基于博弈的Web应用程序中访问控制漏洞检测方法

何海涛，许可，杨帅林，张炳，赵宇轩，李嘉政
（燕山大学信息科学与工程学院，河北 秦皇岛 066004）

摘　要：针对工业互联网中程序的访问控制策略隐藏在源码中难以提取，以及用户的访问操作难以触发所有访问路径而导致逻辑漏洞的通用化检测难以实现的问题，将博弈思想应用于访问控制逻辑漏洞检测中，通过分析不同参与者在Web应用程序中对资源页面的博弈结果来识别漏洞，使得不同用户的访问逻辑能被有针对性地获取。实验结果表明，所提方法在开源的11个程序中检测出31个漏洞，其中8个为未公开的漏洞，漏洞检测覆盖率均超过90%。

关键词：Web应用程序安全；漏洞检测；访问控制漏洞；访问控制策略；博弈
中图分类号：TP393
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2024078

Game-based detection method of broken access control vulnerabilities in Web application

HE Haitao, XU Ke, YANG Shuailin, ZHANG Bing, ZHAO Yuxuan, LI Jiazheng
School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

Abstract: To solve the problem that the access control strategy of the program in the industrial Internet was difficult to extract from the source code, and that the user's access operation was difficult to trigger all access paths, which led to the difficulty of universal detection of logical vulnerabilities, game theory was applied to the access control logic vulnerability detection for the first time. The vulnerabilities were identified by analyzing the game results of different participants on resource pages in the Web application, so that the access logic of different users could be targeted to obtain. Experimental results demonstrate that the proposed method successfully detect 31 vulnerabilities, including 8 unreported ones, out of 11 open-source applications, with a detection range exceeding 90%.

Keywords: Web application security, vulnerability detection, access control vulnerability, access control rule, game

0　引言

随着信息技术和工业生产方式的不断发展，工业互联网已成为未来发展的重要方向。工业互联网通过融合工业制造与互联网技术，利用互联网平台连接工业系统中的设备、车间、工厂、员工和客户，实现信息共享和智能化，促进不同行业间的互通和资源共享。工业互联网充分利用了互联网的枢纽功能，推动工业领域向更智能、更高效的方向发

收稿日期：2023–10–31；修回日期：2024–02–01
通信作者：许可，xuke_kara@163.com
基金项目：国家自然科学基金资助项目（No.62376240）；河北省省级科技计划基金资助项目（No.226Z0701G, No.236Z0702G, No.236Z0304G）；河北省自然科学基金资助项目（No.F2022203026, No.F2022203089）；河北省创新能力提升计划基金资助项目（No.22567637H）；河北省高等学校科学技术研究基金资助项目（No.BJK2022029）
Foundation Items: The National Natural Science Foundation of China (No.62376240), S&T Program of Hebei Province (No.226Z0701G, No.236Z0702G, No.236Z0304G), The Natural Science Foundation of Hebei Province (No.F2022203026, No.F2022203089), Innovation Capability Improvement Plan Project of Hebei Province (No.22567637H), Science and Technology Project of Hebei Education Department (No.BJK2022029)

展。同时，工业界广泛采用Web应用程序以提升效率和便利性，但也带来了设备、网络和数据安全等多种问题。为了满足实时性和高可靠性的需求，安全功能（如身份认证和授权访问）可能需要做出妥协。然而，这可能会导致安全隐患，增加系统受到攻击或未经授权访问的风险。据调查[1]，2022年共发现了23 900个漏洞，其中44%的漏洞存在于Web应用程序中。根据应用安全项目发布的十大安全风险问题榜单（2021版）[2]，失效的访问控制漏洞已经从2017年的第五名升至首位，成为最严重和最普遍的安全问题。这类漏洞在Web应用程序中出现的频率高于其他安全问题，相关的通用漏洞披露（CVE, common vulnerabilities & exposures）数量达到34个。因此，通过有效减少访问控制漏洞数量可以显著提高Web应用程序的质量，并减少这些漏洞带来的潜在危害和损失。

为确保Web应用程序的安全性，严格管理用户权限变得至关重要。目前，访问控制的实现方式[3]主要包括自主访问控制、强制访问控制、基于规则的访问控制、基于属性的访问控制以及基于角色的访问控制（RBAC, role-based access control）[4]。其中，RBAC方法因其便捷性和易维护等特点，成为当前Web应用程序中主流的访问控制模式。

为了应对工业互联网中面临的访问控制管理策略挑战，Zaidi等[5]提出了一种将区块链技术与RBAC策略相结合的方法，以解决工业互联网环境中具有相同角色的多个用户同时访问多个资源可能引发的角色冲突问题。此外，Ma等[6]提出了工业互联网身份识别与解析系统的访问控制与权限管理异构兼容架构，以防止工业互联网中非法用户入侵或合法用户误操作造成的损害。

Zhong[7]对Web应用程序中频繁出现的访问控制漏洞进行了全面分析，采用基于动态分析的方法捕获程序执行过程中的数据和信息，例如，程序运行时生成的结构化查询语言（SQL, structured query language）语句和访问控制逻辑。Li等[8]基于SQL虚拟查询概念设计了SENTINEL，通过从应用程序正常执行每个SQL查询的响应中提取的用户参数、会话变量等，来表征应用程序状态和数据约束。随后通过观察应用程序与数据库之间的交互来检测漏洞，虽然SQL虚拟查询提高了查询效率，但是应用程序中含有大量SQL语句，可能导致时间和资源浪费。基于SQL虚拟概念的检测方法简单可控，适用于稳定流程任务，但在动态Web应用程序中，SQL语句状态多变且灵活，限制了该方法的应用范围。

因此，一些研究人员采用基于有限状态机（FSM, finite-state machine）的方法进行访问控制漏洞检测。Deepa等[9]将程序的预期行为建模为参数注释的FSM，先通过分析FSM推导出与输入参数、访问控制和工作流相关的约束条件，再通过模拟违反约束条件的漏洞攻击来检测访问控制漏洞。Li等[10]设计了基于FSM的LogicScope检测方法。

漏洞检测的关键是推断出访问控制策略，这与敏感数据和资源保护相关。Le等[11]使用机器学习技术进行推导，提出了半自动化框架，该框架通过挖掘访问日志的域输入规范策略，生成更多访问请求来应用于机器学习模型。然而，该方法的局限性在于需要大量访问控制策略才能实现自动化，并且仍需要开发人员进行判断。为了解决策略生成的难题，文硕等[12]设计实现了一种基于策略推导的测试用例生成方法。

为了解决访问控制策略隐藏在Web程序源码中难以提取的问题，研究人员基于静态分析技术开发了基于图模型的技术和基于上下文分析的方法。基于图模型的技术将访问控制过程解析为图论中的图形结构，而基于上下文分析的方法则通过提取与安全敏感操作相关的信息来避免直接推断策略。夏志坚等[13]提出了基于权限验证图的漏洞检测算法，该算法通过构建权限验证图并计算验证权限来检测访问控制漏洞。Son等[14]设计了RoleCast工具，通过分析敏感操作依赖文件的上下文为角色属性分类，并进行漏洞检测。Monshizadeh等[15]设计了自动检测工具MACE，该工具可对敏感操作进行分析。Pan等[16]改进了敏感操作分析算法，以适应不同的应用程序。然而，上述方法仍存在一些局限性，如误报、漏报、无法处理隐藏的动态行为以及无法捕捉复杂的程序结构或运行时上下文等问题，这些局限性限制了漏洞检测的准确性和全面性。

研究人员还可以借鉴博弈[17-18]思想进行漏洞检测。在博弈中，攻击者和访问者之间存在行为决策和利益争夺过程，而漏洞的本质即攻击者通过漏洞非法获取用户的隐私资源，因此基于博弈思想的漏洞检测方法可以进一步揭示漏洞产生的根本原因，

并提高漏洞检测的准确性和全面性。Sadineni 等[19]提出了一种基于分布式数据包跟踪的博弈论方法。Arisdakessian 等[20]将博弈论思想用于攻击检测和预防，通过分析和模拟攻击者与系统的博弈行为，可以提前预测潜在攻击并采取防御措施，从而减小攻击者对系统的影响。

此外，还可以针对筛选出的特权页面直接进行访问控制漏洞检测，这类页面通常是需要重点保护的，也较容易出现漏洞。Sun 等[21]通过分析源码获取特权页面，并主要针对特权页面进行漏洞检测，Gauthier 等[22]设计了 ACMA（access control model analyzer）访问控制模型分析器，Ren 等[23]提出了 DetAC 方法。

