一种自动构建STRIDE威胁规则模型和更新规则库的方法

付昌兰[1),2)] 张 贺[1),2)] 管兴政[1)] 李凤龙[3)]

[1)](南京大学软件学院 南京 210023)

[2)](计算机软件新技术国家重点实验室(南京大学) 南京 210023)

[3)](华为云计算技术有限公司 杭州 310053)

摘 要 威胁建模是一种识别和应对威胁的结构化方法,STRIDE方法在实践中已经成为事实上的主流威胁识别技术。目前,对STRIDE威胁的分析和威胁识别规则的构建在很大程度上依赖于人类的专业知识,导致威胁识别规则不完整,威胁建模数据量不足,分析准确性和效率不足。随着每年新的互联网软件威胁的快速出现,迫切需要自动构建和更新一个相对完整的规则库,以提高威胁分析的有效性和自动化程度。本文基于STRIDE方法提出了结合类型规则和交互规则的完整的威胁识别规则模型,收集和整理了Web安全领域全面的规则库数据,构建了高质量的规则库。随后,本文提出了一种对STRIDE威胁进行分类的自动化方法(TextCPR)。该方法结合了TextCNN文本分类模型和产生式规则。首先,对数据进行预处理;然后,使用TextCNN分类模型确定威胁内容的漏洞基础类别;最后,通过产生式规则的方法获取对应其漏洞基础类别的STRIDE威胁类别。本文进一步提出了一种用于构建规则库的自动化方法(ACUTIRule)。此外,本文为规则库设计了自动更新机制,以确保其有效性。该方法首先基于TF-IDF算法获取核心动词短语组,并根据威胁分类生成的STRIDE类别与元素对照表组装关系三元组表达式;然后,使用文本相似度算法提取组件,将威胁描述文本、威胁类别和组件等进行组合,实现类型规则匹配生成;然后,根据组件关系表提取生成交互规则;最后,将两者整合为完整威胁识别规则库,并基于从开源威胁数据平台定时爬取的威胁以及规则自动构建方法实现规则库的自动更新。本文通过对比实验对所提出的方法进行评估,结果表明所提出的TextCPR方法在CNNVD数据集上的精度达到92.5%,召回率达到87.6%,$F1$-score达到89.3%。与基线方法相比,TextCPR方法显著提高了精度、召回率和$F1$-score,且分别提高了11.2%、8.2%和9.2%。为了验证ACUTIRule方法的有效性,本文对采用的基础类型规则库进行拓展作为测试规则集,以验证准确率。然后,通过定量指标将提出的ACUTIRule方法与人工构建方法进行对比。实验结果表明,自动构建规则的准确率达到89.5%。在将相同条目的威胁构建为可使用的规则方面,ACUTIRule方法比手动方法花费的时间要少得多,并且不需要额外的人力成本。与人工构建方法相比,该方法提高了规则构建的自动化程度和效率。

关键词 威胁建模;STRIDE方法;威胁识别规则;威胁自动分类;规则模型自动构建和更新

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2026.00132

An Automated Approach to Constructing STRIDE Threat Rule Model and Updating Rule Base

FU Chang-Lan[1),2)] ZHANG He[1),2)] GUAN Xing-Zheng[1)] LI Feng-Long[3)]

[1)](Software Institute, Nanjing University, Nanjing 210023)

收稿日期:2025-01-28;在线发布日期:2025-08-08。本课题得到江苏省自然科学基金(BK20241195)、江苏省重点研发计划(BE2021002-2)、国家自然科学基金青年基金(62202219,62302210)、CCF-华为胡杨林基金-软件工程专项(CCF-HuaweiSE2021003)、南京大学计算机软件新技术全国重点实验室创新项目(ZZKT2025A12, ZZKT2025B18, ZZKT2025B20, ZZKT2025B22)、海外开放课题(KFKT2025A17, KFKT2025A19, KFKT2025A20, KFKT2024A02, KFKT2024A13, KFKT2024A14, KFKT2023A09, KFKT2023A10)资助。**付昌兰(通信作者)**,博士研究生,主要研究领域为软件安全、威胁建模、软件工程方法与理论。E-mail:changlanfu@smail.nju.edu.cn。**张 贺(通信作者)**,博士,教授,博士生导师,中国计算机学会(CCF)杰出会员,主要研究领域为软件研发效能、开放运维一体化、软件安全、软件过程、软件架构、人工智能系统工程、经验软件工程等领域的科研与实践。E-mail:hezhang@nju.edu.cn。**管兴政**,硕士,主要研究领域为软件安全、威胁建模。**李凤龙**,硕士,主要研究领域为网络安全、云安全、软件工程、威胁建模。

