基于AI的业务逻辑漏洞安全架构研究

Research on AI-based Business Logic Vulnerability Security Architecture

刘　果[1]，李发财[2]，杨丽丽[1]，戚大强[1]，张　彬[1]（1.中讯邮电咨询设计院有限公司，北京 100048；2.中国联合网络通信集团有限公司，北京 100033）

Liu Guo[1], Li Facai[1], Yang Lili[1], Qi Daqiang[2], Zhang Bin[1] (1. China Information Technology Designing & Consulting Institute Co., Ltd., Beijing 100048, China; 2. China United Network Communications Group Co., Ltd., Beijing 100033, China)

摘　要：

在软件开发过程中，业务逻辑漏洞因其隐蔽性和复杂性，对系统安全构成严重威胁。综述了人工智能技术在业务逻辑漏洞检测中的应用，旨在提高业务逻辑漏洞识别的效率和准确性。介绍了业务逻辑漏洞的定义及其对软件开发的影响，详细探讨了模式识别、自然语言处理、数据流分析等技术在业务逻辑漏洞识别中的应用场景，展示了AI如何帮助开发者更准确地识别业务逻辑漏洞。最后讨论了这一领域未来的研究方向。

关键词：

软件开发；逻辑漏洞；AI技术；业务安全

doi：10.12045/j.issn.1007-3043.2024.08.009

文章编号：1007-3043（2024）08-0044-05

中图分类号：TP311.5

文献标识码：A

开放科学（资源服务）标识码（OSID）：

Abstract：

During the software development process, business logic vulnerabilities pose a serious threat to system security due to their hidden nature and complexity. It reviews the application of artificial intelligence technology in detecting business logic vulnerabilities, aiming to improve the efficiency and accuracy of identifying these vulnerabilities. Firstly, it introduces the definition of business logic vulnerabilities and their impact on software development. Next, it delves into the application scenarios of pattern recognition, natural language processing, data flow analysis, and other technologies in identifying business logic vulnerabilities, demonstrating how AI helps developers more accurately detect these issues. Finally, it discusses future research directions in this field.

Keywords：

Software development; Logical vulnerability; AI technology; Business security

引用格式：刘果，李发财，杨丽丽，等. 基于AI的业务逻辑漏洞安全架构研究[J]. 邮电设计技术，2024（8）：44-48.

0 引言

在当今软件开发领域，随着应用复杂性的不断增加，业务逻辑漏洞已成为一个日益突出的问题。这类漏洞不像传统的安全漏洞那样易于被识别，因为它们深藏于应用的业务流程之中，而非仅仅产生于技术层面的缺陷[1]。业务逻辑漏洞的存在可能导致数据泄露、功能滥用以及其他安全事件，严重威胁企业和用户的安全。

收稿日期：2024-06-07

随着人工智能技术的快速发展，其在软件安全领域的应用已成为研究热点。特别是在业务逻辑漏洞的检测中，AI技术能够通过学习和分析大量数据，识别出那些非直观的、复杂的业务逻辑错误。本文将深入探讨AI在此领域的应用，分析其对业务逻辑漏洞识别能力的提升，并讨论其实现的技术架构和面临的挑战。

1 业务逻辑漏洞的挑战

业务逻辑漏洞的检测面临若干挑战，首先是其隐蔽性强。此类漏洞通常不会直接违反常规的编程规

则或安全实践，而是隐藏在看似正常的业务流程中。因此，它们很难通过传统的安全工具进行检测，如静态代码分析或动态测试工具，这些工具更多地关注于代码的直接错误或已知的安全漏洞[2]。其次，业务逻辑的复杂性本身就是一个挑战。每一个业务流程可能都涉及多个组件和交互，而且可能因应用场景的不同而有所变化。这种多变性和复杂性使理解全局的业务流程并识别潜在的漏洞变得极其困难。最后，业务逻辑漏洞的修复往往需要对业务流程本身进行调整，这可能会涉及到跨部门的协调和复杂的代码修改。这不仅增加了修复的难度，也可能影响到业务的正常运行，从而在企业内部造成较大的阻力。

鉴于这些挑战，探索新的漏洞检测能力，对预防和应对业务逻辑漏洞具有至关重要的意义。

2 AI技术与业务逻辑漏洞检测

2.1 自然语言分析

自然语言处理（NLP）可以桥接业务需求与软件开发，打破需求分析人员与代码研发人员之间的壁垒，确保业务需求与实际编写的代码之间的一致性，减少需求设计及业务编码阶段的逻辑漏洞问题[3]。