综上所述，本文提出了一种在 RBAC 模式下检测 Web 应用程序中访问控制漏洞的方法 DetBAC（detection of broken access control）。该方法将访问控制漏洞的检测问题抽象为资源博弈问题，以更好地理解漏洞的本质，即恶意用户对程序资源的非法利用；明确定义了攻击者和访问者之间的行为决策和利益争夺过程，从而有效地进行漏洞检测，并满足工业互联网环境中的逻辑安全和应用场景需求；通过以动态分析为主、静态分析为辅的程序执行流追踪映射方法，实现对 Web 应用程序访问控制过程的抽象表征和分析。同时，本文还设计了一种多类型漏洞检测方法，可以精准检测出不同类别的访问控制漏洞。本文的主要贡献如下。

1) 将博弈的思想用于检测访问控制逻辑漏洞，并通过 DetBAC 实现工业互联网中的访问控制逻辑漏洞检测。DetBAC 通过模拟不同参与者的博弈行为，全面深入分析角色、用户和资源页面之间的访问逻辑关系，并采用多种不同的博弈策略，以涵盖更多用户访问行为和漏洞产生行为，进而扩大漏洞的检测范围。

2) 提出了一种以动态分析为主、静态分析为辅的程序执行流追踪映射方法，用于构建访问控制博弈模型。该方法不需要事先知道完整的访问控制行为和规则库，即可得到程序中隐含的访问控制策略。

3) 构建了一个漏洞检测模型，并提出一种多类型漏洞检测方法，从漏洞产生的原理出发，准确地检测出程序中的各类访问控制漏洞，实现对不同类型访问控制漏洞的针对性检测。

本文对 11 个真实的开源 Web 应用程序进行检测，实验结果显示，DetBAC 成功检测出 31 个访问控制漏洞，每个程序的检测覆盖率均超过 90%，并且误报率低。此外，DetBAC 还发现了 8 个未公开披露的漏洞，证明了其具有良好的可扩展性和高重用性。

1　相关概念

RBAC[6]模式访问控制管理机制在当前主流 Web 程序中应用广泛，通过分层设置“角色-用户-权限”并制定访问控制机制对用户的访问请求进行身份验证，提供了一种灵活的权限管理方式。在 RBAC 模式中，权限被授予给角色，而角色被分配给用户，从而实现对 Web 应用程序权限的有效管理和控制。通常情况下，一个角色可以同时被分配给多个用户，一个权限也可以同时被授予给多个角色和用户，这些权限定义了用户和角色可以执行的操作。以下是与 RBAC 相关的一些术语。

1) 用户

用户定义了访问 Web 程序资源的实体，每个应用程序存在多个不同用户，用户集合表示为 $U=\{u_0,u_1,\cdots,u_i,\cdots,u_k\}(0\leqslant i<k)$，任意一个用户 $u_i\in U$。

2) 角色

在 Web 程序中，角色代表用户的身份。一般而言，角色可以分为匿名用户角色、普通用户角色和管理员用户角色。令角色集合 $R=\{r_i|i=0,1,2\}$，其中 r_i 代表角色，当 $i=0$ 时，r_0 为匿名用户角色，用户未登录即可查看部分应用程序信息；当 $i=1$ 时，r_1 为普通用户角色，可对网站进行部分操作；当 $i=2$ 时，r_2 为管理员用户角色，可管理整个 Web 应用程序。

3) 权限

权限定义了程序的可执行操作或可访问资源页面，本文将可执行操作映射为不同角色和用户对资源页面的访问权限。令权限集合 $P=\{p_i|i=0,1,2\}$，其中 p_i 表示角色 r_i 的权限组，p_2 拥有的权限最完整，p_1 次之，p_0 权限等级最低。

4) 资源页面

资源页面是指 Web 应用程序中可供访问页面的统称。在应用程序运行过程中，程序前端展示了不同用户可以访问的被授权的资源页面，这些页面用于直接与用户进行数据交互，并处理用户请求等一系列操

作，本文将其称为用户资源页面（UP, user-resources page）。而在 Web 后端程序中，本文将用于处理后台业务逻辑或封装资源且与用户不直接交互的页面称为功能资源页面（FP, function-resources page）。

博弈是指在一定规则约束下，各参与者靠自身所掌握的信息选择行动策略，以实现利益最大化的过程。本文将博弈引申到攻击者和访问者之间，依照不同策略访问 Web 应用程序资源页面，以实现收益最大化的过程。以下介绍一些与博弈相关的术语。

1) 完全信息博弈

完全信息博弈指每个参与者都拥有所有参与者的特征、规则及得益分数等信息的博弈。

2) 博弈模型

博弈模型用于描述参与者、目标、规则、策略和收益等要素，通常包含以下内容。

① 参与者：参与博弈的个体，主要分为攻击者和访问者，不同角色、不同用户均为一个参与者。

② 博弈目标：博弈中所争夺的目标资源。

③ 博弈规则：博弈进行的基本规则和约束。

④ 博弈行动：参与者在博弈中采取的决策和行动，也被称为策略。

⑤ 博弈收益：每个参与者在博弈中获得的收益。

2　基于博弈的访问控制漏洞检测方法

本节基于博弈的思想提出了一种检测 RBAC 模式下 Web 应用程序中访问控制漏洞的方法，并在工业互联网领域得到应用和扩展。图 1 为基于博弈的访问控制漏洞检测方法框架，该框架主要包括访问控制博弈模型构建阶段、访问控制策略提取和攻击博弈策略制定阶段以及漏洞检测阶段。将访问控制漏洞的检测问题抽象为资源的博弈收益问题，进一步检测出 Web 应用程序中的垂直越权和水平越权两类访问控制漏洞。

2.1　访问控制博弈模型

本节构建了一个访问控制博弈模型 G，并设计了一种基于动态分析辅以静态分析的程序执行流追踪映射方法，以实现模型的构建。该方法对 Web 应用程序中不同角色类型的用户行为操作进行分析和整理，并实现了程序在基于“角色–用户–权限收益”访问控制执行过程中的映射和表征。如图 2 所示，访问控制博弈模型的构建主要包括以下 3 个步骤。