[2)](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023)
[3)](Huawei Cloud Computing Technologies Co., Ltd., Hangzhou 310053)

Abstract Threat modeling is a structured method for identifying and responding to threats, and the STRIDE method has become the de facto mainstream threat identification technology in practice. At present, the analysis of STRIDE threats and the construction of the rules for threat identification largely rely on human expertise, resulting in incomplete rules for threat identification and data volume of threat modeling as well as insufficient analysis accuracy and efficiency. Along with the rapid emergence of new Internet software threats every year, there is an urgent need to automatically construct and update a relatively complete rule base to leverage the effectiveness and automation of threat analysis. This paper proposes a complete threat identification rule model based on the STRIDE method, which combines type rules and interaction rules. Comprehensive rule base data in the domain of Web security is collected and sorted out, and a high-quality rule base is constructed. Then, this paper proposes an automated approach (TextCPR) for classifying STRIDE threats. The approach combines the TextCNN text classification model and production rules. First, the data is preprocessed; then, the vulnerability basic category of the threat content is determined by using the TextCNN classification model; finally, the STRIDE threat category corresponding to its vulnerability basic category is obtained by the method of production rules. This paper further proposes an automated approach (ACUTIRule) for constructing the rule base. In addition, this paper designs an automatic update mechanism for the rule base to ensure its effectiveness. The approach first obtains the core verb phrase group based on the TF-IDF algorithm, and assembles the triplet expression according to the STRIDE category generated by threat classification and the element comparison table; then text similarity algorithm is used to extract components, and threat description text, threat categories and components are combined to match and generate type rules; then the interaction rules are extracted and generated according to the component relation table; finally, the two are integrated into a complete threat identification rulebase, and the rulebase is automatically updated based on the threats periodically crawled from the open source threat data platform and the automatic rule construction approach. This paper evaluates the proposed approach by conducting comparative experiments, and the results show that the precision of the proposed TextCPR approach on the CNNVD dataset reached 92.5%, the recall at 87.6%, and the $F1$-score at 89.3%. Compared with the baseline method, the TextCPR approach significantly improve sprecision, recall, and $F1$-score by 11.2%, 8.2%, and 9.2% respectively. In order to validate the effectiveness of the ACUTIRule approach, this paper expands the basic type rule base used as a test rule set to validate accuracy. Then, the proposed ACUTIRule approach is compared with the manual construction approach through quantitative indicators. The experimental results show that the accuracy of the automatically constructed rules reached 89.5%. The ACUTIRule approach takes much less time than the manual approach and requires no additional labor costs in terms of constructing the same entry threats into usable rules. Compared with manual construction approach, this approach improves the automation level and efficiency of rule construction.

Keywords threat modeling; STRIDE method; threat identification rules; threat automatic classification; automatic constructing and updating of rule models

1 引 言

近年来,随着互联网软件攻击的不断演变,威胁的数量和类型不断增加,软件系统安全态势呈现显著恶化趋势,软件安全已成为一个新兴的全球性挑战。SonicWall 的 2024 年网络威胁报告[1]揭示了网络安全环境令人担忧的局面,即网络攻击的数量、攻击频率以及恶意软件变体显著增加。报告表明 2024 年前五个月恶意软件的数量激增了 30%,平均每天会发现 536 种新的恶意软件变种。这些攻击给人们带来了严重的安全威胁并造成了巨大的经济损失。在 2017 年发生了历史上最严重的一次网络攻击,数百万消费者和数千家企业受到 WannaCry、Equifax 和 Uber 等各种攻击,造成大量的数据泄露[2]。据 Symantec 公司发布的 2021 年威胁态势①报告,勒索攻击仍然是各类组织面临的最严峻的网络安全威胁之一。攻击者正呈现定向化趋势,将目标瞄准大型企业和关键基础设施,这类目标不仅能造成更广泛的社会经济影响,还能迫使受害者支付更高额赎金。作为高风险威胁的典型代表,勒索攻击具有显著的不可逆破坏性。正如报告所述,若在潜在威胁演变为实际攻击后才采取补救措施,将面临双重困境:一方面,攻击已直接造成重大损失,包括关键业务中断导致的服务停摆、核心数据资产泄露、高额勒索赎金支付,以及难以量化的品牌商誉损害;另一方面,事后补救不仅需要投入超常规的人力物力资源,还需耗费漫长恢复周期,往往维护代价巨大。

