wang 等人^[9]指出,中文凭证倾向使用拼音、数字和特定短语,在线猜测攻击中比英文凭证更加脆弱。

泄露凭证:泄露凭证是指由于数据泄露事件而暴露的用户凭证信息。如果这些泄露的凭证被用户在多个平台或服务上重复使用,则会增加凭证填充攻击成功的风险。从 2009 年社交应用网站 Rockyou^[10]发生数据泄露为起点,全球已有超过 700 起的大规模数据泄露事件接连不断发生^[11],大量用户凭证面临被滥用的风险。凭证泄露和管理主要集中在以下几个方面,(1) 凭证使用习惯分析:2016 年 Liu 等人^[12]

分析了泄露出的 26 个口令数据库,约 2.78 亿条数据,发现个别网站有超过 50% 的用户使用纯数字口令。2017 年 Li 等人^[8]系统地调研了个人信息在口令中的使用情况,发现人们偏向于在口令中加入自己的姓名拼音、生日、身份证号等个人信息。(2) 凭证复用现象:2019 年 Wang 等人^[9]通过大规模的调研,指出用户在向一个新网站注册口令时,往往是重复使用之前的口令(44.8%),仅 14.5% 的用户会构建一个从未使用的口令。2023 年, Nisenoff 等人^[13]分析了某大学二十年来的凭证使用情况,发现用户在口令策略调整后,往往只进行简单的修改(如改变字符大小写或添加特殊字符),而不是创建完全独特的凭证。此外,许多用户在凭证泄露后并不知情,即使在被告知需修改凭证后,一些用户仍然会重复使用相同的凭证。(3) 凭证泄露的普遍性:2022 年的 PassFinder^[14],揭示了 GitHub 平台上口令泄露的严重性和普遍性。75 天内新上传的公共代码文件中,超过 6,000 个仓库存在凭证泄露问题。泄露口令的规模也相当庞大,调研 539,012 个仓库,存在泄露凭证的仓库占比 11.88%,其中 78.48% 的泄露口令为敏感口令。此外,在检查后的 16 天内,有 82.32% 的泄露口令未被移除。

不安全口令的广泛存在使得信息系统容易受到未经授权的访问和攻击。因此,如何先于攻击者,检查登录安全性并广泛评估口令安全性是一项有意义的工作。

2.2 口令安全检查

针对口令的攻击主要包括暴力破解和凭证填充两种方式。暴力破解侧重尝试所有可能的口令组合,获取可进入目标系统的有效凭证,通常包括对弱口令、默认凭证和易猜测凭证的攻击。凭证填充侧重于利用泄露数据中的已知凭证,对重复使用的泄露凭证进行攻击。

口令攻击的门槛较低,攻击者只需掌握基本的 Web 系统测试原理即可发起。然而,其潜在收益极高,一旦成功,攻击者可以冒名顶替进入系统,从而进行更深入的攻击。由于其简单易行且回报丰厚,攻击者频繁通过口令攻击尝试突破系统的防线。

中资网络信息安全科技有限公司(中资网安)从 2023 年 7 月到 2024 年 6 月中央企业的态势感知数据显示,暴力破解告警占比较高,且弱口令告警数量居高不下。2023 年 8 月,暴力破解告警的占比达到峰值,占有告警的 35%;2024 年 4 月,弱口令告警数量达到峰值,高达 935 万条(见图 1)。

面对频繁的攻击,主动识别不安全口令是保障信息系统安全的关键。为此,开发通用且高效的自动

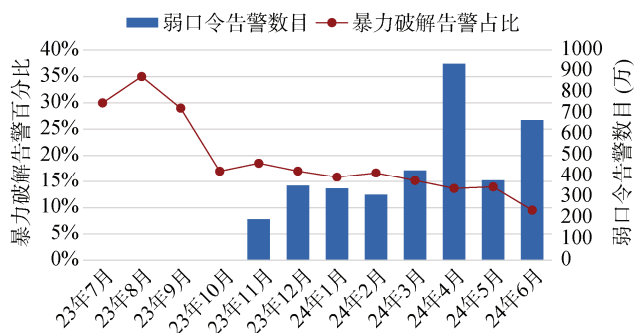


图 1 中央企业凭证攻击态势感知

Figure 1 Attack Situation Awareness of Central Enterprises

化口令安全检测方法, 提前筛查不安全口令并抵御潜在威胁, 显得尤为重要。然而, 自动化方法的设计并非易事, 由于 Web 系统缺乏统一的身份认证标准, 开发者可以根据需求和场景选择合适的身份认证方案, 或者以自定义的方式实现身份认证。这使得对登录模型的定义成为 Web 系统口令安全检测的一大挑战。

现代 Web 应用程序不再是静态 HTML 页面, 开始强调与用户的动态交互。Ajax 技术和虚拟 DOM 技术的使用, 使登录功能的实现更加自由和复杂, 无法仅通过简单的表单分析和报文组装提交凭证信息, 因此凭证信息的提交成为口令安全检测的另一挑战。具体来说, 登录功能实现的复杂性体现在以下几个方面。

动态渲染的登录页面: 一些前端框架(React、Vue)使用虚拟 DOM 技术来生成网站, 页面内容会在内存中构建成虚拟 DOM 树, 之后与实际 DOM 进行比较, 并通过 JavaScript 代码在客户端进行更新。不具备 JavaScript 代码执行能力的方法, 无法得到登录相关代码。

防护机制: 某些 Web 系统设置了一系列防护机制, 如签名、报文加密、使用 CSRFtoken 等。此外, 多数据包登录也具有一定的防护效果。应对含防护机制的登录需要对数据包构造流程进行深入分析。

自定义实现: 在 Web2.0 时代, JavaScript、Ajax 的出现给前端带来许多新特性, 开发者可以为任意标签动态绑定事件和事件处理函数, 这使得凭证提交实现方式多样化且难以归纳。

2.3 相关工作

现有的 Web 系统凭证攻击技术和工具大多集中于特定系统, 如物联网设备管理系统、内容管理系统以及论坛软件系统等。这些系统的登录形式较为统一, 登录建模相对简单, 登录页面的识别和报文构造也较为容易。此外, 上述系统流行度较高, 比如某

品牌路由器的登录页面广泛存在于所有使用该路由器的场景。因此, 针对这类系统的自动化方案可以“一次解决, 广泛适用”。然而, 尽管这些方法在特定场景下表现出色, 但由于各类系统的登录认证机制、数据流动和交互流程存在显著差异, 其通用性往往较弱。此外, 现有的技术难以应对复杂的 Web 登录页面, 如动态渲染的页面、具有防护能力的页面以及基于动态绑定实现的登录页面。因此, 当前工作在有效性和适用范围上存在局限性。现有的主流方法主要可以分为以下几类。

报文重放方法: 报文重放方法是最经典、最普遍的口令安全检测方法, 一般需要借助 Web 应用程序安全测试软件(如 BurpSuite^[36])的力量。

如图 2 所示, 该方法通常通过设置上游代理拦截 Web 的人工登录请求, 然后手动分析数据包, 寻找登录相关的报文, 并标记用户名和口令参数, 最后修改信息执行重放。这种方法只能处理无防护的情况, 若登录中存在加密、签名和动态刷新 token 等防护机制, 则无法通过单一的报文分析处理。此时, 需要人工分析前端 JavaScript 代码中的数据处理逻辑, 定位加密、签名使用的密钥, 以及获取 token 位置和刷新规则。

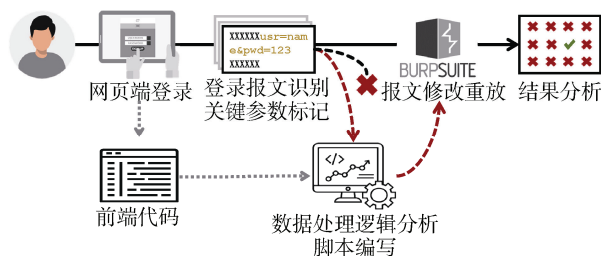


图 2 报文重放攻击流程

Figure 2 Packet Replay Attack Process

脚本集成方法: 为了实现自动化, 出现了基于人工分析的自动化脚本集成方法。这种方法需要手动分析目标系统的登录流程, 然后构建相应的自动化登录脚本, 最后将脚本集成在工具中。s0md3v 的 Web 系统弱口令暴力破解工具 Blazy^[15]和 Rapid7 的物联网默认口令扫描工具 IoTSeeker^[16]是两项典型的开源工作。它们首先对目标登录页面进行指纹识别, 判断目标页面的脚本是否集成在工具中。然后, 调用对应的自动化脚本实现口令安全检测。2020 年 Yu 等人^[19]设计了 DBSCAN 算法, 在识别出设备的类型和品牌后, 调用自动化脚本进行登录尝试。这类方法未能摆脱人工分析, 用户需要通过手动分析登录页面的结构和凭证提交逻辑, 来构建相应的自动化脚