1) 基于动态执行的过程流追踪：跟踪不同类型角色的用户在访问控制执行程序中的行为和操作过程，同时捕获程序资源页面，并为其赋予基于角色的差异收益值。

2) 基于静态源码的访问控制流分析：解析程序源码的路径，获取程序的所有静态资源页面，并赋予此类资源页面确定收益值。

3) 对上述 2 个步骤中获得的动态执行信息和静态源码分析结果进行整合，构建访问控制博弈模型 G。

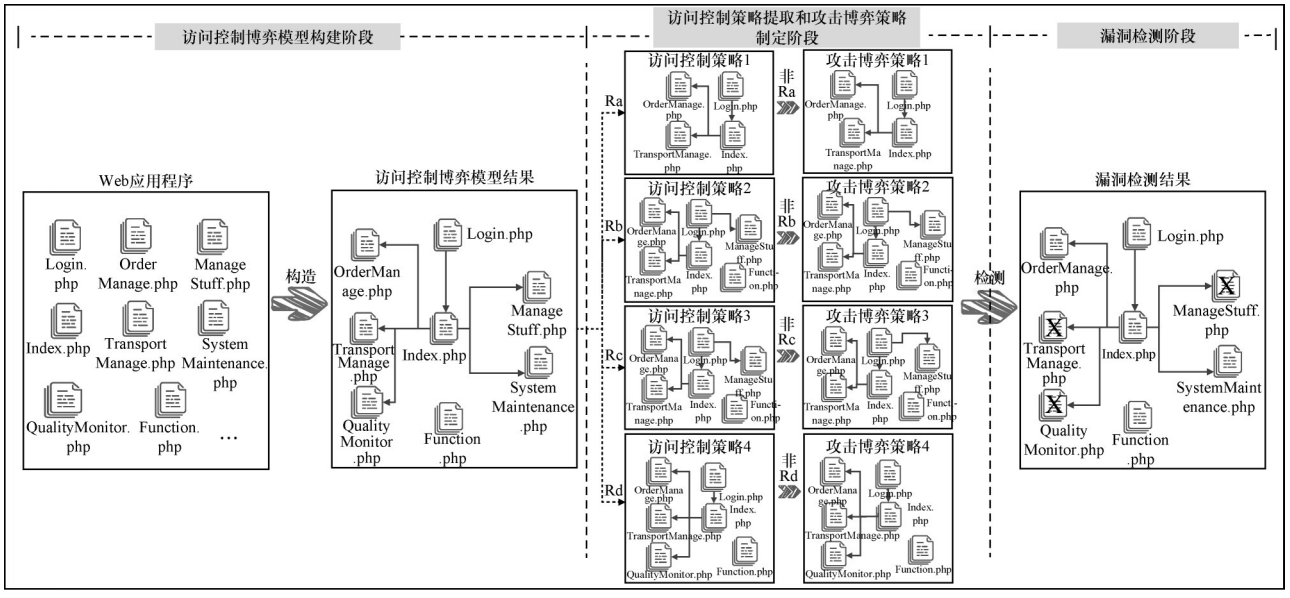


图 1　基于博弈的访问控制漏洞检测方法框架

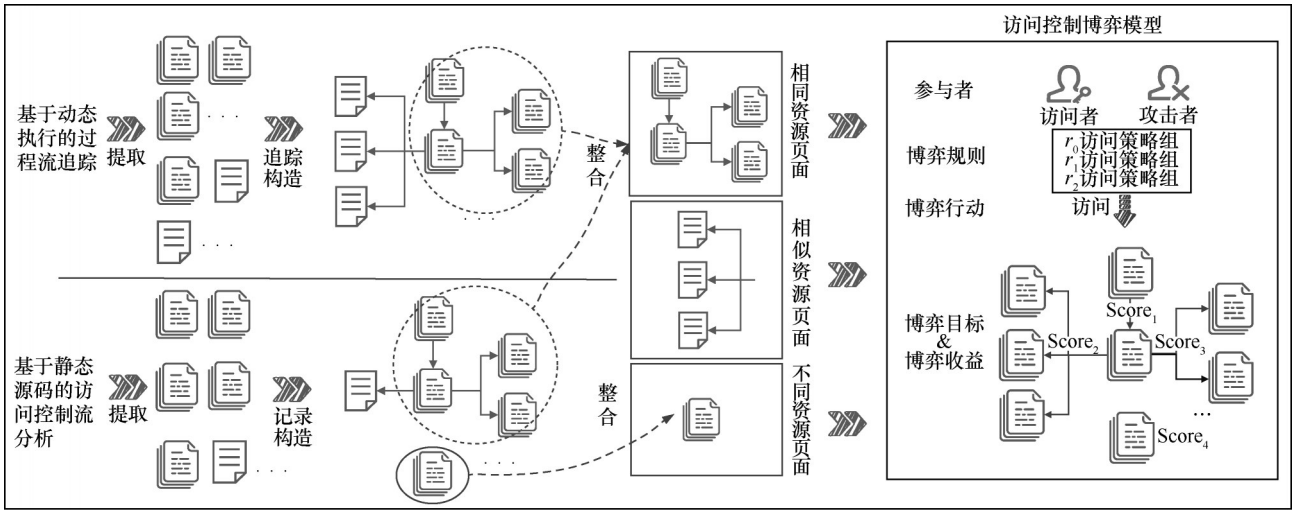


图2　访问控制博弈模型构建

2.1.1　基于动态执行的过程流追踪

在分析目标Web程序前，首先提取Web应用程序数据库中关于角色和用户之间关联关系的数据，并将其标记为[*r*:*u*]，以确定角色的类型和不同用户所属的角色。然后根据应用程序的入口链接URL（如登录界面节点login.php、程序主界面节点index.php等）和用户的账号密码信息，通过深度优先访问方式模拟Web应用程序的访问过程，并记录相应的页面响应内容。最后以结束节点（如logout.php退出链接）结束访问过程。在模拟访问过程中，全程跟踪记录不同角色类型的用户对程序的可访问操作和页面的请求响应结果。

本文将不同角色用户的所有预期操作所访问的UP的链接表示为n_i，并为捕获到的资源页面设置基于角色的收益值。这些值应为正数，并且与不同角色相关联。角色等级越高，被该角色访问的资源页面的收益值越高，以确保高权限角色的最小收益总值远大于低权限角色的最大收益总值。因此，本文选取任意底数的指数函数来初始化角色(r_2,r_1,r_0)相关的3类资源页面的收益值$(\text{score}_{r_2}, \text{score}_{r_1}, \text{score}_{r_0})$。

本文在指数函数e^x上选取3个特定的值，并基于角色的权限大小设定可被角色r_2访问的资源页面的收益值为幂运算的正指数$e^x(x>0)$，设定可被角色r_1访问的资源页面的收益值为幂运算的负指数$e^x(x<0)$。由于可被角色r_0访问的资源页面为公共页面，不存在BAC漏洞发生的可能，因此该页面的收益值被设置为0。此外，为了扩大程序的漏洞检测范围，本文还对无角色属性的FP进行检测，设其收益值为幂运算的零指数$e^x(x=0)$。最终构建动态执行的过程流追踪四元组G_0

$$G_0 = \sum\{ n_i, r_j, \text{score}_{r_j}, W \} \tag{1}$$

$$W = \sum w_k = \sum\{[\text{var} : \text{val}]\} \tag{2}$$

其中，$n_i \in \text{UP}$，r_j为角色属性，可能的取值为r_0, r_1, r_2，score_{r_j}为页面基于角色r_j的收益；w_k记录响应页面中一个参数var及其对应的参数值val，最终构成参数和值集合W。

2.1.2　基于静态源码的访问控制流分析

动态分析方法存在两个不足：程序页面节点覆盖不全；在进行建模时，过于依赖应用程序的实际运行环境。这将导致两个问题，一个是节点未能完全覆盖，无法全面捕获应用程序的行为；另一个是由于环境依赖性，无法在不同环境和时间下进行重复建模和检测。因此，本节结合静态分析方法覆盖范围广和稳定性强的优势，通过解析程序源码的文件根目录，进行目录层级的页面资源扫描，得到访问控制流分析二元组G_1

$$G_1 = \sum\{ n_i, \text{score}_\phi \} \tag{3}$$

其中，$n_i \in \text{UP} \cup \text{FP}$。与2.1.1节所得的四元组$G_0$相比，二元组$G_1$内存在3类UP的链接节点。1) 与$G_0$节点相同的链接节点，称为相同资源页面节点；2) 与G_0中前缀相同而参数不同的链接节点，称为相似资源页面节点；3) G_0中未发现的链接节点，称为不同资源页面节点。此外，本文还发现与访问控制验证相关的功能大多位于FP中，而FP通常在动态访问中不被发现，因此，FP集合也属于不同资源页面。

2.1.3 模型构建

本节通过融合上述两个元组构建访问控制博弈模型，以全面表征 Web 应用程序中的访问控制过程。在 G_0 中，相似资源页面的链接即使通过携带不同的参数表示执行不同的功能，也可能被识别为同一节点，但在 G_1 中，该类页面仅识别为单个链接节点。因此，G_0 模型中节点的状态和数量比 G_1 模型更复杂。为了解决该问题，需要对模型中的节点进行统合，并对该资源节点的收益进行整合。

针对 2 个表征元组中的 3 类节点，本文进行以下操作。

1) 相同节点和相似节点：保留 G_0 集合中节点状态更多的节点，以突显程序节点状态的多样性。将不同表征元组设置的收益值量化，并进行求和，用于计算每个资源节点的收益。

2) 不同节点：添加该节点及其收益值。

通过上述操作，可以全面表征程序为访问控制博弈模型 G

$$G = \{ n_i, R, \text{scores}, W \} \tag{4}$$

其中，$n_i \in \text{UP} \cup \text{FP}$。

2.2 策略获取

为了明确访问者和攻击者的博弈过程，并清晰地了解不同行动和策略对博弈结果的影响，本节进行了访问控制策略的提取和攻击博弈策略的制定，旨在为访问者和攻击者提供行动指南。针对不同的漏洞类型，在不同场景下帮助攻击者和访问者构建对应的行为模式，并做出最优的策略选择。