如若能在软件系统设计初期就尽可能发现潜在的威胁,对于帮助系统架构师设计与开发具有现实意义的安全需求有着重要作用[3]。对发现的潜在威胁进行充分且全面的分析,这不仅有利于针对性地提前制定消减措施,以消除或避免潜在的威胁和被实际攻击的可能性,降低后期面临攻击造成的损失和维护成本,同时可以形成一份易于理解的威胁列表文档,用于后续指导软件开发和测试人员进行实际的安全开发和测试工作。

Howard 等人[4]相信开发人员构建安全软件的唯一途径是理解威胁。而威胁建模(Threat Modeling)[5]是一种结构化方法,可以用于在软件开发生命周期的早期阶段识别、分析并应对系统中潜在的威胁和漏洞,利用抽象的方法来帮助思考和理解风险。威胁建模强调在安全威胁被对手利用之前主动识别威胁,定义防范或减轻系统威胁的对策,并实施消减措施以缓解安全威胁,使组织能够在开发生命周期的早期解决漏洞。可靠性、完整性、不可否认性、机密性、可用性及授权六种安全属性是保护系统数据安全的重要属性。威胁建模中的 STRIDE 方法[5-6]从安全属性角度分析威胁带来的危害,根据威胁影响的安全属性进行分类,用于在威胁识别时帮助发现系统威胁。STRIDE 威胁建模方法最初是由微软公司的安全研究人员开发的[6],旨在帮助开发人员和安全专业人士在设计和开发应用程序时更好地理解系统中的威胁。STRIDE 方法不仅可以帮助开发者尽早发现威胁,而且通过对威胁造成影响的安全属性进行分析,还可以制定相应的消减措施。因为这些优势,STRIDE 方法已成为目前应用最为广泛的威胁建模方法[7]。

目前,威胁建模仍存在一些局限,如自动化程度不高,主要为安全专家人工分析,专业知识要求高,运用于实践中有效地解决安全问题的门槛较高。由于威胁识别的目的是分析系统中所有的潜在威胁,所以首先会将系统分解为其逻辑或结构组件,再逐步分析每个组件可能出现的威胁。使用基于 STRIDE 方法的威胁建模工具需要建立一套威胁对应组件的威胁识别规则库,并基于规则库建立威胁识别规则模型,以评估潜在威胁的可能性和影响程度。这通常需要专业的知识和经验,对非专业人士来说往往比较困难。目前在构建威胁识别规则模型时需要手动对威胁进行 STRIDE 类别分类,然后再手动建立威胁识别规则库。这种方法存在效率低的问题,并且难以保持一致的标准。

威胁建模目前在研究、工具支持和实践方面都还不够成熟[8]。学术上的研究成果尚未应用于实践,未能满足业界的实际需求[8-9]。现有的威胁建模工具高度依赖于人工构建威胁识别规则。为了实现威胁建模的自动化以提高威胁建模的质量和效率,需要有数据支持,需要自动分类威胁、自动构建用于识别威胁的规则库。目前缺乏全面的威胁识别规则模型和规则库数据,也缺乏自动构建规则库的方法。这些问题给开发人员带来了繁重的工作负担和较高的技术门槛,导致威胁识别的准确性和效率不足。随着每年新的互联网软件威胁的快速出现,实时更新规则库的需求使得问题的解决更加急迫。因此,迫切需要一种自动识别 STRIDE 威胁的方法,以及一种支持自动构建和更新全面威胁识别规则库的方

① https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/threat-landscape-2021。

法。由于目前缺乏带有 STRIDE 类型标记的威胁数据,且 STRIDE 类别与威胁描述的语义信息之间缺乏直接联系,因此直接从威胁描述中自动识别和分类 STRIDE 威胁是一项挑战。此外,规则模型的自动构建往往需要通过规则挖掘来构建规则[10],而威胁建模领域的威胁识别规则模型往往由不同的安全专家根据自己的认知建立,难以挖掘出通用规则。目前缺乏全面的威胁识别知识库数据和相关技术。因此,如何自动构建威胁识别规则模型是一个挑战。