本。虽然其在分析过的登录形式上表现良好,但无法处理未集成的 Web 系统,适用范围受到人工能力和时间投入的制约,无法应对未分析集成的登录页面,它通常被视为范围受限的自动化方案。

基于分析登录表单构造报文的方法:为了扩展适用范围并进一步提升口令安全检测的自动化程度,一些工作开始聚焦于通过分析登录表单来构造报文,即利用规则库来处理登录表单实现登录报文构造。最具代表性的是 2019 年 Xu 等人的 AWKD^[18]、2022 年的 Long 等人的 WebCrack^[19]以及一些商业漏洞扫描器 AWVS^[20]、Nessus^[21]。这类工作预设了一些常见的用户名和口令的参数名作为规则库,然后对目标网页所有的表单进行匹配,随后定位表单中发送数据包的相对 URL,以此构造登录数据包,并通过不断替换数据包中输入的凭证信息后进行重放。这类方法主要适用于基于表单的自动化登录,但无法有效应对登录过程中的加密、签名等复杂情况。虽然它们实现了真正意义的自动化,但无法应对更复杂的登录逻辑和 Ajax 提交的表单,而且仅能处理同时提交用户名和口令的登录方式。

脚本录制方法:为了提升对复杂登录的解决能力,一些工作通过人工录制登录脚本实现自动化登录。如商业漏洞扫描工具 AWVS 和 Microsoft 开发的提供 Web 应用程序自动化测试能力的开源库 Playwright^[22]。这类方法通过记录用户在登录过程中输入和点击的标签,来复现人工登录流程,并生成登录脚本。这种方法理论上适用于任何登录页面,但在大规模口令安全检测中,需要为每个系统录制脚本,工作量庞大。因此,这种方法实际上是一种半自动化的解决方案,不具备大规模检测的能力。

Web 系统登录安全机制研究:目前,在 Web 系统登录安全机制研究领域,有很多工作对 Web 系统自动化登录进行研究。2017 年 Van Acker 等人^[23]首次提出通过登录自动化扩展网络测量的规模,并设计了一个半自动化登录模型,在人工识别登录区域后,利用爬虫实现登录。2018 年 Ghasemisharif 等人^[24]的工作和 2022 年的 SSOscan^[25]实现了单点登录(Single Sign On, SSO)的自动化登录,但 SSO 并不具有广泛性(占 Alexa Top 10M 的 5.8%)。2020 年 Drakonakis 等人^[26]的研究提出了一种利用表单字段的启发式规则和 HTML 正则分析的自动化登录方法。然而,该方法的适用范围较为有限,仅能处理单步式登录,无法有效应对更复杂的登录流程。同年 Shepherd^[27]首次实现了通用的 Web 系统自动化登录,其通过 Selenium^[28]框架,控制无头浏览器与网页的

交互,使用多种启发式规则判断网页是否存在登录功能,并定位登录区域实现凭证提交。这项工作可广泛适用于单步式登录和分段式登录,为大规模的网络测量提供了基础,后续 2022 年 Senol 等人^[29]的工作和 2023 年 Rautenstrauch 等人^[30]的工作,均采用与 Shepherd 相似的原理实现自动化登录。然而,Shepherd 难以准确区分登录页面与注册和年龄认证等干扰页面,而且无法处理弹出式登录。2023 年 Al Roomi 等人^[31]通过机器学习方法提升登录页面的识别准确性,训练了一个支持向量机(Support Vector Machine, SVM)区分登录与注册页面,并模拟用户行为进行登录或注册。此外,2024 年 Senol 等人^[32]的一项工作,使用机器学习的方法提升了登录、注册和非登录页面识别的准确性,但这项工作仅进行了页面的判定,未能进一步实现登录。这类工作往往使用已知有效的凭证(如 BugMeNot^[37]的共享凭证或注册获取的凭证)进行登录提交,凭证提交的次数有限,其报文重构采用与首次登录同样的流程,在面对大规模口令爆破场景下频繁的报文重构时,时间成本较高。

综上所述,虽然上述方法在自动化口令安全检测方面取得了一定进展,但它们均存在局限性,难以同时兼顾通用性和高效性。具体来说,现有方法存在以下三个问题。

- **登录页面识别准确性难保障:**登录页面的识别和关键操作的定位存在误报和漏报的风险。此外登录页面的误报和关键操作的错误识别,也会引入无效的凭证提交操作,带来额外的时间开销。
- **报文构造通用性不足:**主要针对热门或流行的站点,对个性化登录页面的报文构造能力不足,且难以应对弹出式登录以及其他复杂登录形式。
- **大规模爆破时间成本高:**未对不同页面自动化登录凭证提交的难度进行区分,采用相同的报文重构方式,时间成本未能满足高效口令爆破的需求。

2.4 难点与挑战

本文从登录页面识别与登录表单定位、登录凭证提交以及登录报文重构三个核心步骤入手,分析了 Web 系统口令安全检测的关键环节,并识别出以下三项核心挑战。

- **登录模型难定义。**Web 系统的身份认证缺乏统一的标准,开发者可以根据需求和场景选择合适的身份认证方案,或者以自定义的方式实现身份认证。这种多样性使得直观定义登录模型

的难度显著增加。此外,注册表单等干扰页面也给登录模型的定义带来挑战。

- **凭证信息难提交。**现代 Web 应用程序不再是静态 HTML 页面,开始强调与用户的动态交互。Ajax 技术和虚拟 DOM 技术的使用,使登录功能的实现更加自由和复杂。这使得传统的表单分析与报文组装方法无法有效地提交凭证信息,导致传统方法失效。
- **通用性和高效性难兼顾。**凭证提交的适用范围,决定方法的通用性;而报文重构的时间成本,决定方法的效率。为适应 Web 系统的多样性,凭证提交往往为了实现通用性牺牲了一定的效率。而现有方法在报文重构时,通常沿用与凭证提交相同的方法,这在保障通用性时,较难兼顾效率,在大规模检测中表现尤为突出。

针对上述挑战,本文提出了一种基于行为仿真的 Web 系统口令安全检测方法。与现有的脚本集成、登录表单分析或脚本录制方法不同,本文创新性地从用户的视角出发,构建三种登录行为模型,这些模型通过模拟用户的实际登录过程,能够广泛适配不同 Web 系统中的身份认证机制。此外,针对传统方法中报文重构效率低下的问题,本文提出了一种基于差分比较的报文重构策略,通过差分分析识别加密和签名等防护机制,选择最合适的报文重构方式,极大提高了检测效率。这一创新性的检测方法在扩大口令安全检测适用范围的同时,也显著提高了检测的效率,为自动化口令安全检测技术的发展提供了新的思路和解决方案。

3 Web 系统登录行为

本文分析 Web 系统登录行为,得到透明的提交逻辑和凭证信息随报文提交两项重要观察。基于此设计了登录行为模型,该模型可以广泛应用于不同的登录流程和界面。本文提出了差分报文分析方法,该方法可以筛选登录相关报文,过滤非登录页面,并分析登录的安全防护机制。

3.1 登录行为分析

本节对 Web 系统的登录行为进行了全面的分析与调研,将登录流程划分为 6 个步骤,涉及三个主要参与角色:用户、浏览器和服务端,如图 3 所示。具体来说,用户作为登录的发起者,需要判断页面是否为登录页面,并在确认后输入凭证信息、发起登录请求;浏览器作为登录的实际执行者,负责执行登

录逻辑和构造数据包;服务端作为登录合法性的验证者,对接收到的数据包进行解密、签名验证和请求合法性判断,并在数据库中校验凭证的有效性,随后将验证结果通过数据包回传至浏览器,提示用户凭证是否合法或进入登录后状态。

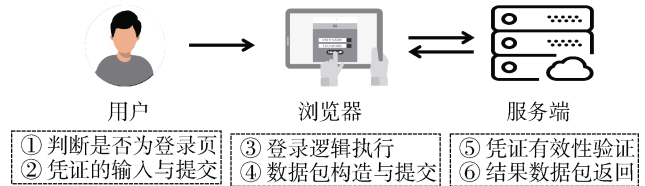


图 3 登录全流程

Figure 3 Complete Login Process

通过对登录流程的深入分析,本文得出以下两项重要观察。

(1) 用户透明的登录报文构造:Web 系统中的登录凭证提交逻辑和数据包构造对用户来说是透明的。用户只需在定位到网页中的登录元素后,进行简单的输入和点击操作,而复杂的前端逻辑处理由浏览器自动完成,包括执行 JavaScript 代码以实现前端复杂的数据处理逻辑。因此,相较于人工分析页面中 JavaScript 代码的执行流和动态绑定事件等复杂逻辑,直接模拟用户的登录行为是一种更简洁、直接且具有更高通用性的自动登录方案。

(2) 凭证信息随登录报文传递:用户提交的凭证信息会通过浏览器构建的数据包传输到服务端进行有效性校验。因此,一次成功的登录凭证提交后,在浏览器发送的大量数据包中,定然存在携带凭证信息的数据包。因此,可以通过对数据包的分析,验证登录凭证提交是否成功。并依据是否包含凭证信息过滤登录无关报文。