2.2.1 访问控制策略提取

为了从 Web 程序中提取出满足不同用户和资源间需求差异的策略，本节考虑角色类型和页面资源属性，从访问控制博弈模型 G 中提取访问控制策略 G_r，并将得到的结果直接作为访问者进行资源博弈的基础策略，以指导博弈过程中的行动，如式(5)~式(9)所示。

$$G_r = G_{r_0} + G_{r_1} + G_{r_2} + G_\phi \tag{5}$$

$$G_{r_0} = \sum\{ r_0, \text{score}_{r_0} + \text{score}_\phi, \sum\{[\text{var}:\text{val}]\}\} \tag{6}$$

$$G_{r_1} = \sum\{ r_1, \text{score}_{r_1} + \text{score}_{r_0} + \text{score}_\phi, \sum\{[\text{var}:\text{val}]\}\} \tag{7}$$

$$G_{r_2} = \sum\{ r_2, \text{score}_{r_2} + \text{score}_{r_1} + \text{score}_{r_0} + \text{score}_\phi, \sum\{[\text{var}:\text{val}]\}\} \tag{8}$$

$$G_\phi = \sum\{-, \text{score}_\phi, -\} \tag{9}$$

其中，$G_{r_0}, G_{r_1}, G_{r_2}$ 为不同角色的 UP 的访问控制策略，G_ϕ 为无角色属性的 FP 的访问控制策略。

图 3 为访问控制博弈模型 G 的示例，从该模型中提取出的访问控制策略为 G_r。示例中包含 8 个可访问资源页面，这些页面中的信息按照属性和有效信息整理为 3 个部分，分别是该页面允许访问的角色 R、访问该页面的收益 score、该页面内容解析后的参数及其参数值[var:val]。同时，该模型中可提取出访问控制策略 G_r。

2.2.2 攻击博弈策略制定

访问控制漏洞的根本原因在于攻击者通过模仿访问者正常的行为来窃取其他用户的资源。为了应对这些漏洞，本文在攻击者了解全部访问控制策略组合的情况下构建博弈策略，以实现资源抢夺。根据漏洞的产生原理，攻击者的资源博弈方式包括垂直越权（VPE, vertical privilege escalation）和水平越权（HPE, horizontal privilege escalation）。垂直越权是指低权限用户获取高权限用户的资源访问权限，从而执行高权限用户特有的操作。水平越权是指同一角色下的某一用户成功获取其他同级别用户

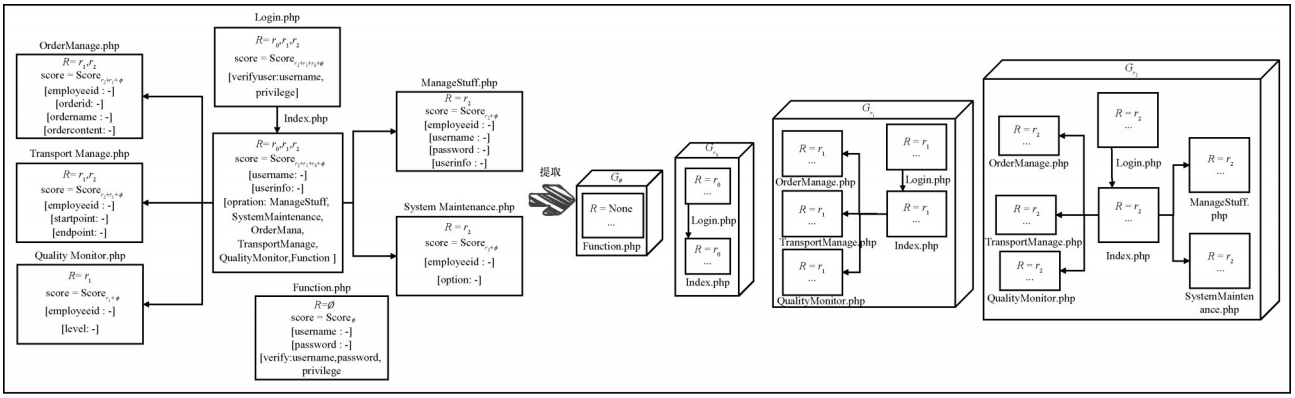


图3 访问控制策略 G_r 的提取过程

的资源访问权限。

综合上述两种漏洞产生方式制定攻击博弈策略，可有效减少冗余和无效的策略组合，具体如式(10)所示。

$$\Phi\text{atts} = \{\{n_a, < r_k, u_i >, \sum\{[\text{var}]\}\} \rightarrow \{n_a, < r_z, u_j >, \theta\_\text{score}, \sum\{[\text{var: val}]\}\} \mid n_a \in \text{UP} \cup \text{FP}, u_i, u_j \in U, r_k, r_z \in R, i \neq j\} \quad (10)$$

其中，$\{n_a, < r_k, u_i >, \sum\{[\text{var}]\}\}$表示攻击者及其所携带的信息；→表示模拟攻击者的博弈访问；$\{n_a, < r_z, u_j >, \theta_\text{score}, \sum\{[\text{var: val}]\}\}$表示攻击者所博弈的资源；$\theta$_score表示角色$r_z$的用户$u_j$的当前资源收益。考虑基于RBAC模式的Web应用程序中采用了角色属性的特点，本文设计了以下4种详细的攻击博弈策略组合，具体描述如下。

1) 匿名用户角色越级访问普通用户角色的资源页面，即当$i,j = 1, 2\cdots$、$k = 0$、$z = 1$时，得到垂直越权攻击博弈策略Φatts_{v_1}。

2) 普通用户角色或匿名用户角色越级访问管理员角色的资源页面，即当$i,j = 1,2\cdots$、$k = 0,1$、$z = 2$时，得到垂直越权攻击博弈策略Φatts_{v_2}。

3) 匿名用户越级访问程序中独立的资源配置页面[7]（如FP），即当$k = 0$、$z = \phi$时，得到垂直越权攻击博弈策略Φatts_{v_3}。由于配置页面属于管理员的权限范围内，若被匿名用户角色r_0成功访问，也属于产生了垂直越权漏洞，因此增设此博弈策略以增大程序的漏洞检测覆盖率。

4) 同级别角色的普通用户或管理员用户互相攻击访问彼此的资源页面，即当$i,j = 1,2\cdots$、$i \neq j$、$k = z = 1,2$时，得到水平越权攻击博弈策略Φatts_h。

上述4种攻击博弈策略中，前3种可触发垂直越权漏洞的产生，第4种可触发水平越权漏洞的产生。

2.3　漏洞检测

访问控制漏洞指的是在访问资源页面时，未授权的用户或非法攻击者通过某些方式访问其他用户的资源页面的行为。这种漏洞关注的是Web应用程序资源页面的所属权限，如果低权限的用户或角色成功访问到不属于它们的资源，就表示绕过了访问控制策略的限制，从而导致访问控制漏洞的发生。因此，漏洞检测的本质在于判断以下两点：1) 各个资源页面的权限设置是否符合程序正确的访问策略；2) 具有相同权限的角色所访问到的资源响应内容是否相同。简单来说，即判断访问者的访问控制策略和攻击者的攻击博弈策略所访问同一资源页面的页面响应结果是否相同。

本节构建了访问控制漏洞检测模型，如图4所示。首先模拟攻击者和访问者的身份，然后基于上文提取的访问者的访问控制策略G_r和制定的攻击者的4种攻击博弈策略Φatts_{v_1}、Φatts_{v_2}、Φatts_{v_3}和Φatts_h，对Web应用程序进行模拟访问。最后在该模型中设计了一种多类型漏洞检测方法，进行访问控制漏洞的检测。该检测方法主要包括公共页面筛选、收益评判检测机制和权值匹配检测机制3个步骤。

在进行公共资源页面筛选时，需要对目标资源页面进行判断，若为公共资源页面，则跳过。因为该类资源页面可被所有用户角色共同访问，不属于目标检测页面，但程序将会响应访问。若目标资源不为公共资源页面，则进行收益评判检测和权值匹配检测，具体细节在2.3.1节和2.3.2节进行解释。

2.3.1　收益评判检测机制

在模拟访问Web应用程序时，采用攻击博弈策略Φatts_{v_1}、Φatts_{v_2}、Φatts_{v_3}和不同角色的访问控制策略G_r，通过用户访问资源页面的所得分数来检测Web应用程序的安全性。如果攻击者和访问者对同一资源的收益值不同，则表示存在垂直越权漏洞。