针对上述问题,本文基于 STRIDE 方法提出了完整的威胁识别规则模型,提供了全面的规则库数据。本文提出了一种自动化构建规则库的方法,首先,为了解决 STRIDE 标签数据不足的问题,我们从中国信息安全国家漏洞数据库(China National Vulnerability Database of Information Security, CNNVD)①中获取数据,结合 TextCNN 文本分类模型和产生式规则对 STRIDE 威胁进行自动化识别和分类;然后,采用文本相似度算法实现类型规则匹配,并基于类型规则和组件关系表生成交互规则;最后,将这两种类型的规则集成为一个完整的威胁识别规则库。此外,本文还设计了规则库的自动更新机制,以保证规则库的时效性。

本文的主要贡献如下:

(1) 基于 STRIDE 方法提出了结合类型规则和交互规则的完整的威胁识别规则模型,收集和整理了 Web 安全领域全面的规则库数据,构建了高质量的规则库;

(2) 提出了一种自动构建 STRIDE 威胁识别规则模型并更新规则库的方法(ACUTIRule),该方法首先基于 TF-IDF 算法获取核心动词短语组,并根据威胁分类生成的 STRIDE 类别与元素对照表组装关系三元组表达式,然后利用文本相似度算法提取组件,将威胁描述文本、威胁类别和组件等进行组合实现类型规则匹配生成,再根据组件关系表提取生成交互规则,最后将两者整合为完整威胁识别规则库,并基于从开源威胁数据平台定时爬取的威胁以及规则自动构建方法实现规则库的自动更新;

(3) 提出了一种针对 STRIDE 威胁的自动分类方法(TextCPR),该方法结合了 TextCNN 文本分类模型和产生式规则,首先将数据进行预处理,然后利用 TextCNN 分类模型确定威胁内容的漏洞基础类别,最后通过产生式规则的方法获取对应其漏洞基础类别的 STRIDE 威胁类别。

我们通过对比实验对所提出的 STRIDE 威胁自动分类方法和威胁识别规则库自动构建方法进行了评估,结果表明 STRIDE 威胁自动分类在 CNNVD 数据集上的精确度达到 92.5%,召回率达到 87.6%,F1-score 达到 89.3%。与基线方法相比,我们的分类方法显著提高了精确度、召回率和 F1-score,分别提高了 11.2%、8.2% 和 9.2%。我们的规则构建方法自动构建的规则准确率达到 89.5%。与人工构建相比,我们的方法提高了规则构建的自动化水平和效率,误差在 10% 以内。本文提出的方法可以支持自动威胁识别和安全风险分析等安全实践的左移。

本文第 1 节介绍引言;第 2 节介绍背景及相关工作;第 3 节介绍威胁识别规则库的制定;第 4 节详细介绍自动构建和更新威胁识别规则库的方法;第 5 节对本文提出的方法进行实验评估;第 6 节展示一个实际应用示例;第 7 节讨论本文工作的重要性、相关考虑和局限性;第 8 节总结全文并提出下一步工作。

2　背景及相关工作

本节介绍研究背景及相关工作。

2.1　背　景

(1)威胁建模背景

威胁建模[5]是使用抽象的概念来思考、分析和预测系统可能存在的风险。威胁建模是一套方法论,也是一套过程,即通过识别威胁和漏洞,定义防范或减轻系统威胁的对策,从而优化系统安全的过程。威胁建模过程包括预设场景、图形化建模、识别威胁、处理威胁、验证等步骤:首先是对系统预设场景,定义系统的安全需求;其次是对系统进行建模,图形化有助于理解系统以及定位威胁的攻击面;然后需要借助特定的模型和方法来识别威胁;在处理威胁阶段通过实施相应的措施缓解和处理威胁,可以从设计和开发中提高软件的安全性;最后,验证阶段需要测试是否已经对相关威胁进行有效处理,确保威胁被缓解的同时,系统的安全性得到有效提高。