首先，NLP可以自动化地进行需求提取和分析。通过对项目文档（如需求说明书、会议记录、详细设计等）的语义分析，NLP工具能够识别出关键需求以及核心业务流程，并将其转换为更结构化的格式，便于开发人员理解和跟踪。

其次，NLP技术能够在代码开发过程中提供实时反馈。利用AI模型对代码逻辑的理解并结合代码库中的注释，和原始需求进行比较，NLP工具可以检测出代码实现与原始需求之间的偏差。这种技术可以在开发早期及时发现潜在的业务逻辑漏洞，从而避免在项目后期进行昂贵的修改。

2.2 模式识别与异常检测

模式识别主要依赖于机器学习算法，通过训练模型识别正常的业务流程模式。一旦模型建立，它可以自动识别偏离正常模式的行为，这些行为可能指示存在业务逻辑漏洞。例如，如果一个电商网站的支付流程被异常地修改，模式识别技术可以帮助识别出这种非典型的支付行为，从而提示可能的漏洞[4]。

异常检测技术则专注于寻找在数据中不符合预期模式的异常行为。这种技术通常使用统计学方法来确定数据中的异常值。在业务逻辑检测中，异常检测能够有效地识别出在正常业务流程中不应该发生的行为，如意外的用户权限提升、不寻常的交易金额等。通过监控这些异常行为，可以及早发现潜在的业务逻辑漏洞[5]。

本章简述了软件开发不同阶段对业务逻辑漏洞的AI技术应用。下面将介绍基于AI技术的业务逻辑漏洞安全架构的整体设计。

3 架构设计

3.1 架构模块设计

本架构主要包括数据收集、分析处理、漏洞检测、报告与反馈、集成与部署等关键模块（见图1）。

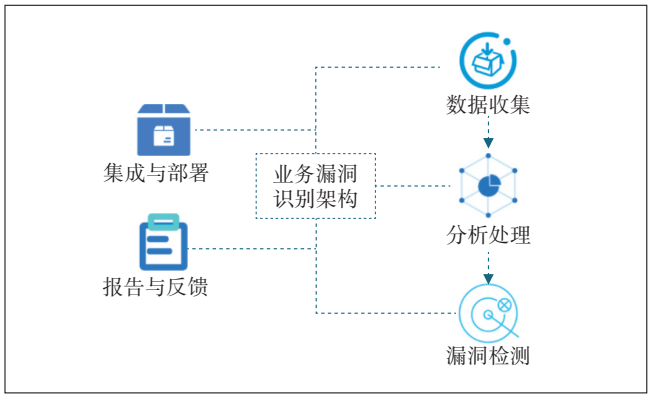

图1　核心功能模块

a）数据收集模块。它负责收集项目中的各种数据，包括代码库、开发文档、用户操作日志、业务数据等。这些数据不仅包括静态的文本信息，还应包括动态的行为数据，为后续的分析提供原始材料。

b）分析处理模块。通过自然语言处理和AI模型分析，对收集到的数据进行挖掘研判。这一模块的核心是从各类需求文档中提取出有价值的信息进行归纳总结和初步的分析判别，识别需求文档中可能存在的业务逻辑漏洞。

c）漏洞检测模块。一方面，基于自然语言处理技术和代码检测分析技术实现代码需求一致性检查和代码逻辑漏洞识别。另一方面，基于预先定义的业务逻辑模式和异常行为模式，使用模式识别和异常检测技术来识别可能发生的业务漏洞。此模块不断更新其检测算法，以适应新的漏洞和攻击手段。