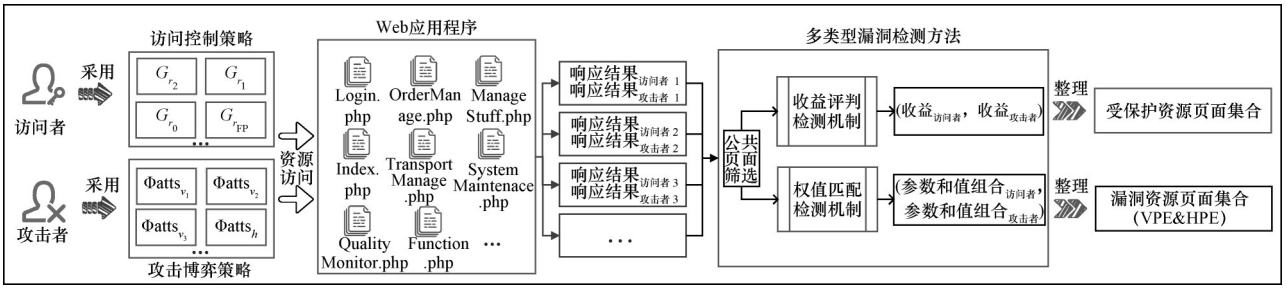


图4　访问控制漏洞检测模型

基于收益评判的检测机制如表 1 所示。其中，A、B 为不同角色的用户，并且 A 的角色等级高于 B 角色，页面资源的检测收益为 θ_score。基于不同类角色的收益值 θ_score 可详细分为：1) 采用 Φatts_{v_1} 攻击博弈策略，则 θ_score > score_{r_1}；2) 采用 Φatts_{v_2} 攻击博弈策略，则 θ_score > score_{r_2}；3) 采用 Φatts_{v_3} 攻击博弈策略，则 θ_score > score_{ϕ}。

表 1　　基于收益评判的检测机制

资源页面(R_A>R_B)		B	
		Visitor	Attacker
A	Visitor	(θ_score,0)	(θ_score,θ_score)

垂直越权的重点在于低角色权限对高角色权限的越权访问，因此高权限的用户 A 只作为访问者（Visitor），而用户 B 可以为攻击者（Attacker）或访问者（Visitor）。(θ_score,0) 表示当用户 A 与 B 均为访问者时，两者访问用户 A 的私有资源页面的收益分别为 θ_score 和 0，说明程序中设置了完善的访问控制机制，使得用户 B 不能访问用户 A 的私有页面，即不存在漏洞；反之，当用户 B 为攻击者时，两者访问用户 A 资源页面的收益为(θ_score, θ_score)，则说明 B 成功越权访问用户 A 的私有资源页面，进而获得该页面的收益值，即存在垂直越权漏洞。

最终资源博弈结果如式(11)所示，结果若为 θ_score，表示该资源页面无漏洞；若为 2θ_score，表示存在垂直越权漏洞。

$$\textbf{dectVPE} = \begin{pmatrix} \theta\_\text{score} \\ 2\theta\_\text{score} \end{pmatrix} \tag{11}$$

2.3.2　权值匹配检测机制

在该步骤中，采用攻击博弈策略 Φatts_h 和访问控制策略 G_r，从访问控制博弈模型 G 中获取每个资源页面的参数和值组合 W，以检测 Web 应用程序的安全性。W 中包括属性 var 和 val，通过比对同一参数 var 的值 val，如果值 val 相同，则表明当前资源页面存在水平越权漏洞；如果值 val 不同，则表明不存在漏洞。基于权值匹配的检测机制如表 2 所示。

表 2 中，A、B 为同类角色的不同用户，W 为参数值对[var:val]的集合。当用户 A 和 B 都是访问者时，由于其属于同一类角色，拥有的资源页面是相同的，但两者实际的页面访问结果应包含各自的隐私数据，使得页面的内容不完全相同，即 W 属性的参数 var 和值 val 不完全相同，如表 2 中 (W_A, W_B) 所示。然而，若用户 A 是攻击者，当用户 A 和 B 访问资源页面的结果完全相同时，则证明存在水平越权漏洞，如表 2 中 (W_B, W_B) 或 (W_A, W_A) 所示。基于漏洞的产生原理，不存在用户 A 和 B 均为访问者的情况，表示为—。最终资源博弈结果如式(12)所示，结果若为 $W_A + W_B$，表示资源页面不存在漏洞；若为 $2W_A$ 和 $2W_B$，表示存在漏洞。

表 2　　基于权值匹配的检测机制

资源页面(R_A=R_B)		B	
		Visitor	Attacker
A	Visitor	(W_A, W_B)	(W_A, W_A)
	Attacker	(W_B, W_B)	—

$$\textbf{dectHPE} = \begin{pmatrix} W_A + W_B & 2W_A \\ 2W_B & 0 \end{pmatrix} \tag{12}$$

图 5 展示了一个程序中访问控制漏洞检测的示例。在这个示例中，两个访问者 Visitor_a 和 Visitor_b 对 Web 应用程序进行访问，Web 应用程序内部进行用户的访问控制验证，包括身份验证和权限分配，并生成访问者的访问控制策略，攻击者 Attacker_c 和 Attacker_d 根据提取的策略设置并制定了相应的攻击博弈策略。图 5 右侧展示了提取的访问控制策略、制定的攻击博弈策略以及结果，虚线框表示通过对比后检测出的 VPE 和 HPE。

3　实验评估与分析

为了评估本文提出的漏洞检测方法 DetBAC 的性能，本节选取了 8 个基于 PHP 语言的真实 Web 应用程序进行评估，用于验证本文方法在真实环境中的可行性。其中，OpenIT 为工业互联网环境下用于实现员工管理和工业设备管理的真实 Web 应用程序。此外，考虑工业环境的复杂性可能导致更多非常规逻辑引发的漏洞[24]，为验证本文方法的拓展性和适用性，本节增设了靶场程序 DVWA、Bwapp 和 Wackopicko。这些靶场程序可适用于工业环境，帮助评估工业互联网中 Web 应用程序的安全性，特别是与访问控制相关的验证问题，并保证在不干扰工业设备运行的情况下进行相关的安全研究。评估对象包括程序构建的博弈模型 G 中提取的访问控制策略 G_r 的可扩展性、攻击博弈策略的行动执行效

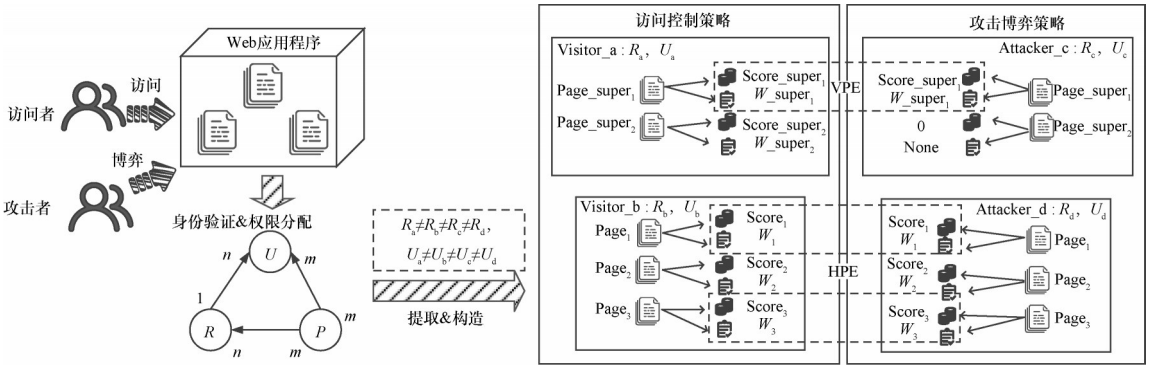


图5　访问控制漏洞检测示例

率以及漏洞检测模型的检测准确率、检测覆盖范围和检测速度。在此基础上，还与现有的一些访问控制漏洞检测方法进行对比，对比结果表明了本文所提漏洞检测方法的可行性。

3.1　被测Web程序概述

被测Web应用程序基础属性如表3所示，本文实验的11个开源Web应用程序均基于RBAC模式。其中，被测真实Web应用程序列为Web程序的名称，括号中标识了程序的版本号；角色和用户列分别为对应程序中存在角色的种类以及对应设置的不同用户数量；程序所包含的文件数量和数据库表数量如PHP文件和DB列所示。由于PHP5编程语言的版本较低，存在可能被探知的访问控制漏洞，易于验证方法的实用性和准确性，因此本文主要选取基于该语言编写的Web程序。同时，本文还选取了基于PHP7语言编写的Web程序用于探究DetBAC的泛化性和可扩展性，程序的版本如表3中PHP版本列所示。