在系统未构建之前,具体来说可以是在软件开发生命周期的早期如需求分析和系统设计阶段,通过威胁建模分析将要构建的软件可能存在的安全漏洞,识别威胁,定义防范或减轻系统威胁的对策,将威胁分析结果集成到其他阶段,尽早处理潜在的安全漏洞,防止攻击者通过该漏洞控制或破坏系统。

① https://www.cnnvd.org.cn/home/dataDownLoad。

通过威胁模型对系统进行结构化分析，可以发现需求和设计中潜在的安全缺陷，另外还可以用于对系统进行场景识别，并对场景对应的安全需求进行识别。设计阶段开展威胁建模工作可以依照威胁建模结果评价现有安全需求是否全面、现有安全设计在细粒度和有效性方面是否与安全目标相符。

表 1 STRIDE 类型与 DFD 元素之间的对应关系

	仿冒	篡改	抵赖	信息泄露	拒绝服务	权限提升
外部实体(EE)	√		√			
处理过程(P)	√	√	√	√	√	√
数据存储(DS)		√		√	√	
数据流(DF)		√		√	√	

表 2 业界威胁建模工具

工具		微软 TMT	Threat Dragon	IriusRisk	ThreatModeler
绘图方法		DFD（数据流图）	DFD（数据流图）	DFD（数据流图）	PFD（过程流图）
威胁识别	自动识别	支持	不支持	支持	支持
	威胁识别规则	规则基于组件间与数据流的交互	规则基于组件类别	规则基于组件类别	规则基于组件类别
	威胁库支持	有威胁库支持	无威胁库支持	有威胁库支持	有威胁库支持
自定义模板功能		支持	不支持	支持	支持
威胁报告生成		支持	支持	支持	支持

可靠性、完整性、不可否认性、机密性、可用性及授权六种安全属性是保护系统数据安全的重要属性，从这六种安全属性的角度对威胁进行分析，有助于开发人员充分了解攻击者可能使用的不同威胁类型。因此，STRIDE 威胁建模方法[5-6]从安全属性角度分析威胁带来的危害，根据威胁影响的安全属性进行分类，用于在威胁识别时帮助发现系统威胁。STRIDE 威胁建模方法最初是由微软公司的安全研究人员于 1999 年创建[6]，旨在帮助开发人员和安全专业人士在设计和开发应用程序时更好地理解系统中的威胁。STRIDE 方法不仅可以帮助开发者尽早发现威胁，而且通过对威胁造成影响的安全属性进行分析，还可以制定相应的消减措施。STRIDE 方法已成为目前应用最为广泛的威胁建模方法[5,7,9]。

STRIDE 方法根据六种安全属性，将威胁分为仿冒(Spoofing)、篡改(Tampering)、抵赖(Repudiation)、信息泄露(Information disclosure)、拒绝服务(Denial of service)和权限提升(Elevation of privilege)六类。该方法基于数据流图(Data Flow Diagram，DFD)建模系统场景，并在 DFD 元素即外部实体(External Entity，EE)、过程(Process，P)、数据存储(Data Store，DS)、数据流(Data Flow，DF)与六种 STRIDE 威胁之间设置了映射关系[5]，如表 1 所示。STRIDE 可用于通过分析场景 DFD 中的元素类型，根据对应的映射关系，可以帮助软件开发人员识别软件可能会面临的潜在的威胁类型[5]。

威胁建模作为设计和开发安全软件的重要实践[5]，正被越来越多的团队整合到项目的开发流程中，以帮助开发更安全的软件应用。为进一步落实可信计算和构建安全的软件，微软将威胁建模融入软件开发过程中，并形成了软件开发生命周期(Software Development Lifecycle，SDL①)[11]。同时，结合 STRIDE 模型，构建了微软威胁建模工具(Threat Modeling Tool，TMT②)[12]，以让开发人员可以更便捷地进行威胁建模。在微软提出的软件开发安全流程 SDL 中，基于 STRIDE 的威胁建模作为其核心要素，主要包含四个主要步骤，分别是绘制数据流图(DFD)、识别威胁、制定消减措施和验证。其中，识别威胁是核心步骤，决定了威胁分析的全面性和准确性。该步骤产出的威胁列表不仅作为后续步骤的输入，还将帮助开发人员理解和分析威胁。该步骤当前主要依靠安全专家人工分析，需要基于对数据流图的观察与理解，结合专业领域知识经验，对数据流图涉及的各组件进行逐个剖析，才能整理得出威胁列表。