基于上述观察,本文从以下两个方面展开研究。针对第一项观察,本文提出单步式登录、分段式登录和弹出式登录三种登录行为模型,并基于登录行为模型构造登录行为链,从而实现基于行为仿真的凭证提交。针对第二项观察,本文提出了差分报文分析方法,筛选登录页面和有效的登录行为链,并进一步分析登录报文是否为有防护登录。

3.2 登录原子操作

为实现精确的登录行为仿真,需要对用户的每个操作进行细粒度的分析。为此,本文提出了登录原子操作(Login Atomic Operation, LAO)的概念。

定义 1(登录原子操作).登录原子操作指用户在提交身份凭证(用户名和口令)信息时,单次的输入或点击操作。本文使用 $LAO=<attribute, action>$ 表示登

录原子操作, 其中, **attribute** 表示操作对象的属性, **action** 表示操作类型, 分为 **input** 和 **click** 两种类型。

本文通过对大量登录页面的观察和分析, 归纳出用户登录主要涉及六个登录原子操作, 如表 1 所示。其中<用户名, **input**>、<口令, **input**>和<登录, **click**>是登录的核心操作, 也是登录的必选项, 分别用来标识用户身份、保证用户合法性以及提交凭证。其他三个登录原子操作虽然作为可选项, 但也在不同登录流程中被广泛采用。<验证码, **input**>作为一种典型的凭证攻击缓解措施, 广泛存在于对账户安全存在较高要求的网站。而<下一步, **click**>则是一类特殊的登录方式(分段式登录)中的必选项。此外, 随着近些年隐私合规相关法律法规的出台, <合规协议, **click**>也频繁出现在登录页面中, 如用户隐私协议等。

表 1 登录原子操作

Table 1 Login Atomic Operation

登录原子操作	可选/必选	标签
<用户名, input >	必选	input (text\ none)
<口令, input >	必选	input (password)
<验证码, input >	可选	input (text)
<登录, click >	必选	button \ img \ a \ span \ div
<下一步, click >	可选	button \ img \ a \ span \ div
<合规协议, click >	可选	input (radio\ checkbox)

用户在登录过程中的每一个登录原子操作的实际操作对象均为 **HTML** 标签, 通过对这些标签的精准定位, 可以模拟用户对输入框和登录按钮的定位过程, 从而实现登录点的识别。登录原子操作及其相关标签的对应如表 1 中标签列所示。

3.3 登录行为模型

基于登录原子操作, 本文进一步构建登录行为模型。

定义 2(登录行为模型)。尽管网页登录的具体实现方式多种多样, 但其在核心逻辑上具有一定的共性。登录行为模型是登录过程中最基本的登录原子操作序列, 其中的每一个登录原子操作都是实现登录的必选项。本文使用 $M=(LAO_a, LAO_b, \dots, LAO_m)$ 来表示登录行为模型。

定义 3(登录行为链)。登录行为是指用户在提交身份认证信息时, 依次进行的输入和点击操作, 这些操作构成了登录行为链。由于登录行为链可以看作多个登录原子操作的排列组合, 本文使用 $LBS=(LAO_1, LAO_2, \dots, LAO_n)$ 表示某网页的登录行为链(Login behavior sequence, LBS)。例如, 一个典型的登录行为链可以表示为 $LBS=(\text{<用户名, input>, <口令, input>, <登录, click>})$ 。

令, **input**>, <登录, **click**>, <合规协议, **click**>)。M 给出的操作序列可以看作登录行为链中的关键字序列, 其中的每一个登录原子操作均为对应登录方式中的必选项, 而其他可选登录原子操作可能出现在登录行为链中的任意位置。

本文对各类网站的登录过程进行了广泛的调研, 总结出三种主流登录方式, 并分别为其构建登录行为模型。

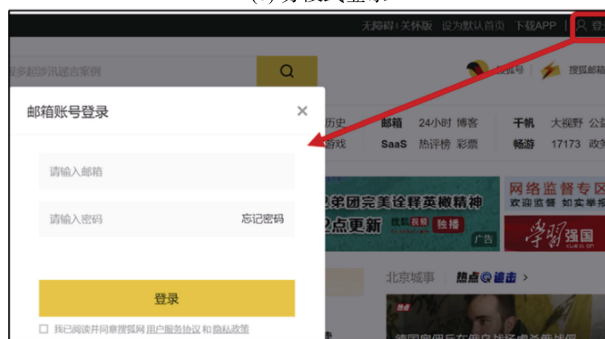
• **单步式登录**: 所有需要执行的原子操作均在同一个页面上进行, 如图 4(a)所示, 用户只需多次进行输入操作后, 点击登录即可。这是最为普遍的一种登录方式, 其登录行为模型 $M_1=(\text{<用户名, input>, <口令, input>, <登录, click>})$ 。



(a) 单步式登录



(b) 分段式登录



(c) 弹出式登录

图 4 登录方式示例

Figure 4 Example of Login Methods

• **分段式登录**: 口令字段初始不可见, 在用户输入用户名并提交后, 才可看到口令输入表单, 如图 4(b)所示。这种方式通过将用户名和口令输入步骤分离, 显著增强了安全性。分段式登录能够抵御一些

常见的登录页面检测技术,尤其是自动化脚本分析页面结构(如表单分析法)的检测方法。因其动态加载口令框的设计,静态分析工具难以准确定位密码框,从而导致漏报。分段式登录的登录行为模型 $M_2=(\langle \text{用户名}, \text{input} \rangle, \langle \text{下一步}, \text{click} \rangle, \langle \text{口令}, \text{input} \rangle, \langle \text{登录}, \text{click} \rangle)$ 。

• **弹窗式登录:** 登录表单不可见,仅保留一个登录按钮,需先点击登录按钮,弹出登录框后再执行后续操作,如图 4(c)所示。这种设计便于开发者在页面展示更多的内容,常出现在企业主页或门户网站上。其登录行为模型 $M_3=(\langle \text{登录}, \text{click} \rangle, \langle \text{用户名}, \text{input} \rangle, \langle \text{口令}, \text{input} \rangle, \langle \text{登录}, \text{click} \rangle)$ 。

3.4 差分报文分析方法

浏览器提交的数据包中存在大量登录无关的报文,包括输入标签错误识别触发的报文、无效登录行为链触发的报文以及网站自身的轮询和心跳机制引

入的报文。这些报文的存在会对以下两个方面产生负面影响:首先,它们会影响登录类型的判定,例如将无关报文识别为多数据包登录。其次,登录无关报文会干扰对数据包中凭证相关信息的定位,进而降低口令安全检测的效率。

为了解决上述问题,基于第二项观察,本文设计了一种差分报文分析方法。该方法通过使用两组不同的凭证信息分别进行登录行为仿真,并对产生的两组报文进行差分比较,以过滤无关流量,从而实现登录类型的识别,如图 5 所示。具体而言,两次行为仿真的登录行为链是不变的,且两次登录仿真同时发起,因此两组报文中产生的数据包个数和顺序都是一致的,而导致两组报文集发生改变的唯一变量就是不同的凭证信息。由于登录无关报文不受登录凭证信息改变的影响,因此在两组报文集中保持一致。

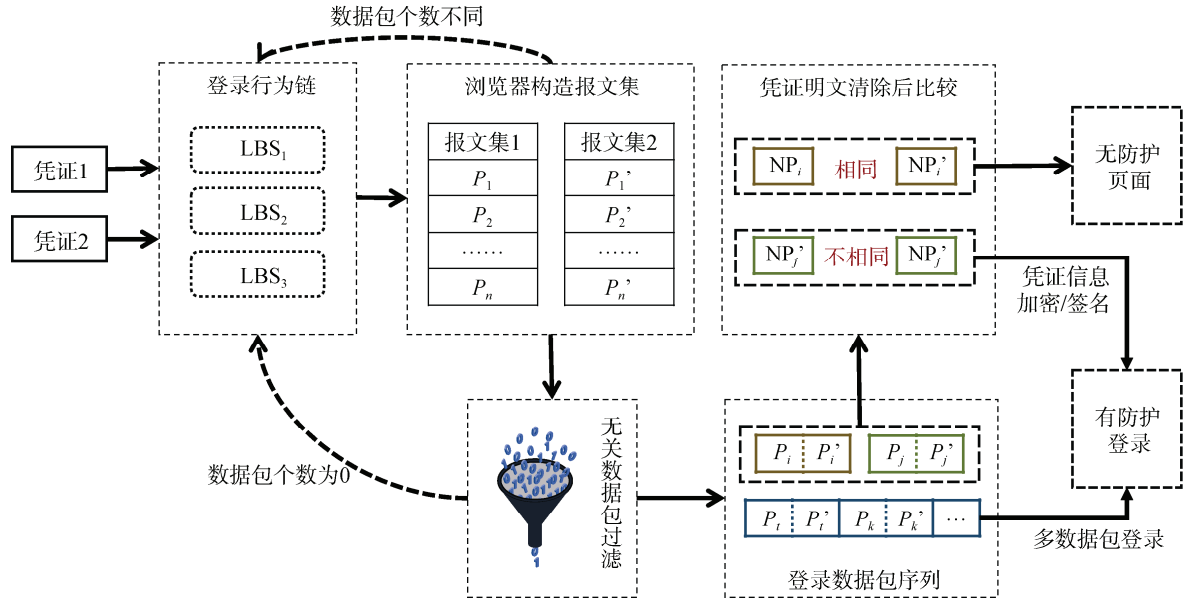


图 5 差分报文分析方法

Figure 5 Differential Packet Analysis Method

基于上述思路,本方法做出以下假设:

- 若两组报文集的数据包数量不同,则可能为网络故障导致的报文缺失,应退出并重新执行仿真。
- 两组报文集中完全相同的数据包被视为登录无关流量。过滤登录无关流量后,剩余数据包被视为登录相关报文,并存入登录数据包序列。
- 若两组报文集中数据包数量均为零,则该次仿真构造的登录行为链 LBS_1 未能成功提交登录凭证,应更换其他登录行为链进行仿真。若多

组登录行为链(LBS_2, LBS_3)仿真后数据包数量均为零,则该页面可被认定为非登录页面。

- 若登录数据包序列中数据包对数量大于零,则表明登录行为链成功拟合了用户登录行为,且成功提交登录凭证,可以进一步对登录类型进行分类。

差分报文分析方法的过程如算法 1 所示。

算法 1. 报文分析算法。

输入: 两组报文集 $\text{reqP}=\{P_1, P_2, \dots, P_n\}$ 和 $\text{reqP}'=\{P_1', P_2', \dots, P_n'\}$, 两组凭证($\text{uname}_1, \text{pwd}_1$)和

(uname₂, pwd₂)

输出: 登录页面识别标志位 isLoginPage, 登录类型 loginType, 无防护模式登录报文 pkt

BEGIN

初始化 loginReq1 $\leftarrow \{\}$, loginReq2 $\leftarrow \{\}$

// 初始化登录数据包序列

IF |reqP| $\neq$ |reqP'| **THEN**

RETURN (FALSE, NULL, NULL)

FOR (P_i, P_i') $\in$ (reqP, reqP') **DO**

IF MD5(P_i) = MD5(P_i') **THEN**

CONTINUE

ELSE

 APPEND P_i TO loginReq1

 APPEND P_i' TO loginReq2

END FOR

IF |loginReq1| = 0 **THEN**

RETURN (FALSE, NULL, NULL)

IF |loginReq1| > 1 **THEN**

 loginType $\leftarrow$ 2 //判断为有防护登录

RETURN (TRUE, loginType, NULL)

IF |loginReq1| = 1 **THEN**

 pkt1 $\leftarrow$ loginReq1[0/

 pkt2 $\leftarrow$ loginReq2[0/

 npkt1 = pkt1 \ {uname1, pwd1}

 npkt2 = pkt2 \ {uname2, pwd2}

 // 去除明文的用户名和密码

IF npkt1 = npkt2 **THEN**

 loginType $\leftarrow$ 1 //判断为无防护登录

RETURN (TRUE, loginType, pkt1)

ELSE

 loginType $\leftarrow$ 2 //判断为有防护登录

RETURN (TRUE, loginType, NULL)

在登录类型分类中, 若登录数据包序列 loginReq1 中的数据包对的数量大于 1, 则该页面实现了多数据包登录, 是一种有防护登录。当登录数据包序列中数据包对数量等于 1 时, 需进一步处理数据包对中的凭证信息。具体包括清除数据包对中明文的凭证信息, 如 (P_i, P_i') 中的用户名和口令, 以及比较清除后的数据包对 (NP_i, NP_i')。若清除凭证后数据包内容完全一致, 则该登录方式可被判定为无防护登录; 若数据包内容存在差异, 则表明可能对凭证信息进行了加密或签名操作, 从而可判定为有防护登录。

4 基于行为仿真的口令安全检测方法

本节介绍了基于行为仿真的不安全口令检测方法, 以释放登录行为模型和差分报文分析的能力, 用于实现通用高效的自动化口令安全检测。

4.1 方法概述

图 6 揭示了基于行为仿真的口令安全检测方法的整体架构, 包含 4 个核心部分: 网页打开与预处理、基于行为仿真的凭证提交、基于差分比较的报文重构策略选择、基于状态比较的凭证有效性判定。

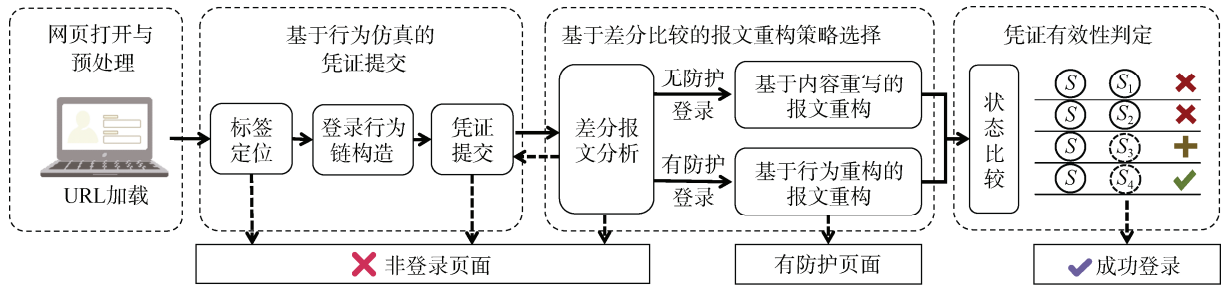


图 6 基于行为仿真的口令安全检测方法

Figure 6 Password Security Detection Method Based on Behavioral Simulation

首先, 本方法调用无头浏览器框架 chromedp^[38] 打开目标网页进行预处理工作, 先通过监听 loadEventEnd 事件确定页面初始化完成, 再通过 Hook 技术监听浏览器的 JavaScript 上下文实现对浏览器所发送和响应的 HTTP 报文的监听, 获取浏览器构造的登录报文。其次, 对登录原子操作相关的标签进行定位, 并基于三种登录行为模型构建登录行为链, 登录行为链模拟点击和输入, 通过浏览器进行凭证提交。然后, 通过差分报文分析方法, 对非登

录页面、无效的登录行为链条和登录无关报文进行过滤, 对登录是否有防护进行分析, 并进行报文重构的选择。最后, 设定错误状态基准, 通过比较响应状态, 实现凭证有效性的判定。

4.2 预处理与报文监听

预处理与报文监听是本方法的基础, 其中的关键问题是页面的快速加载、报文监听中 JavaScript 代码注入的时机选择和页面锁定。

首先, 初始化浏览器的兜底超时、退出等待时间

等参数, 规避因服务器故障产生的拥塞。然后, 调用 `chromedp` 库开启无头浏览器并进行优化, 如随机生成一个 `User-Agent` 头以绕过部分网页的安全保护, 忽略证书错误以实现自签名网站的保护, 以及禁用一些安全策略与图片加载以提高页面加载效率。

其次, 网页加载时的流量与登录无关, 因此本方法使用了非标准的实现方式。HTML 标准组织 W3C 的定义中, `loadEventEnd` 事件标志了整个网页加载结束, 标准流程在浏览器传回 `loadEventEnd` 事件后开始监听流量。但实际情况下, 存在由于 Ajax 请求过多, 网页并未在 `loadEventEnd` 事件出现后完成渲染的情况。因此, 可以设置一个时间阈值, 在打开网页后, 直接休眠一段时间以等待网页加载完成。并在行为仿真之前挂载报文监听, 以获取行为仿真构造的报文。本方法通过在浏览器上下文中插入 JavaScript 代码添加事件监听器, 在 `requestStart` 事件触发时, 打印请求信息获得整个报文。此外, 为后续分析的便利, 本方法对静态资源的加载进行了拦截。

最后, 本方法对 302 页面重定向进行了处理, 通过将返回的状态码修改为 204(No Content), 确保浏览器可以停留在待测试页面。预处理与报文监听为后续的凭证提交和报文重构奠定了坚实的基础。

4.3 基于登录行为仿真的凭证提交

本方法的凭证提交基于对登录行为的建模与行为仿真, 分为登录标签定位、登录行为链构建和凭证提交三部分。首先, 通过标签定位, 识别登录页面中可能存在的登录原子操作 LAO; 随后, 进行三种登录行为模型(M_1 , M_2 , M_3)的匹配; 接着, 基于登录行为模型构建登录行为链 LBS; 最后, 浏览器根据 LBS 实施行为仿真, 提交凭证。

4.3.1 标签定位

用户登录过程中的每一个 LAO 的实际操作对象均为 HTML 标签, 因此可以通过标签定位来模拟用户对登录相关的输入框和登录按钮的定位过程, 并识别可能存在的 LAO。本方法提出以下五点启发式规则, 用于实现 LAO 相关的标签定位。