3.2　访问控制策略结果分析

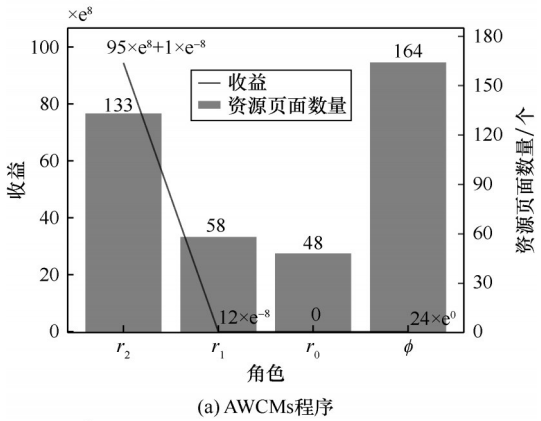
DetBAC对11个Web应用程序分别构建访问控制博弈模型，并从中提取每个角色访问者的访问控制策略G_r。策略的组成包括角色R、每个角色可访问到的资源页面数W（为可视化不同角色属性间资源的不同以及为后续检测做铺垫，将W属性简化为可访问资源的数量，其参数和参数值对的数量暂时不予计算）和每个角色的总收益值score。

图6展示了AWCMs和Events Lister程序的访问控制策略组成。其中，横坐标R属性具体包括r_2,r_1,r_0和ϕ；G_r中score属性标识为折线图，其值如左侧纵坐标所示；G_r中W属性标识为柱状图，其值如右侧纵坐标所示。

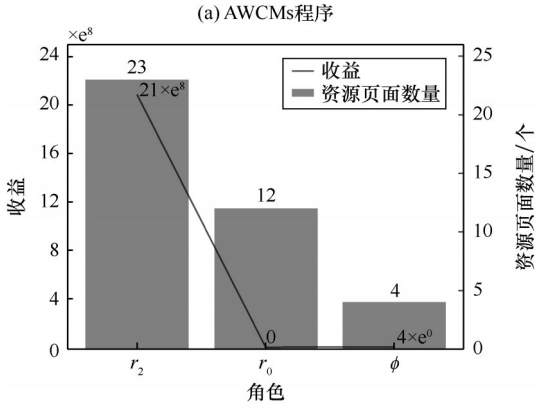
根据表3可知，AWCMs应用程序拥有3类角色(r_2, r_1, r_0)，Events Lister程序只包含两类角色(r_2, r_0)。根据角色类型进行划分，程序的页面数量按照权限等级的大小为r_2的页面数量最多，r_1次之，r_0最少。为了确保r_2角色的最小收益一定远高于r_1角色的最大收益，本文经计算发现，当指数$x = 8$时，更符合差异值大的要求。因此，最终设置r_2、r_1、r_0、ϕ与角色相关的资源页面的收益值分别为e^8、e^{-8}、0、e^0。

表3　　**被测Web应用程序基础属性**

被测真实Web应用程序	角色/类	用户/个	PHP文件/个	DB文件/个	PHP版本
MyBloggie (2.1.6)	3	4	59	4	5.3.1
Mybb (1.6.7)	3	4	436	75	5.3.1
Phpns (2.1.1 alpha)	2	4	59	13	5.3.1
AWCMs (2.2)	3	4	336	43	5.3.1
SCARF (1.0)	2	4	19	7	5.3.1
Events Lister (2.03)	2	2	28	3	5.3.1
DVWA (1.9)	2	2	119	2	5.3.1
Phpoll (097beta)	2	2	74	3	5.3.1
OpenIT (1.0)	3	4	195	27	5.3.1
Bwapp (2.6.184)	2	2	202	5	7.3.21
Wackopicko(1.0)	3	4	49	13	7.3.21



(a) AWCMs程序



(b) Events Lister程序

图6　访问控制策略组成

3.3　攻击博弈策略结果分析

本节评估了本文制定的攻击博弈策略的行动执行效率。如图7所示，与传统方法使用随机组合方式构建的策略数量相比，DetBAC制定的攻击博弈策略的策略数量明显更少。

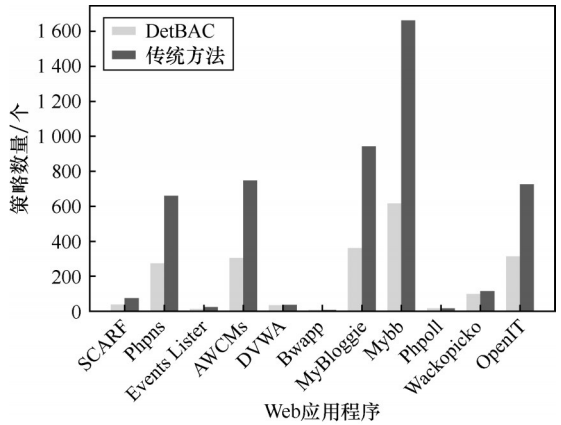


图7　攻击博弈策略生成数量对比

DetBAC方法的效果优于传统方法，主要原因是它从访问控制漏洞产生的原理出发，通过限制角色和资源之间的访问组合来模拟攻击者的访问行为和操作，从而制定更有效的博弈策略。而传统方法直接对角色和访问资源进行随机组合，容易产生大量冗余和无效策略，导致资源浪费。

从图7可以看出，MyBloggie和Mybb程序中的策略数量差异较明显，主要原因是：1) 这些程序的基础资源很多，导致随机组合的数量增大；2) 不同角色能够访问的资源数量差异明显，DetBAC基于漏洞的产生原理可以成功剔除多数无效策略。其中，无效策略包括符合程序权限分配的合法访问行为以及非向上访问的越权行为（超级用户管理普通用户的数据）。

3.4　漏洞检测结果

本节评估了DetBAC漏洞检测方法检测到的两种访问控制漏洞的检测准确率、漏洞检测覆盖率和检测速度。

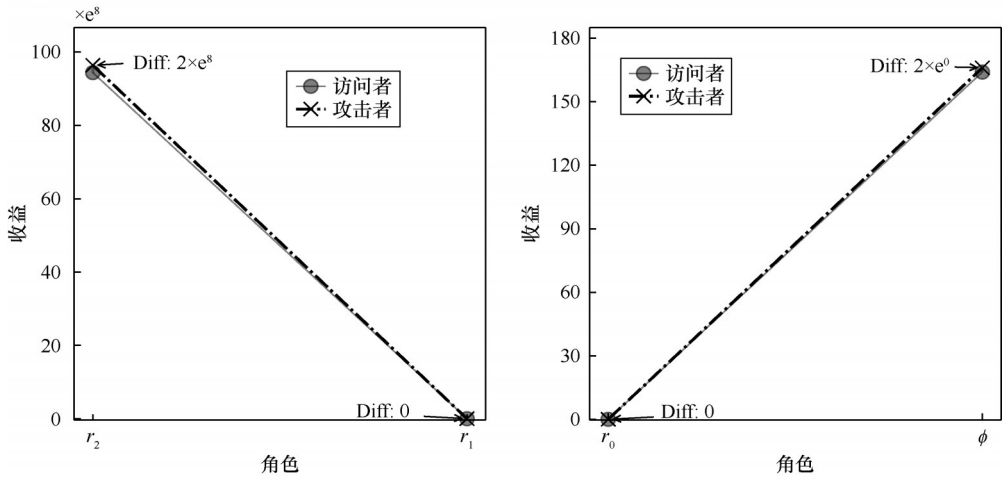
3.4.1　检测判别过程

AWCMs和Events Lister程序的垂直越权漏洞的检测结果如图8所示。其中，实线记录了不同角色访问者的实际收益总值，虚线记录了不同角色攻击者博弈后的收益总值，两条线之间若存在收益差值Diff，则存在垂直越权漏洞。由图8(a)可以看出，横坐标为角色r_2对应的访问者和攻击者的收益差值Diff为$2\times e^8$，表明攻击者成功采用2.2.2节的攻击策略$\Phi atts_{v_2}$找到两个VPE漏洞；横坐标为ϕ对应的收益差值为$2\times e^0$，表明攻击者成功采用2.2.2节的攻击策略$\Phi atts_{v_3}$找到一个VPE漏洞。图8(b)的分析同理，最终AWCMs程序的具体漏洞为“m_cp_avatar. php? KeepThis=true &TB_iframe=true& height=400&width=50”“member.php?id=1”“/install /index.php”“/control/db_backup.php”，Events Lister程序的具体漏洞为“/admin/user_add. php”“/admin/setup.php”和“/admin /add_user.php”。