目前，可以代表业界水平的威胁建模工具包括微软 TMT、Threat Dragon③、IriusRisk④、ThreatModeler⑤。这四款可以代表业界水平的工具均是基于规则识别威胁，但是规则不全面。微软工具的规则基于组件间与数据流的交互及其对应的 STRIDE 威胁类别，其余工具的规则基于组件对应的威胁类别。这两种规则的制定都以确定威胁类别为基础。Threat Dragon 工具没有提供数据支持，需要人工输入，其余三款工具有数据支持但是数据量特别少且描述简略难以理解。如最知名的微软 TMT 工具威胁的粒度粗而量少，且其中威胁的标

① https://www.microsoft.com/en-us/securityengineering/sdl/。
② https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling。
③ https://github.com/OWASP/threat-dragon。
④ https://www.iriusrisk.com/threat-modeling-platform。
⑤ https://threatmodeler.com/。

题与描述相对简略,部分描述与标题相同,整体质量有待提升。威胁建模工具在业界的现状如表 2 所示。虽然这些工具能够根据规则生成一些威胁,但是其规则不够全面,主要为人工构建的规则,技术门槛较高,导致威胁识别准确度与自动化程度不足。安全专家需要经过长时间的分析,手动将威胁分类为 STRIDE 类别,构建威胁识别规则,这需要较高的专业知识和相当大的工作量。此外,由于新威胁的不断出现,威胁库的更新机制问题也尚待解决。因此,急需完善的威胁识别规则库,可以实现规则库的自动构建和更新,以便支持威胁建模安全实践。

(2)应用场景

图 1 为威胁识别规则的使用场景图。用户根据系统场景绘制场景 DFD,基于预制的威胁识别规则识别场景中可能的威胁。本文所提出的构建 STRIDE 威胁规则模型和更新规则库的方法用于支持威胁识别规则的构建和更新,支持识别场景中的威胁。上游设计阶段通过威胁建模识别的结果可以用于指导下游测试阶段生成安全测试设计等[13]。从识别的威胁进一步生成安全测试,通过生成的测试检测并缓解威胁,将安全问题更好地更全面地嵌入到软件安全设计和实现之中,提高软件的安全性[13]。上游阶段的威胁建模工作,即基于本文构建的威胁规则分析和识别系统威胁,与下游的测试生成工作[13]之间存在前后对接关系,两者相互作用,构成一个完整的整体,在软件安全实践中起着重要作用。