- 关键字匹配: 具体来说, 标签定位的登录相关标签的周围, 通常会有与其 LAO.attribute 相关的关键字。如<用户名, input>的相关标签 input 中的各个属性如 id、class、name 以及 placeholder 的取值中, 会有 username、uname 等关键字出现。因此可以通过关键字匹配, 定位对应的标签。
- 口令特殊属性: input 标签的 type 属性为 password, 则该标签与<口令, input>相关。

- 分段式登录特性: 只有<用户名, input>相关标签, 但<口令, input>相关标签不存在, 可推断其与<下一步, click>相关。
- 操作局部性: <口令, input>与<登录, click>相关标签通常在页面布局上距离较近。因此, 本文提出了一种基于“贪心策略”的设计方法, 旨在提升标签定位的成功率。具体而言, 将<口令, input>关联标签后的 3 个具有 onclick 属性的标签, 均视作潜在的<登录, click>标签。通过这一设计, 在用户输入口令后, 扩展了对潜在登录操作标签的识别范围, 从而覆盖更多可能的登录按钮, 降低了由于页面布局或动态元素变化导致的识别失败风险。然而, 尽管该方法可以有效扩大登录标签的识别范围, 但也可能引入一些误识别的标签(将与登录无关的标签误判为相关标签)。为了消除这种副作用, 本文进一步引入了差分报文分析方法(详见 3.4 节), 通过对比标签点击后的响应差异, 过滤掉登录无关标签, 从而提高系统的识别准确性。
- 关键字过滤: 设置关键字黑名单, 过滤注册、忘记密码等易混淆操作对<登录, click>的干扰。

4.3.2 登录行为链构造

在完成标签定位后, 将识别到的 LAO 与三种登录行为模型进行匹配, 推测登录形式为单步式登录(M_1)、分段式登录(M_2)还是弹出式登录(M_3)。并基于相应的登录行为模型将 LAO 排列组合构建登录行为链 LBS。由于对 LAO 的识别可能出现误报, 例如操作局部性中提到的贪心设计, 因此一个登录页面的 LBS 可能有多条。

4.3.3 行为仿真与凭证提交

为保障凭证提交的成功率, 需要浏览器对构造的多条 LBS 进行仿真, 因此, 浏览器需要进行多次凭证提交。通过上述三个关键步骤, 本方法能够有效模拟用户登录行为并提交凭证信息。

4.4 基于差分比较的报文重构策略选择

在口令安全检测中, 每个测试目标, 都需要遍历字典中用户名和口令的多种组合进行登录尝试, 这需要频繁多次的登录报文构造。采用基于登录行为仿真的报文构造方法需调用无头浏览器模拟人工登录, 时间成本较高, 需要数十秒。Al Roomi 等人^[31]的工作指出大量网站登录页面传输未加密的用户凭证。在这种情况下, 采用一次凭证提交时间成本仅为毫秒级的基于内容重写的报文重构方法成为更优的选择。因此, 本方法提出一种基于差分比较的报文重构策略选择方法。首先, 通过 3.4 节中介绍的差分报

文分析方法,同时用两个不同的凭证进行登录仿真,获取登录相关报文。然后,在清除明文凭证信息后,进行报文差分比较。报文一致的为无防护登录,报文不一致或涉及多数据包的为有防护登录。对于无防护登录,本方法选择了基于内容重写的报文重构策略。由于浏览器在构造无防护登录报文时,只需要将用户名和口令简单赋值给数据包内的参数。因此,通过定位登录报文中行为仿真提交的凭证信息的位置,遍历用户名和口令字典的所有组合,替换凭证信息便可实现报文重构。

对于有防护登录,本方法选择了基于行为重构的策略。这种策略的实现依赖于 JavaScript 执行环境,每次凭证提交都需要浏览器的参与,包括浏览器执行前端的数据处理逻辑,实现对凭证信息的加密或签名,以及依赖浏览器实现的多数据包登录。该方法先对登录行为链 $LBS=(\dots, \langle \text{用户名}, \text{input} \rangle, \dots, \langle \text{口令}, \text{input} \rangle, \dots)$ 进行重构,将其中的用户名和口令替换为用字典中的待测试凭证。然后,调用浏览器对登录行为链进行行为仿真,完成报文重构。

4.5 基于状态比较的凭证有效性判定

每个网站的登录功能的实现是多样的,但每个 Web 系统在凭证有效和凭证无效时响应的报文一定不同。有效凭证的响应一般是为当前用户设置合法用户的 Cookie 或重定向到其他页面,而无效凭证的响应包括提示词告知凭证无效以及停留在原始页面。因此,本方法提出了基于状态比较的凭证有效性判定方法,如图 7 所示,并将状态定义为响应报文中的 Set-Cookie 字段、Location 字段、HTTP 状态码和响应报文长度的集合。具体来说,首先提交随机生成的无效凭证,并提取响应报文的状态 S ,作为无效基准。然后,提交待测试凭证,提取响应报文的状态 S_i ,并与无效基准状态 S 进行比较,若状态相同,则为无效凭证,反之需进入无效基准二次验证流程。在二次验证流程中,再次提交无效凭证,并获取相应的报文状态 S' ,并与 S 进行比较。若二者相同,则待测试凭证为有效凭证,反之则服务端可能识别到攻击行为,凭证提交被拦截返回相应提示,如 WAF 的拦截反馈。

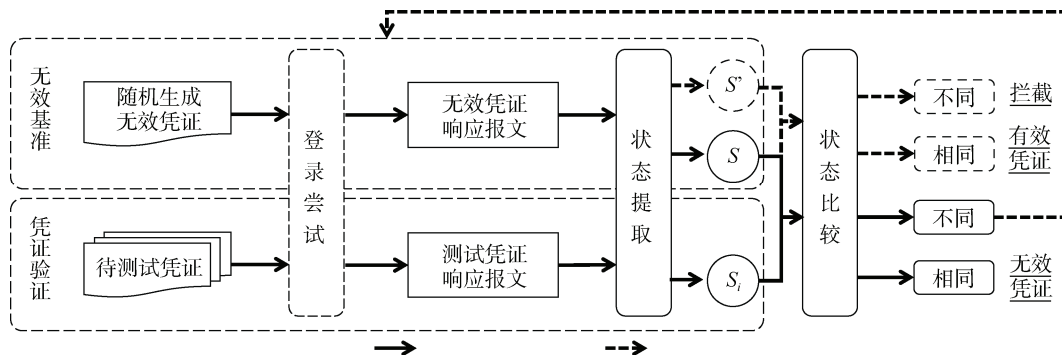


图 7 基于状态比较的凭证有效性判定

Figure 7 Credential Validity Determination Based on State Comparison

5 实验

考虑到凭证爆破攻击的敏感性,并为了确保研究的道德与法律的合规性,本文对实验进行了以下限制。首先,在不安全口令检测实验中,仅使用包含 3 个用户名和 3 个弱口令的凭证字典,见表 2,并将凭证提交的次数限定为 5 次。此限制的设定基于常见暴力破解防护机制,如账户锁定和 IP 封禁,这些机制通常在 5~10 次失败尝试后自动触发。因此,设定 5 次尝试的次数既符合实际防护策略,也避免对实验环境造成过度影响。其次,由于无法获取有效凭证(大量网页无法通过 BugMeNot 获取共享凭证,而使用社工库则存在法律和伦理风险),本实验仅评估基于行为仿真的安全口令检测方法在登录页面检

测、登录报文构造和报文重构策略选择三方面的能力。后续的凭证有效性判定,仅通过在本地搭建的 DVWA、PIKACHU 等靶机进行测试,未进行大规模实验。此外,为了减少对 Web 系统正常服务的影响,本实验安排在夜间网络流量较低的时段进行。鉴于上述限制,本实验从以下三方面集中探讨登录识别准确性、报文构造通用性和报文重构策略选择的时间成本三个核心问题,具体如下。

- RQ1: 本方法在登录页面识别准确性方面是否优于现有工作。
- RQ2: 本方法在登录报文构造方面是否具有广泛的适用性。
- RQ3: 基于差分比较的报文重构策略选择是否能降低报文重构中的时间成本。

表 2 凭证字典

Table 2 Credential Dictionary

用户名	口令
admin	abc123
test	123456
system	1234qwer

5.1 数据集

对上述问题的评估依赖大量多样化的登录页面。然而, 互联网上网页更新快、变动频繁, 该领域尚无公开统一的数据集。在本文的实验评估中, 采用了两个数据集, 行业提供的数据集(D1)和定制数据集(D2), 数据集由统一资源定位符(URL)与标签(是否为登录页面)组成。

作为评估登录报文构造能力的第三方数据集, 行业数据集的样本分布具有一定的代表性, 共包含 627 个样本, 其中登录页面 54 个, 非登录页面 573 个。然而, 由于行业数据集样本规模较小, 补充定制数据集提升测试数据的广泛性和代表性。