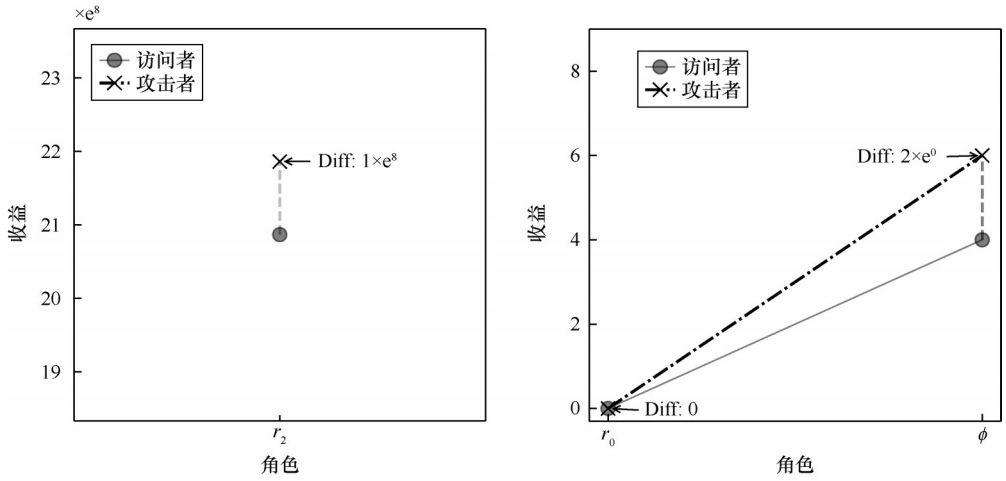
此外，DetBAC成功检测到同一角色下不同用户访问MyBloggie程序中“/blog. php/index. php?mode =editcom&post_id=1&comment_id=2/”的[var: val]数量为112，数量相同表示存在HPE漏洞。

3.4.2　漏洞检测结果及漏洞细节

漏洞检测结果如表4所示。其中，TP和FP分别表示程序中检测到的真实漏洞和误报漏洞的数量。Events Lister程序中存在一个误报，这是因为在初始设计中，“recover.php”页面可以在未经身份验证的情况下被访问，可能导致潜在的安全问



(a) AWCMs程序的垂直越权漏洞的检测



(b) Events Lister程序的垂直越权漏洞的检测

图8　垂直越权漏洞的检测结果

表4　　**漏洞检测结果**

被测Web应用程序	TP/个	FP/个	VPE/个	HPE/个	（已知，未知）/个	CVE_ID
MyBloggie	3	0	2	1	（3,0）	2007–3650
Mybb	2	0	2	0	（1,1）	2005–1833
Phpns	7	0	5	2	（7,0）	2008–6546
AWCMs	5	0	4	1	（2,3）	2010–1066 2010–4810
SCARF	3	0	3	0	（2,1）	2006–5909
Events Lister	3	1	3	0	（3,0）	2009–3168
DVWA	1	0	1	0	（1,0）	—
Phpoll	1	0	1	0	（1,0）	—
OpenIT	2	0	1	1	（2,0）	—
Bwapp	2	0	2	0	（0,2）	—
Wackopicko	2	0	1	1	（1,1）	—
总和	31	1	25	6	（23,8）	—

题。然而，该程序增加了邮箱密码重置认证方式，确保了在未经身份验证的情况下无法执行密码更改操作。但是本文方法未检测到这个防护措施，导致误报。

表 4 中的 VPE 和 HPE 列分别为垂直越权漏洞和水平越权漏洞的数量，（已知，未知）列列出了检测出的已知漏洞（即在 CVE 中报告或由其他研究报告检测所得的漏洞）和未公开披露的漏洞（未公开披露报告、未被修复以及其他工具没检测出的漏洞）。由检测结果可知，共检测出 31 个真实漏洞和 1 个误报漏洞，漏洞检测率为 96.9%，本文漏洞检测方法的检测能力较强，在检测出已知漏洞的基础上又检测出多个未公开披露的漏洞。此外，CVE_ID 列列出了程序中已知漏洞的参考情况，表明实验中所涉及的漏洞是真实存在的，并且已经被公开披露和记录在全球漏洞数据库中。本文漏洞检测方法可以覆盖更多资源页面，扩大了漏洞的检测覆盖率。同时，充分考虑访问控制漏洞产生的不同原因，并针对不同类型的漏洞采取针对性的检测方法。

3.4.3 漏洞检测效率评估

表 5 评估了漏洞检测覆盖率和时间效率，展示了 DetBAC 很好的漏洞检测能力和检测效率。在表 5 中，本文所使用的漏洞检测方法对开源的 11 个 Web 程序的漏洞检测覆盖率均可达到 90% 以上，同时针对不同访问控制漏洞类型的时间占用均较低。

表 5　　漏洞检测覆盖率和时间效率

被测 Web 应用程序	漏洞检测覆盖率	VPE 漏洞检测时间/s	HPE 漏洞检测时间/s
MyBloggie	90.00%	2.024	2.038
Mybb	100%	1.601	1.836
Phpns	100%	2.111	2.086
AWCMs	94.21%	1.751	1.923
SCARF	96.55%	1.951	1.937
Events Lister	91.67%	1.296	1.993
DVWA	93.94%	1.287	2.001
Phpoll	91.67%	1.188	1.762
OpenIT	91.80%	1.049	1.650
Bwapp	91.67%	1.171	1.189
Wackopicko	100%	1.369	1.359

由于 Web 应用程序中包含用于安装、错误处理等功能的代码，甚至存在不会被执行的代码，模型在预测漏洞时无法完全覆盖。因此，要达到对 Web 应用程序的漏洞检测范围完全覆盖较困难。另外，Web 应用程序的编写方式不统一。一些程序将每个用户的操作视为一个可访问的资源页面，而其他程序则采用在同一个资源页面中使用条件语句来实现多个用户操作的编写方式。

本文所提漏洞检测方法发现 Mybb、Phpns 和 Wackopicko 这 3 个程序采用简单方式来构建程序，并且没有发现无法执行的页面或无法访问的功能页面等情况，因此其漏洞检测覆盖率达到 100%。

3.4.4 检测结果对比

图 9 对 DetBAC 与现有的漏洞检测方法进行比较，仅以检测到的漏洞数量为评估指标。访问控制漏洞的检测方法缺乏统一的评判标准，并且由于本文所选的 Web 程序无法实现统一，这增加了其他指标的评估难度。相比而言，漏洞数量是一个相对容易衡量的指标，利用该指标可以对漏洞进行计数。因此，这个比较结果主要关注漏洞数量，而未考虑其他因素。

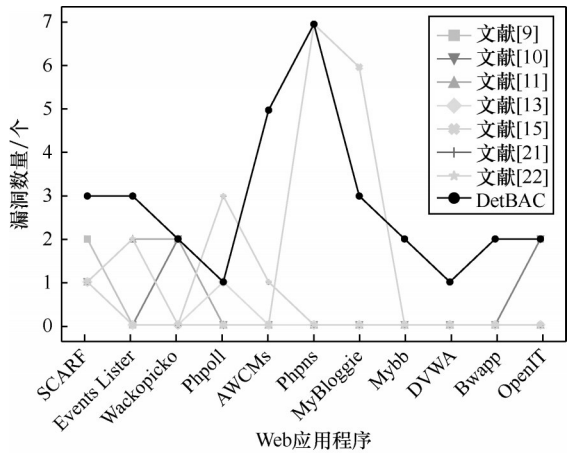


图 9　不同方法检测漏洞数量比较

本文选择包含更多通用 Web 应用程序的方法进行实验，主要原因是这些应用程序已经披露了已知 CVE 漏洞或在其他研究中得到了详细介绍。此外，为了展示实验的可扩展性和漏洞检测能力，本文引入 DVWA 和 Bwapp 两个靶场程序以及 Mybb 应用程序。