d）报告与反馈模块。它将检测结果以报告的形式提供给开发者和安全专家，包括漏洞的详细信息、风险等级和修复建议[6]。同时，它支持反馈机制，用户

可以对漏洞报告提出疑问和修正意见，系统据此优化检测算法。

e）集成与部署模块。它确保安全架构可以无缝集成到现有的开发流程中，不干扰正常的开发活动。支持各种主流的开发工具和平台，如IDE、持续集成系统等。

3.2 架构模型设计

3.2.1 基础场景分析模型

基础场景分析模型旨在从需求文档中进行问题识别[7]，以便在开发早期分析出潜在的业务逻辑漏洞，从而提高业务流程的安全性和可靠性（见图2）。

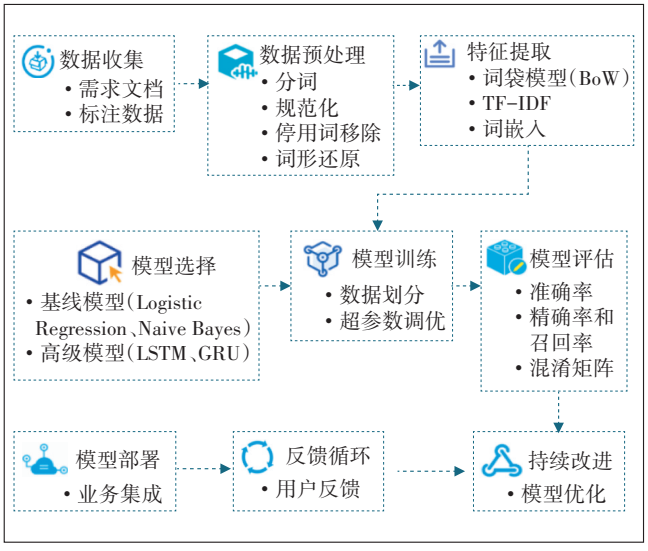

图2　基础场景分析模型

收集涉及暴露登录、横向越权、纵向越权、信息泄露、验证码绕过、短信炸弹等场景的需求设计文档[8]。手动标注一部分文档中的潜在漏洞，创建标注数据集。通过分词、规范处理、词形还原等处理方式形成标准数据。通过（Word2Vec、GloVe、BERT）预训练嵌入进行语义捕获。基线模型使用Logistic Regression、Naive Bayes或SVM等简单模型来建立性能基线，高级模型使用LSTM、GRU或基于Transformer的模型（如BERT、RoBERTa）来提高性能。

在训练过程中，通过训练基线模型了解初始性能，微调高级模型，使用预训练嵌入和迁移学习技术来完成预训练模型的适配。

3.2.2 一致性分析模型

一致性分析模型在需求文档的基础上结合源代码进行数据标注（见图3），并通过代码词法分析、代码

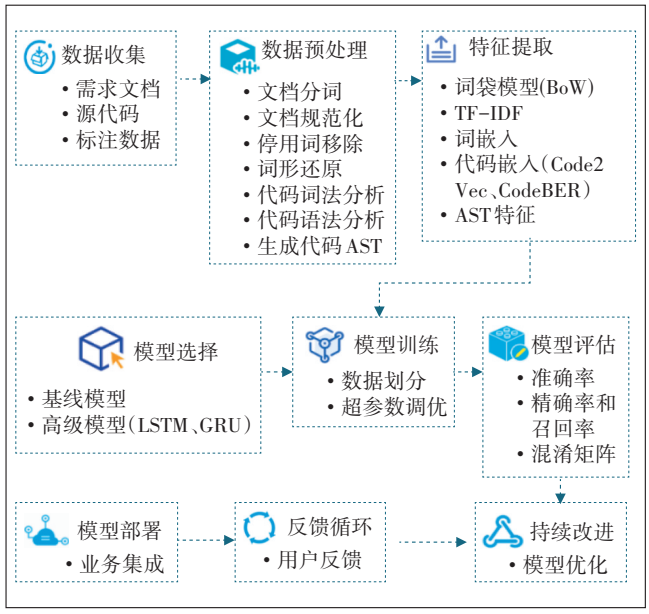

图3　一致性分析模型

语法分析等技术进行标准数据处理[9]。在词嵌入的基础上加入代码嵌入技术进行语义捕获以及代码向量生成。基线模型采用简单的相似性度量（如余弦相似度）完成基线建立，高级模型使用神经网络模型（如LSTM、GRU、Transformer）处理文本和代码嵌入。

在训练过程中，通过训练基线模型了解初始性能，训练高级模型，使用预训练嵌入，结合需求文档和代码的嵌入完成一致性比对训练。

3.2.3 异常分析模型

异常分析模型对业务流程数据和异常行为数据进行标注（见图4）。通过数据预处理和特征提取完成时序特征、行为特征和统计特征的提取。模式识别模型使用聚类算法（如K-means）或分类算法（如SVM、决策树）[10]。异常检测模型使用统计方法（如z-score）或机器学习算法（如Isolation Forest、One-Class SVM）。最终逐项完成模式识别模型、异常检测模型的训练。