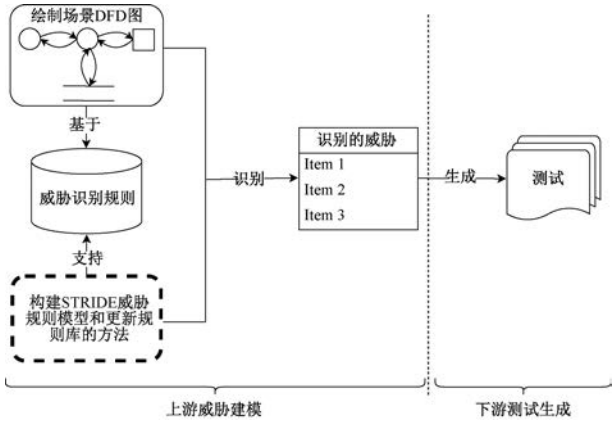

图 1　威胁识别规则使用场景图

2.2　相关工作

本节概述了威胁建模和规则构建方面的最新研究。

(1)威胁建模

目前,威胁建模的相关研究主要包括系统文献综述(SLR)[7,9,14-15]、威胁识别[16-18]、检测[19-22]和分析[23-26]、威胁建模语言[27-32]、基于威胁建模的安全测试[33-34]等。对威胁建模的文献综述[7,9,14]表明,大多数威胁建模工作仍需人工完成,其验证的保证有限,威胁建模过程的自动化水平需要提高。在威胁建模语言方面,Xiong 等人[29]提出了一种名为 enterpriseLang 的领域特定语言(DSL),该语言可以对其系统模型实例进行攻击模拟,以支持企业系统中威胁行为的分析和安全评估。在威胁识别方面,Zou 等人[17]提出了一种用于企业安全高级持续威胁(Advanced Persistent Threats,APT)策略自动识别的框架。Hanvey 等人[16]提出了一种形式化的方法来识别社交网站中基于传递性的隐私泄露。Assen 等人[35]提出了一种资产驱动的威胁建模方法,为建模与人工智能(AI)相关的资产、威胁、对策以及量化剩余风险提供了指导和自动化,用于识别基于大语言模型(LLM)的应用程序中的风险。在威胁检测和分析方面,Rouland 等人[21]介绍了一种通过安全需求来检测和处理威胁的集成方法。Alidoost 等人[20]提出了一种利用基于属性的攻击图检测新一代威胁的方法。Wilhjelm 等人[26]提出了一种用于基于机器学习的系统中安全需求提取的威胁分析方法。基于威胁建模的安全测试方面,Mahmood 等人[33]提出了一种将威胁建模与安全测试相结合的系统安全测试方法,使用 STRIDE 方法枚举威胁,并基于攻击树生成测试用例。此外,一些研究将威胁建模应用于信息物理融合系统(CPS)[36-38]、工业控制系统(ICS)[39,14]、物联网(IoT)[40-42]、云服务[43]和 5G 移动通信[44-47]等领域,这些威胁建模应用方面的研究主要采用的都是 STRIDE 方法。

威胁识别是威胁建模的核心步骤,现有的威胁识别方法主要有 STRIDE[5-7,9]、攻击树[5,48-49]、攻击图[29]、误用案例[50]和基于时序图的技术[51]。攻击树[5]采用树形结构分析攻击者的攻击步骤和攻击手段[48]。攻击图[29]用于模拟企业系统受到的攻击,识别系统威胁行为,评估系统安全风险。误用案例[50]类似于常规用例,但重点在于描述攻击者的行为,并侧重于分析用户与系统交互时的威胁。基于时序图的威胁建模技术[51]主要侧重于对进程和操作进行表征,以及对业务流程中的威胁进行识别。其他识别威胁的方法包括 LINDDUN①[52]、PASTA②[53-54]、Trike[55]、OCTAVE[37,56]、NIST[53]等。其中,STRIDE 方法[5-6]自 1999 年提出以来,不仅在微软内部实践了很长时间,而且被业界的学者

① https://www.linddun.org/。
② http://onlinelibrary.wiley.com/book/10.1002/9781118988374。

和组织广泛使用，是目前最主流和公认的方法[7,9]。STRIDE 方法基于 DFD 对系统进行建模。DFD 具有普适性和易用性，并且易于与软件开发的后期阶段进行交互。DFD 是威胁建模中最适当的思考方式[5]。安全问题经常出现在数据流中，数据流模型通常是威胁建模中最理想的模型[5]。DFD 基本上可以涵盖其他方法所关注和侧重的方面。基于 DFD 可以对系统威胁进行全面的表征和分析，以满足业务需求。

目前，现有的威胁识别方法主要依靠人工分析[7]，面临着处理时间长、技术门槛高的问题。为了解决此问题，一些学者初步探索了威胁建模的自动威胁识别。通过将安全专家的知识以识别模型的形式进行保存，以达到重用模型进行威胁识别和缩减工作量的目的。微软 TMT 工具采用基于规则模型的方式支持了自动识别功能，并提供了三个基础模板进行参考，但其规则模型不够全面而且其构建依赖人工分析。后续有许多研究者在此基础上，依赖于微软预留的模板功能，通过对特定领域的研究和规则制定，完成了智能电网[57]、边缘计算系统[58]、工业控制系统(ICS)[59]、电力配电系统[60]等领域的自动识别模型。Rouland 等人[21]和 Casola[61]等人以类似于微软 TMT 的方式自动识别威胁，将威胁绑定到组件。Rouland 等人[21]通过定义抽象级组件、端口和连接器元模型，以及定义组件和连接器之间的行为和属性，将六类 STRIDE 威胁绑定到特定组件的行为值。该方法可以支持识别组件对应的 STRIDE 威胁类型，但不能提供详细的威胁列表。Casola 等人[61]以类似微软 TMT 工具的自动识别模型的构建方式，将组件与威胁的关联关系预置成为规则模型进行识别，然而其构建的威胁库仅包含大约百例威胁，且其威胁仅包含短语式的标题与简要描述，不易于理解。Välja 等人[62]尝试基于本体论实现威胁建模自动识别，该方法通过人工输入的方式将组件属性相关知识输入本体论模型之中，从而得到相关威胁，对于输入的知识质量要求较高。目前，关于威胁建模自动识别依然还处于探索阶段，人工进行威胁分析依旧是主流。本研究基于主流的 STRIDE 方法提出了威胁识别规则库以及其自动构建和更新的方法，在一定程度上提高了威胁识别的自动化程度。