图 9 结果显示，在 Phpoll 和 MyBloggie 中，与其他方法相比，DetBAC 检测出的漏洞数量较少，主要原因是在寻找通用程序版本方面，DetBAC 遇到了困难，导致一些漏洞已被修复。未找到的漏洞包括 Phpoll(097beta) 程序“modifica_band.php”页面和“modifica_configurazione.php”页面中的访问

控制漏洞，以及MyBloggie(2.1.6)程序中“del.php”“delcat.php”和“deluser.php”中的访问控制漏洞。在其他9个程序中，本文方法仍然表现出较强的漏洞检测能力，并且相较于其他方法能够检测到更多漏洞。此外，本文方法能够成功检测出已知的漏洞，这进一步证明了该方法在提高Web应用程序安全性方面的有效性。

4　结束语

尽管已有许多研究关注于访问控制漏洞检测，但在降低误报率和提高漏洞检测率方面仍有待改进，基于此，本文提出了一种用于检测基于RBAC模式下Web应用程序中访问控制漏洞的方法DetBAC，该方法引入了访问者和攻击者之间的博弈概念，并将Web应用程序视为资源争夺的博弈。DetBAC方法能够深入理解访问控制漏洞的本质，并通过模拟博弈的方式更准确地分析访问者和攻击者的不同行为，以进一步实现程序中访问控制漏洞的检测。同时，该方法综合考虑了工业互联网中不同类型的访问控制漏洞和可能的场景，能够更全面地发现这些漏洞，以提高安全性和可靠性，并减少潜在的安全威胁和损失。实验结果表明，在对11个开源Web应用程序进行漏洞检测时，DetBAC能够有效检测出31个漏洞，准确率达到96.9%，检测覆盖率超过90%。

接下来的工作包括改进DetBAC方法，以解决存在的问题。未来工作将扩展DetBAC模型的适用范围，以处理大规模工业互联网中Web应用程序的复杂逻辑。同时，还将研究引入适用于大规模应用程序的技术和策略，确保模型的可扩展性和高效性。此外，将进一步提升漏洞检测的全面性和准确性。除了访问控制漏洞外，还将探索其他类型的漏洞检测能力（如输入验证漏洞），为工业互联网提供更全面、更精确的安全保障。

参考文献：

[1] 中国信息安全测评中心. 2022年网络空间安全漏洞分析研究报告[R]. 2022.
China Information Technology Security Evaluation Center. 2022 cyberspace security vulnerability analysis and research report[R]. 2022.

[2] SHAHID J, HAMEED M K, JAVED I T, et al. A comparative study of Web application security parameters: current trends and future directions[J]. Applied Sciences, 2022, 12(8): 4077-4100.

[3] BENANTAR M. Access control systems: security, identity management and trust models[M]. Berlin: Springer, 2010.

[4] BLUNDO C, CIMATO S, SINISCALCHI L. Managing constraints in role based access control[J]. IEEE Access, 2020, 8: 140497-140511.

[5] ZAIDI T, USMAN M, AFTAB M U, et al. Fabrication of flexible role-based access control based on blockchain for Internet of things use cases[J]. IEEE Access, 2023, 11: 106315-106333.

[6] MA B L, LIU Y, CHI C, et al. Research on access control and authority management of industrial Internet identification and resolution system[C]// Proceedings of the 2022 3rd International Conference on Electronics, Communications and Information Technology (CECIT). Piscataway: IEEE Press, 2022: 78-82.

[7] ZHONG L. A survey of prevent and detect access control vulnerabilities[J]. arXiv Preprint, arXiv: 2304.10600, 2023.

[8] LI X W, YAN W, XUE Y. SENTINEL: securing database from logic flaws in Web applications[C]//Proceedings of the Second ACM Conference on Data and Application Security and Privacy. New York: ACM Press, 2012: 25-36.

[9] DEEPA G, THILAGAM P S, PRASEED A, et al. DetLogic: a black-box approach for detecting logic vulnerabilities in Web applications[J]. Journal of Network and Computer Applications, 2018, 109: 89-109.

[10] LI X W, XUE Y. LogicScope: automatic discovery of logic vulnerabilities within Web applications[C]//Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. New York: ACM Press, 2013: 481-486.

[11] LE H T, SHAR L K, BIANCULLI D, et al. Automated reverse engineering of role-based access control policies of Web applications[J]. Journal of Systems and Software, 2022, 184: 111109.

[12] 文硕, 许静, 苑立英, 等. 基于策略推导的访问控制漏洞测试用例生成方法[J]. 计算机学报, 2017, 40(12): 2658-2670.
WEN S, XU J, YUAN L Y, et al. A test case generation approach for exploiting access control vulnerabilities based on policy inference[J]. Chinese Journal of Computers, 2017, 40(12): 2658-2670.

[13] 夏志坚, 彭国军, 胡鸿富. 基于权限验证图的Web应用访问控制漏洞检测[J]. 计算机工程与应用, 2018, 54(12): 63-68.
XIA Z J, PENG G J, HU H F. Detection of access control vulnerabilities in Web applications based on privilege verification graph[J]. Computer Engineering and Applications, 2018, 54(12): 63-68.

[14] SON S, MCKINLEY K S, SHMATIKOV V. RoleCast: finding missing security checks when you do not know what checks are[C]//Proceedings of the 2011 ACM International Conference on Object Oriented Programming Systems Languages and Applications. New York: ACM Press, 2011: 1069-1084.

[15] MONSHIZADEH M, NALDURG P, VENKATAKRISHNAN V N. MACE: detecting privilege escalation vulnerabilities in Web applications[C]//Proceedings of the Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2014: 690-701.

[16] PAN K F, WANG Q. Static detection of access control vulnerabilities in vue applications[J]. Journal of Physics: Conference Series. 2020, 1646(1): 012021.

[17] LI Q Y, LI Y, LIU S C, et al. Incomplete information stochastic game theoretic vulnerability management for wide-area damping control against cyber attacks[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2022, 12(1): 124-134.

[18] WANG J, GONG J X, LIN Z Q, et al. Multidimensional depth oriented fuzzing method of java Web applications[J]. Netinfo Security, 2024, 24(2): 282-292.

[19] SADINENI G, ARCHANA M, TANGUTURI R C. A highly efficient intrusion detection and packet tracking based on game theory approach[C]// Proceedings of the 2021 Emerging Trends in Industry 4.0 (ETI 4.0). Piscataway: IEEE Press, 2021: 1-5.

[20] ARISDAKESSIAN S, ABDEL W O, MOURAD A, et al. A survey on IoT intrusion detection: federated learning, game theory, social psychology, and explainable AI as future directions[J]. IEEE Internet of Things Journal, 2023, 10(5): 4059-4092.

[21] SUN F Q, XU L, SU Z D. Static detection of access control vulnerabilities in Web applications[C]//Proceedings of the 20th USENIX Conference on Security. Berkeley: USENIX Association, 2011: 1-16.

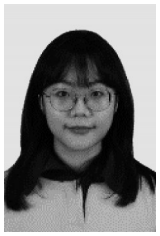
[22] GAUTHIER F, MERLO E. Fast detection of access control vulnerabilities in PHP applications[C]//Proceedings of the 2012 19th Working Conference on Reverse Engineering. Piscataway: IEEE Press, 2012: 247-256.

[23] REN J D, WU M Y, ZHANG B, et al. DetAC: approach to detect access control vulnerability in Web application based on sitemap model with global information representation[J]. International Journal of Software Engineering and Knowledge Engineering, 2023, 33(9): 1327-1354.

[24] ZHOU W, CAO C, HUO D D, et al. Reviewing IoT security via logic bugs in IoT platforms and systems[J]. IEEE Internet of Things Journal, 2021, 8(14): 11621-11639.

[作者简介]

何海涛（1968– ），女，云南禄丰人，博士，燕山大学教授、博士生导师，主要研究方向为人工智能、数据挖掘、软件安全和网络安全。



许可（1997– ），女，山西晋中人，燕山大学博士生，主要研究方向为网络安全。

杨帅林（1997– ），男，河北秦皇岛人，燕山大学博士生，主要研究方向为软件安全。

张炳（1989– ），男，湖北黄冈人，博士，燕山大学副教授、博士生导师，主要研究方向为网络安全。

赵宇轩（1997– ），男，河北秦皇岛人，燕山大学博士生，主要研究方向为时序数据挖掘。

李嘉政（1998– ），男，河北邢台人，燕山大学博士生，主要研究方向为网络安全和软件安全。