3.3 架构流程设计

3.3.1 数据收集流程

数据收集模块通过不同的采集方式，采集源代码、需求文档、操作日志、业务数据等关键数据（见图5）。

采集能力通过统一的清洗和格式化处理，实现数据的过滤和治理。然后，对所有数据进行加密操作，保障数据的安全性。最后对不同类型的数据进行打标签操作，实现不同类型的数据分组和打包。为后续

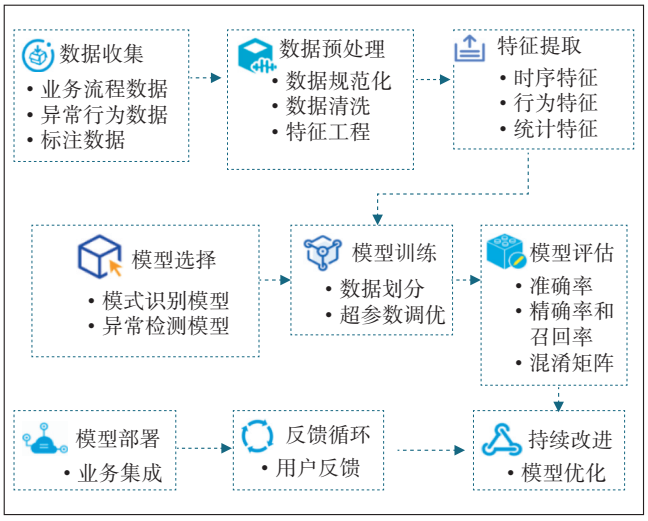

图4　异常分析模型

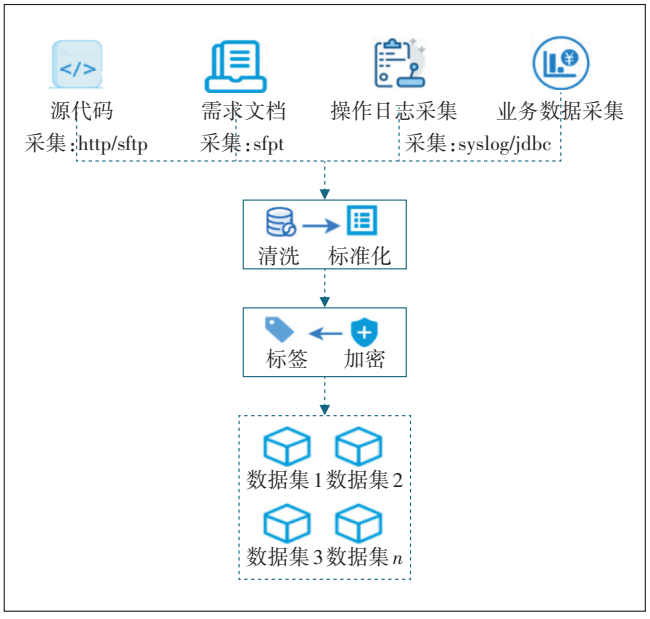

图5　数据采集流程

的关联分析处理做前置准备[11]。

3.3.2　漏洞分析流程

漏洞的分析检测是架构的关键业务流程（见图6）。此流程通过对格式化数据集的场景分析及业务研判实现对业务漏洞的初步研判，实现一些常规需求场景的漏洞过滤。结构化分析实现需求的规范化处理[12]，结合语义和语法分析后的代码数据实现对需求的一致性分析研判，实现代码需求间一致性的检测分析。同时，通过静态代码检测实现通用代码漏洞的增强，辅助规避业务逻辑漏洞。