定制数据集聚焦于中文互联网, 侧重于度量方法的通用性和抗干扰性, 以及评估复杂场景下的性能表现。利用网络搜索引擎 Fofa, 以“login”字段为关键词进行全网扫描, 通过域名和端口筛选, 提取约 5000 个潜在目标 URL。使用网页截图工具 EyeWitness^[39]验证网址存活情况, 对存活网页去重, 并确保多种登录类型网页存在。最后根据页面的实际功能进行手工标注, 将样本划分为登录页面和非登录页面。特别地, 为确保样本多样性和抗干扰性, 非登录页面中特别选取了 100 个与登录页面结构相似的干扰页面(如注册页面、忘记口令页面)。数据集涵盖社交、论坛、证券、教育、医疗、政务、电子商务及网络设备等各个领域, 同时包含企业办公系统的登录场景, 覆盖面向客户和后台管理的登录入口, 确保样本分布的多样性和代表性。该数据集共包含 1012 个样本, 其中登录页面 357 个, 非登录页面 655 个。登录页面涵盖单步式登录、分段式登录和弹出式登录三种登录模式, 类型丰富, 包括静态 HTML 页面、动态渲染页面和基于动态绑定的登录页面。此外, 还包含了加密、签名、多数据包登录以及带有验证码等多种防护机制的页面。非登录页面包含注册页面和忘记口令页面等与登录页面相似的强干扰页面 100 个。

5.2 问题探究

- RQ1: 本方法在登录页面识别准确性方面是否优于现有工作。

为验证登录页面识别能力, 本方法在行业数据集(D1)和定制数据集(D2)上进行实验, 并与基于表单的构造方法 WebCrack 和基于行为仿真的方法 Shepherd 进行了比较。由于 Shepherd 未开源, 本文依据其论文提供的数据进行推算分析, 见表 3。实验结果表明, 基于表单分析的 WebCrack 的登录页面识别的召回率仅有 35.71%(D1)和 36.22%(D2), 而本方法在相同数据集上的召回率分别提升至 92.59%(D1)和 86.27%(D2), 较 WebCrack 提升超过两倍。同时, 本方法在行业数据集中表现出最佳的召回率、误报率、漏报率, 只有精确度略低于定制数据集的结果。本方法在对两数据集的实验中 F1 分数分别为 93.46%和 91.12%, 均高于 Shepherd 的 90.93%。相较于 Shepherd 较高的误报率 30.30%, 本方法的误报率为 0.52%(D1)和 1.68%(D2), 这得益于在标签定位、登录行为链构造和报文分析中, 对登录页面进行三次验证以及关键字黑名单。

表 3 登录页面识别能力

Table 3 Login Page Identification Capability

方法	召回率%	精确率%	误报率%	F1 分数%
WebCrack-D1	35.71	/	/	/
WebCrack-D2	36.22	/	/	/
本方法-D1	92.59	94.35	0.52	0.9346
本方法-D2	86.27	96.55	1.68	0.9112
Shepherd	88.56	93.44	30.30	0.9093

(注: Shepherd 结果为论文数据计算得出, 并非直接复现)

值得注意的是, Shepherd 方法评估的原始数据集为 2018 年 Alexa Top 1 Million 中在 BugMeNot 平台存在公开凭证的站点集合, 其中验证码登录页面的比例仅为 3.89%。其评估标准较为宽松, 仅要求站点中找到任意登录页面即可判定成功。相比之下, 本文的数据集更加多样化, 涵盖单步式、分段式、弹出式登录模式, 并通过筛选和去重增强了网页样本的丰富性和代表性。同时, 本方法采用更严格的评估标准, 要求精确识别目标 URL 是否为登录页面。基于 Shepherd 的原始数据及其评估标准, 推算其在本文数据集和严格标准下的理论性能可能有所下降。

在定制数据集中, 本方法漏报率较高(13.73%), 共有 49 个登录页面未能成功识别。主要原因包括以下两个方面: 一是部分页面引入了非输入性验证码(如滑动验证码等复杂验证方式), 导致本方法在构建登录报文时失败, 并在登录页面的第三次确认中被

认定为非登录页面;二是部分页面的登录相关标签未包含明确的关键字特征,导致识别失败。

综上所述,本方法在登录页面识别方面同时兼顾了召回率和精确率,表现优异。相较于 Shepherd,本方法在降低误报率方面有了显著的提升,在中文互联网领域表现良好。

- RQ2: 本方法在登录报文构造方面是否具有广泛的适用性。

为分析登录报文构造能力,本文对脚本集成方法、基于分析表单构造报文方法以及 Web 系统自动化登录三方面的代表性工作(IoTSeeker、WebCrack 和 Shepherd),和本方法进行了案例研究。表 4 对四种方法进行了比较。

表 4 报文构造能力

Table 4 Packet Construction Capability

	IoTSeeker	WebCrack	Shepherd	本方法
单步式登录	●	●	●	●
分段式登录	○	○	●	●
弹出式登录	○	○	●	●
复杂登录	○	○	●	●
验证码页面识别	○	●	●	●
适用页面类型	预分析	静态	通用	通用
报文构造方式	脚本集成	表单分析	行为仿真	行为仿真 内容重写

(注: ○表示不具备相应能力, ●表示部分具备相应能力, ●表示具备相应能力,最后●表示能力未知)

实验结果表明, Shepherd 和本方法在为单步式登录和分段式登录页面的构造报文方面表现优异,且能够处理动态渲染页面和有防护页面等复杂登录场景。此外,本方法还能够为弹出式登录构造登录报文,并识别和处理含有输入式验证码的页面。与仅适用于预分析页面的脚本集成方法和仅能处理静态登录页面的基于表单分析的方法相比, Shepherd 和本方法有更好的通用性。此外,本方法在弹出式登录页面中展现了更强的适应性。

两数据集中,登录报文构造成功率如表 5 所示。由于 WebCrack 无法为有验证码的页面构造报文,其报文构造成功率较低,分别为 17.86%(D1)和 12.24%(D2)。在本方法中,成功构造有效登录报文为判断登录页面的必要条件,因此报文构造的准确率与登录页面识别的召回率一致,分别为 92.59%(D1)和 86.27%(D2)。综上所述,本方法在登录报文构造方面展现出卓越的成功率和广泛的适用性,显著优于现有方法。

表 5 登录报文构造成功率

Table 5 Success Rate of Login Packet Construction

方法	成功率%
WebCrack-D1	17.86
WebCrack-D2	12.24
本方法-D1	92.59
本方法-D2	86.27

- RQ3: 基于差分比较的报文重构策略选择是否能降低报文重构中的时间成本。

本实验从基于行为仿真的凭证提交时间开销分析、报文重构时间开销分析和数据集中各类登录页面的占比三方面探讨基于差分比较的报文重构策略选择对时间开销的影响。

本实验首先利用无头浏览器模拟用户行为,实现基于行为仿真的凭证提交过程,通过对行业数据集 D1 和定制数据集 D2 进行三次实验取平均值计算时间开销。

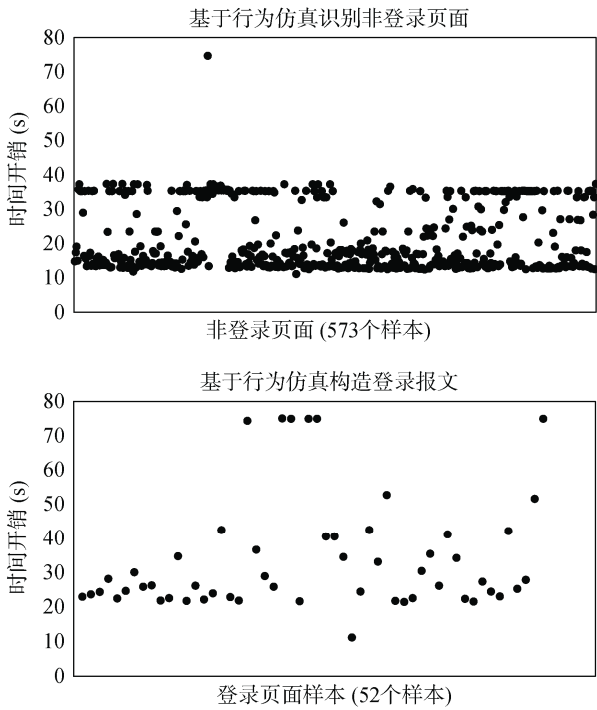


图 8 行业数据集时间开销

Figure 8 Time Overhead of Industry Datasets

行业数据集中,所有样本的时间消耗范围是 10~75 秒,如图 8 所示。可以观察到,对于被识别为非登录页面且无法成功构造报文的样本,其时间开销集中在 10~20 秒(占有所有样本的 66%)和 30~40 秒,占有所有样本的 26%。而对于成功构造登录报文的样本,时间消耗相对分散,但有 61%的样本集中在 20~30 秒。进一步分析发现,非登录页面中,时间开

销较低的样本通常在标签定位或登录行为链构造阶段便被识别为非登录页面, 而时间开销较大的样本, 则需要对登录行为链进行仿真和差分报文分析。

定制数据集中, 登录页面的时间开销分布如图 9 所示。根据实验结果, 未成功识别的样本表现出较低且波动较小的时间开销。大部分未成功识别样本(flag:0)的时间开销低于 30 秒, 其原因为这些网页在标签定位或凭证提交等前置未检测出登录页面特征, 无法构建登录行为链, 从而提前结束了流程。在无防护登录场景(flag:1)中, 平均时间开销为 38.91 秒, 而在有防护登录场景(flag:2)中, 平均时间开销增加至 43.50 秒。其中, 有防护的单数据包登录(Packet:1)平均开销为 40.84 秒, 而有防护的多数据包登录(Packet:n)平均开销进一步增加至 52.24 秒。此外, 无防护登录、有防护单数据包登录和有防护多数据包登录的时间开销中位数逐渐升高, 同时离散程度增大。这表明防护机制的引入和登录复杂性的增加对响应时间产生了累积影响。在有防护的多数据包登录中, 极端值明显增多。这种现象归因于多数据包验证过程中的额外报文构造开销以及潜在的网络负载对登录性能产生了影响。

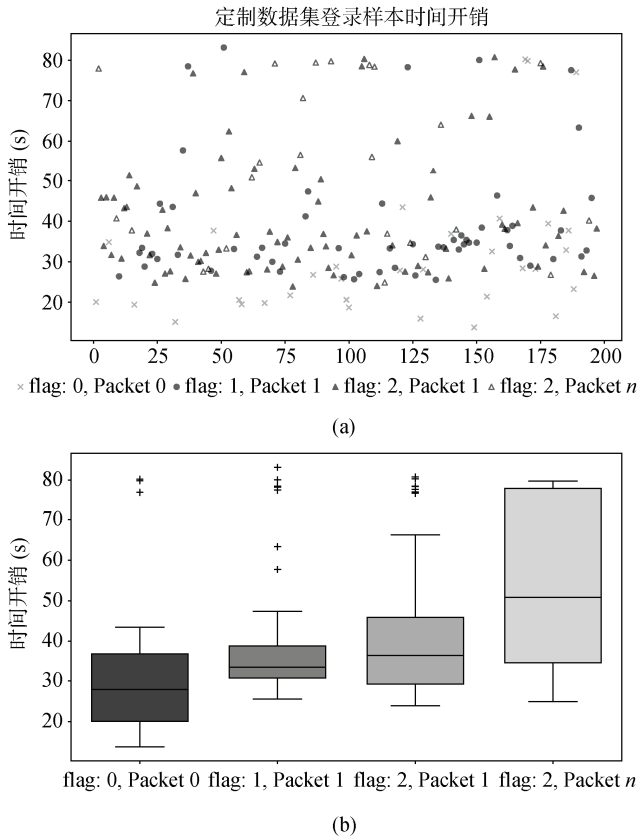


图 9 定制数据集登录样本时间开销

Figure 9 Time Overhead of Login Samples in Custom Datasets

定制数据集中, 非登录页面的开销普遍较低, 平均为 24.21 秒。如图 10 所示, 大部分样本的时间开销集中在 30 秒以内, 反映了大量非登录页面在标签定位和凭证提交阶段能够快速识别的特点。具体来说, 由于非登录页面通常不涉及复杂的登录交互, 它们在前期检测过程中能够被较快地排除, 从而使整体时间开销较低。其中, 开销较大的正确识别样本, 通过了基于行为仿真的凭证提交阶段, 在差分报文分析阶段才被识别为非登录页面。此外, 产生误报的网页(即被错误判定为登录页面的非登录页面)通常表现出较大的时间开销。这是因为这些页面必须先后通过基于行为仿真的报文构造过程和差分报文分析阶段, 并成功构造报文。误报的产生会增加报文重构过程中的时间成本。

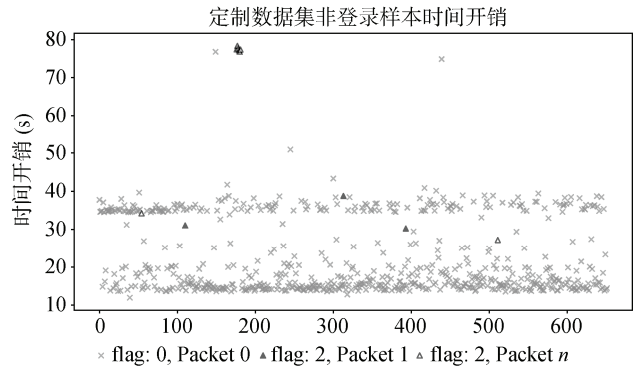


图 10 定制数据集非登录样本时间开销

Figure 10 Time Overhead of Non-Login Samples in Custom Datasets

上述实验结果显示, 成功构造报文的平均时间开销为 33.97 秒(D1)、40.84 秒(D2), 而识别为非登录页面的平均时间开销为 21.08 秒(D1)、24.21 秒(D2)。相比之下, 类似的自动化登录工具如 Shepherd 和 Drakonakis 等人^[26]的工作, 在扫描站点并提交凭证的时间成本分别为 75 秒和 90 秒。

为了进一步评估报文重构的时间开销, 本文在本地搭建 PIKACHU 靶机, 并针对不同的报文重构方式、凭证数量和线程数, 对两种报文重构方式的时间开销进行了全面评估。具体而言, 时间开销的计算方法为: 登录页面识别时间与凭证检测时间之和除以凭证数量。需要注意的是, 本实验, 登录页面的识别时间为 16 秒。

从表 6 的数据可以看出, 在同一报文构造类型下, 凭证数量的增加显著降低了单条凭证的时间开销。这一现象表明, 随着样本量的增加, 报文重构的时间成本呈现出规模效应, 单凭证的处理时间逐渐缩短。在相同凭证数量的情况下, 基于内容重写的报

文重构方式相比基于行为重构的报文重构方式,展现出了显著的时间优势。具体来说,在单线程下对120条凭证进行评估,基于行为重构的时间开销是基于内容重写的45.6倍。多线程技术能够有效提高报文重构的效率。在实验中,当线程数从4增加到8时,报文重构的时间开销有所下降,但基于行为重构的报文重构方式的优化幅度有限。原因在于,随着线程数的增加,无头浏览器的报文构造能力和网络负载逐渐成为瓶颈,限制了性能的进一步提升。值得补充的是,实验中,本文在凭证数据集中设置了三个有效凭证,均成功检测出,确保了报文重构的有效性。

表 6 报文重构时间开销

Table 6 Time Overhead of Packet Reconstruction			
类型	凭证数量	线程	时间/条
行为重构	60	1	12.53s
		4	4.43s
		8	3.83s
行为重构	120	1	11.86s
		4	3.48s
		8	4.03s
内容重写	120	1	0.26s
		4	0.17s
		8	0.15s
内容重写	1800	1	0.13s
		4	0.04s
		8	0.02s

最后,在实验中,本文对两数据集中能够成功构造登录报文的登录页面类型进行了分析与统计,具体统计结果见表7。分析表明,在行业数据集中,无防护登录页面的占比高达56%,而在定制数据集中有防护登录的页面的占比较高,但无防护登录页面的比例仍达到34.8%。

表 7 防护分析

Table 7 Protection Analysis		
	类型	占比
行业数据集	-无防护登录页面	56%
	-有防护登录页面	44%
	--含验证码登录页面	22%
	--多数据包登录	6%
定制数据集	-无防护登录页面	34.8%
	-有防护登录页面	65.2%
	--含验证码登录页面	27.8%
	--多数据包登录	15.8%

基于差分比较的报文重构策略选择方法,对于无防护的登录页面,可以采用基于内容重写的报文

重构方法。而对于有防护的登录页面,则需要依赖于行为重构的报文重构方法。需要注意的是,基于行为重构的报文重构方法时间开销更高,可以达到基于内容重写方法的数十倍。由于数据集中存在大量无防护登录页面,故在大规模口令检测场景中,可以优先采用基于内容重写的报文重构方法处理无防护登录页面。在不影响通用性的前提下,这一策略能够显著降低时间成本,从而显著提升检测效率。

6 总结与展望

本文提出了一种基于行为仿真的 Web 系统口令安全检测方法,旨在解决现有自动化检测技术在登录页面识别、报文构造通用性和大规模报文重构效率方面的不足。实验结果表明,本方法在多个关键指标上均表现出显著优势。登录页面识别方面,在两数据集上的F1分数分别达到93.46%和91.12%,显著优于现有方法 Shepherd。此外,该方法将误报率分别降低至0.52%和1.68%,进一步提升了检测的准确性。报文构造方面,本方法在应对多样化的 Web 系统登录形式(如单步、分段和弹出式登录)时展现了较强的适应性,特别是在处理动态渲染和有防护的复杂登录页面时表现优异。在两数据上的报文构造成功率分别达到了92.59%和86.27%。报文重构方面,基于差分比较的报文重构策略显著提高了无防护页面的检测效率,其中内容重写方法的时间成本仅为毫秒级。上述实验结果表明,本方法兼顾通用性和高效性,为 Web 系统的口令安全检测提供了一种新思路 and 实用工具,对提升信息系统的整体安全性具有重要意义。

尽管本方法在多个实验中表现出色,但仍存在一些局限性。首先,本方法无法有效识别含非输入验证码的登录页面,未来工作可以进一步探索更有效的验证码识别和处理机制。其次,受限于合规要求,本方法尚未在大规模口令爆破场景中进行验证,后续工作中可以寻找契机,在获得授权的前提下进行进一步的实验。

参考文献

[1] Wang D, Huang X Y, Shen C, et al. Preface of Intelligent Mobile Identity Authentication[J]. *Computer Science*, 2020, 47(11): 3-4. (汪定, 黄欣沂, 沈超, 等. 智能移动身份认证专题前言[J]. *计算机科学*, 2020, 47(11): 3-4.)