异常检测及模式识别针对业务数据及业务调用

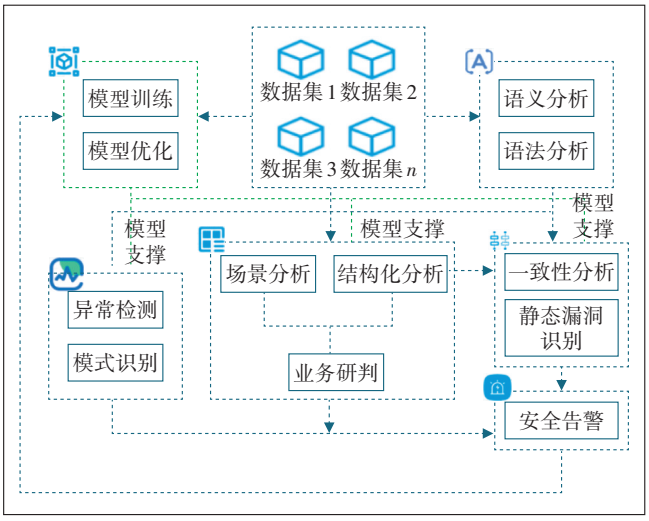

图6　业务漏洞分析流程

链进行分析[13]，通过对实时业务数据及业务链和常规基线数据之间的动态匹配，来分析生产运营过程中未能提前识别的业务漏洞。

3.4　架构设计优势

如图7所示，此架构通过数据收集、分析处理、漏洞检测、报告与反馈、集成与部署实现整个逻辑漏洞分析流程，涵盖需求设计、代码研发、发布部署、生产运营各个阶段。利用AI技术实现分析自动化、检测智能化，完成业务逻辑漏洞的综合检测[14]。

此架构具有如下优势。

a）全面性。架构通过整合代码分析、自然语言处理、流分析和行为模式识别等多种技术，实现对业务逻辑漏洞的全面检测。这种多维度的检测方式可以覆盖从代码层面到业务流程层面的各种潜在风险。