[2] Yan J, Blackwell A, Anderson R, et al. Password Memorability and Security: Empirical Results[J]. *IEEE Security & Privacy*, 2004, 2(5): 25-31.

[3] Understanding Brute Force Attack. Cloudflare. <https://www>

- cloudflare.com/zh-cn/learning/bots/brute-force-attack/. Oct. 2024.
- [4] What is Credential Stuffing. Cloudflare. <https://www.cloudflare.com/zh-cn/learning/bots/what-is-credential-stuffing/>. Oct. 2024.
- [5] Cost of a Data Breach Report 2024. IBM. <https://www.ibm.com/cn-zh/reports/data-breach>. Oct. 2024.
- [6] Anquanjiaojiao. China Network Security Corporation (CNSC). <https://cacts.cn/anquanjiaojiao.html>. Oct. 2024.
- [7] Chou H C, Lee H C, Hsueh C W, et al. Password cracking based on special keyboard patterns[J]. *International Journal of Innovative Computing, Information and Control*, 2012, 8: 387-402.
- [8] Li Y, Wang H N, Sun K. Personal Information in Passwords and Its Security Implications[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(10): 2320-2333.
- [9] Wang D, Wang P, He D, et al. Birthday, name and bifacial-security: understanding passwords of Chinese web users[C]. In: *28th USENIX Security Symposium*, 2019: 1537-1555.
- [10] Keeper Security. Understanding rockyou.txt: A Tool for Security and a Weapon for Hackers. <https://www.keepersecurity.com/blog/zh-hans/2023/08/04/understanding-rockyou-txt-a-tool-for-security-and-a-weapon-for-hackers/>. Oct. 2024.
- [11] Have I Been Pwned. Pwned Websites Database. <https://haveibeenpwned.com/PwnedWebsites>. Oct. 2024.
- [12] Liu G S, Qiu W D, Meng K, et al. Password Vulnerability Assessment and Recovery Based on Rules Mined from Large-Scale Real Data[J]. *Chinese Journal of Computers*, 2016, 39(3): 454-467. (刘功申, 邱卫东, 孟魁, 等. 基于真实数据挖掘的口令脆弱性评估及恢复[J]. *计算机学报*, 2016, 39(3): 454-467.)
- [13] Nisenoff A, Golla M, Wei M, et al. A Two-Decade Retrospective Analysis of a University's Vulnerability to Attacks Exploiting Reused Passwords[C]. *The 32nd USENIX Conference on Security Symposium*, 2023: 5127-5144.
- [14] Feng R H, Yan Z Y, Peng S Y, et al. Automated Detection of Password Leakage from Public GitHub Repositories[C]. *The 44th International Conference on Software Engineering*, 2022: 175-186.
- [15] Blazy. s0md3v. <https://github.com/s0md3v/Blazy>. Oct. 2024.
- [16] IoTSeeker. rapid7. <https://github.com/rapid7/IoTSeeker>. Oct. 2024.
- [17] Yu D, Zhang L L, Chen Y L, et al. Large-Scale IoT Devices Firmware Identification Based on Weak Password[J]. *IEEE Access*, 2020, 8: 7981-7992.
- [18] Xu S C, Chen Y L, Li Z, et al. Automatic Web Weak Key Detection for Internet of Things Devices[J]. *Computer Engineering and Design*, 2019, 40(1): 8-13. (徐顺超, 陈永乐, 李志, 等. 物联网设备 Web 弱口令自动化探测[J]. *计算机工程与设计*, 2019, 40(1): 8-13.)
- [19] Long X, Huang Y, Liu Z D, et al. WebCrack: Dynamic Dictionary Adjustment for Web Weak Password Detection Based on Blasting Response Event Discrimination[EB/OL]. 2022: arXiv: 2210.14582. <https://arxiv.org/abs/2210.14582>.
- [20] AWVS. Acunetix. <https://www.acunetix.com/vulnerability-scanner/>. Oct. 2024.
- [21] Nessus. Tenable. <https://www.tenable.com/products/nessus>. Oct. 2024.
- [22] Playwright. Microsoft. <https://github.com/microsoft/playwright>. Oct. 2024.
- [23] Van Acker S, Hausknecht D, Sabelfeld A. Measuring Login Web-page Security[C]. *The Symposium on Applied Computing*, 2017: 1753-1760.
- [24] Ghasemisharif M, Ramesh A, Checkoway S, et al. O Single Sign-Off, where Art Thou? An Empirical Analysis of Single Sign-on Account Hijacking and Session Management on the Web[C]. *The 27th USENIX Conference on Security Symposium*, 2018: 1475-1492.
- [25] Ghasemisharif M, Kanich C, Polakis J. Towards Automated Auditing for Account and Session Management Flaws in Single Sign-on Deployments[C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 1774-1790.
- [26] Drakonakis K, Ioannidis S, Polakis J. The Cookie Hunter: Automated Black-Box Auditing for Web Authentication and Authorization Flaws[C]. *The 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020: 1953-1970.
- [27] Jonker H, Karsch S, Krumnow B, et al. Shepherd: A Generic Approach to Automating Website Login[C]. *Proceedings 2020 Workshop on Measurements, Attacks, and Defenses for the Web*, 2020: 16-20.
- [28] Selenium. Selenium. <https://www.selenium.dev/>. Oct. 2024.
- [29] Senol A, Acar G, Humbert M, et al. Leaky forms: A study of email and password exfiltration before form submission[C]. In: *Proc. of the 31st USENIX Security Symposium*, 2022: 1813-1830.
- [30] Rutenstrauch J, Pellegrino G, Stock B. The Leaky Web: Automated Discovery of Cross-Site Information Leaks in Browsers and the Web[C]. *2023 IEEE Symposium on Security and Privacy*, 2023: 2744-2760.
- [31] Al Roomi S, Li F. A Large-Scale Measurement of Website Login Policies[C]. *The 32nd USENIX Conference on Security Symposium*, 2023: 2061-2078.
- [32] Senol A, Ukani A, Cutler D, et al. The Double Edged Sword: Identifying Authentication Pages and Their Fingerprinting Behavior[C]. *The ACM Web Conference 2024*, 2024: 1690-1701.
- [33] Paulson L D. Building Rich Web Applications with Ajax[J]. *Computer*, 2005, 38(10): 14-17.
- [34] Psaila G. Virtual DOM: An Efficient Virtual Memory Representation for Large XML Documents[C]. *2008 19th International Workshop on Database and Expert Systems Applications*, 2008: 233-237.
- [35] FOFA Search Engine. FOFA. <https://en.fofa.info>. Oct. 2024.
- [36] Burp Suite. Portswigger. <https://portswigger.net/burp>. Oct. 2024.
- [37] BugMeNot: share logins. BugMeNot. <https://bugmenot.com>. Oct. 2024.
- [38] Chromedp. Chromedp. <https://github.com/chromedp/chromedp>. Oct. 2024.
- [39] EyeWitness. RedSiege. <https://github.com/RedSiege/EyeWitness>. Oct. 2024.



白小凡 于 2019 年在武汉大学信息安全专业获得学士学位。现在武汉大学网络空间安全专业攻读博士学位。研究领域为软件安全、Web 安全。研究兴趣包括: 软件安全、Web 安全、应用安全。CCF 学生会员。Email: bai.mountain@whu.edu.cn



王俊琦 于 2024 年在武汉大学电子信息专业获得硕士学位。研究领域为 Web 安全。研究兴趣包括: Web 安全。Email: junqi-wang000@sina.com



赵易如 于 2024 年在武汉大学网络空间安全专业获得博士学位。现任武汉大学武汉数学与智能研究院博士后。研究领域为漏洞挖掘、模糊测试、人工智能安全。研究兴趣包括: 漏洞挖掘、模糊测试、人工智能安全。Email: zhaoyiru@whu.edu.cn



王笑克 于 2020 年在武汉大学信息安全专业获得学士学位。现在武汉大学网络空间安全专业攻读博士学位。研究领域为软件安全、程序分析。研究兴趣包括: 软件安全、程序分析。Email: xkernel@whu.edu.cn



赵磊 于 2012 年在武汉大学信息安全专业获得博士学位。现任武汉大学网络安全学院教授、博士生导师。研究领域为软件及系统安全、二进制程序的安全分析、软件漏洞的自动化挖掘和分析。Email: leizhao@whu.edu.cn