b）实时性。架构设计了实时数据监控和异常检测机制，能够即时发现并响应业务中的异常行为和潜在风险。这大大减少了漏洞被利用的时间窗口，增强了业务的安全防护。

c）智能化。利用先进的AI技术，自动学习和适应新的业务环境和开发模式。通过持续的学习和优化，架构的检测能力会随时间的推移而不断提高，减少人工干预的需求。

虽然此架构拥有自身的优势和特点，但业务的演进要求我们持续对架构进行优化和改进。

4　未来研究方向

4.1　交叉领域技术

将AI技术与其他领域（如心理学、社会学和经济

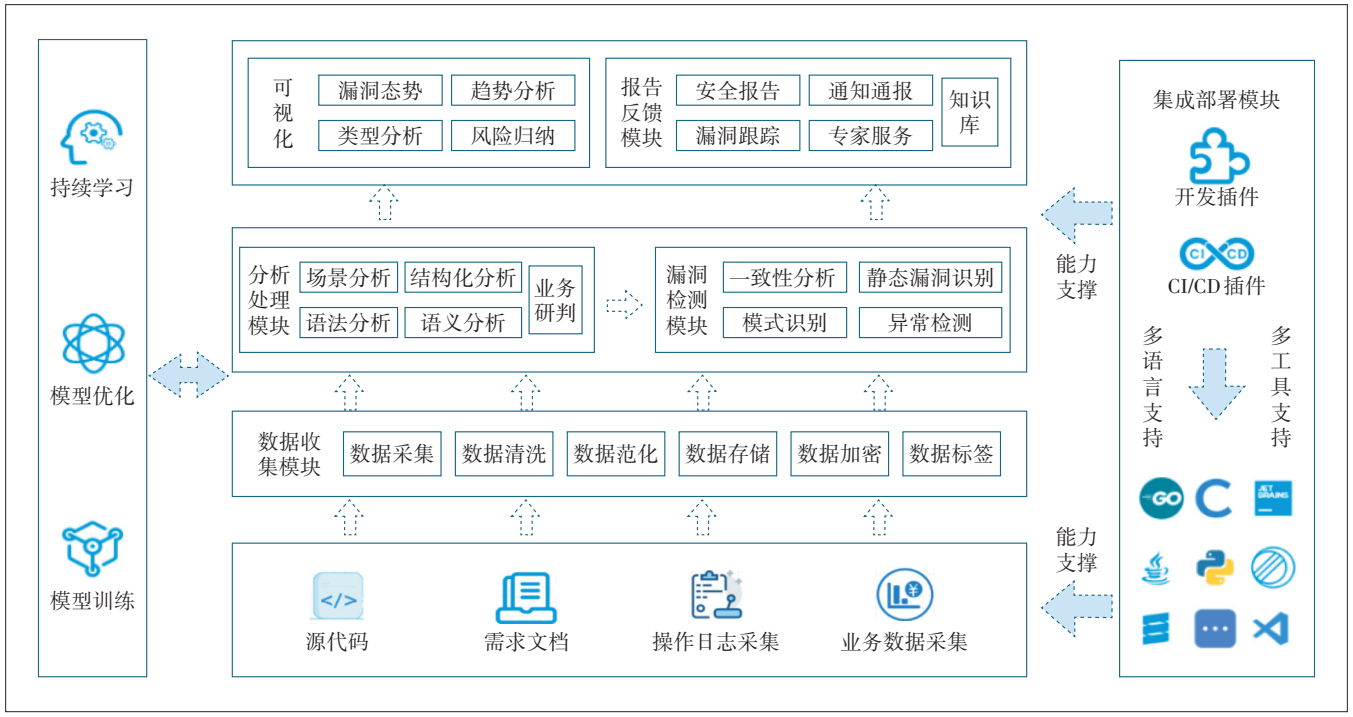

图7　业务逻辑漏洞检测架构

学等领域）的理论和方法结合，可能开辟新的研究路径。例如，理解用户行为的心理学原理有助于更准确地识别和预防那些依赖于用户交互的业务逻辑漏洞[15]。

4.2 高级模型技术

研究如何将不同AI技术（如深度学习、符号逻辑推理等）更有效地融合，以强化系统在处理复杂业务逻辑时的准确性和灵活性。模型融合可以提高漏洞检测的覆盖面，减少误报率。

参考文献：

[1] 张志一. 网络安全技术研究[J]. 中小企业管理与科技（下旬刊），2009(11)：268.
[2] 汪楚娇，蒋志雄，王拓，等. 基于模糊数学的网络安全风险评估模型[J]. 网络安全技术与应用，2003(10)：22-25.
[3] 杨福义，叶其松. 人工智能时代知识工程的初步探索[J]. 人工智能与机器人研究，2021，10(1)：9-28.
[4] 张钹，朱军，苏航. 迈向第三代人工智能[J]. 中国科学：信息科学，2020，50(9)：1281-1302.
[5] 王铮. 计算机网络安全漏洞分析及防范对策探讨[J]. 电脑知识与技术，2020，16(29)：55-56.
[6] 李江灵. 计算机网络安全中漏洞扫描技术的研究[J]. 电脑编程技巧与维护，2021(6)：168-169.
[7] 贾焰，方滨兴. 网络安全态势感知[M]. 北京：电子工业出版社，2020：10-13.
[8] 谢丽霞，江典盛，张利，等. 漏洞威胁的关联评估方法[J]. 计算机应用，2012，32(3)：679-682.
[9] 诺意曼. 计算机与人脑[M]. 甘子玉，译. 北京：商务印书馆，1965：58.
[10] 宋培彦，路青，刘宁静. 一种从术语定义句中自动抽取知识单元的方法[J]. 情报杂志，2014(1)：139-143.
[11] 张炳，任家东，王苧. 网络安全风险评估分析方法研究综述[J]. 燕山大学学报，2020，44(3)：290-305.
[12] 席荣荣，云晓春，金舒原，等. 网络安全态势感知研究综述[J]. 计算机应用，2012，32(1)：1-4，59.
[13] GARG H，KUMAR K. Some aggregation operators for linguistic intuitionistic fuzzy set and its application to group decision-making process using the set pair analysis[J]. Arabian Journal for Science andEngineering，2018，43(6)：3213-3227.
[14] 莫媛淇，陈智慧. 信息通信网络安全威胁与漏洞分析[J]. 电子元器件与信息技术，2021，5(7)：247-248，250.
[15] 刘云，薛盼盼，李辉，等. 基于深度学习的关节点行为识别综述[J]. 电子与信息学报，2021，43(6)：1789-1802.

作者简介：
刘果，工程师，学士，主要从事SOC平台、态势感知、大数据治理平台相关网络安全的研究工作；李发财，高级工程师，硕士，主要从事网络安全产品研究、架构设计、技术选型等工作；杨丽丽，工程师，学士，主要从事抗DDoS、漏洞扫描、网站安全监测等技术方向的研究工作；戚大强，工程师，学士，主要从事网络安全技术的研究工作；张彬，工程师，硕士，主要从事网络安全大数据分析方向的研究工作。