(2) 自动威胁分类

为了解决难以对漏洞和威胁进行有效分类的问题，一些研究人员尝试应用机器学习(ML)方法对漏洞进行自动化分类，这可能在一定程度上有助于提高分类的准确性，避免人工分类错误。Li 等人[63]提出了一种 SOM 聚类方法对漏洞数据进行无监督分类。Chen 等人[64]提出了一种基于支持向量机(Support Vector Machine，SVM)的漏洞自动分类模型，可以对漏洞数据进行分类和预测。Ayoade 等人[65]提出了一种名为 MITRE 的框架，通过构造 ML 分类器来自动提取和分类来自不同组织的威胁报告。与其他现有方法相比，其分类准确率提高到 84%。Islam 等人[66]提出了一种名为 SmartValidator 的框架，通过构建自动预测模型来自动识别和分类网络威胁数据。他们使用多个算法模型来评估构建模型的性能，并选择性能最好的模型来重建预测模型。构建自动预测模型对网络威胁数据进行分类时，75%的模型 $F1$-score 在 0.8 以上。极端梯度提升(eXtreme Gradient Boost，XGB)分类算法在威胁类型分类中准确率最高，达到 87.6%。Liao 等人[67]提出了一种将隐含 Dirichlet 分布主题模型与 SVM 相结合的方法，在主题向量空间构建一个自动漏洞分类器。Qu 等人[68]提出了一种基于卷积神经网络(Convolutional Neural Network，CNN)的漏洞自动分类方法，分类准确率达到 79%。

虽然已经有一些威胁描述自动分类的研究，但这些研究都是根据网络攻击的行为进行分类，目前还没有将威胁描述自动分类为 STRIDE 类别的研究。本文将威胁分类与威胁建模相结合，提出了 STRIDE 威胁的自动分类方法。这是在特定技术组合(ML 结合映射规则)上的首次尝试，以支持威胁识别规则的自动构建。

(3)威胁识别规则模型的自动构建

目前，业界现有的威胁建模工具大多依赖于“if-then”类型规则引擎，规则库的内容由专家手工制定和填充。随着机器学习的兴起，一些工作开始使用基于机器学习的方法来解决规则挖掘和知识库自动构建问题。Kadhim 等人[69]提出了一种多智能体系统，从其资源中获取知识，用于诊断领域基于规则的专家系统知识库的自动构建。系统使用 TMIA、EMIA 和 MIAKDD 三个代理从文本中获取规则实体之间的关系，从资源中提取知识，构建知识库。TMIA 用于文本规则挖掘，EMIA 用于专家规则处理，MIAKDD 用于知识库构建。TMIA 文本挖掘智能代理模块主要采用 ML 和 NLP 技术相结合的方式，将文本内容转换为结构化规则。它通过基于传统术语加权方案(TF-IDF)的文本文档分类来执行，

并使用 Stanford 解析器对该文档中的每个句子进行分析并生成解析树。然后,TMIA 查找所有因果词,并将其作为分离词,在概念数据库的基础上生成模式和子模式,最终实现在获取文本后自动获取规则实体之间关系的目的。Belabed 等人[70]提出了一种基于协同多智能体的数据挖掘系统,该系统可以从真实数据集中提取一组规则来构建知识库。他们专注于关联规则的挖掘。采用 K-Means 算法对变量数据进行聚类,采用先验算法和遗传算法两种关联规则算法挖掘关联规则。他们从大型数据集中提取知识,并利用多智能体系统进行知识建模,实现了在多智能体环境下利用定量数据集获取特定关联规则的目标。

目前,在威胁建模领域还没有关于规则模型自动构建的研究。在对 STRIDE 威胁自动分类的基础上,本文进一步提出了一种自动构建和更新威胁识别规则库的方法。由于规则模型的自动构建往往需要规则挖掘来建立和构建规则[10],而威胁建模领域的威胁识别规则模型往往是由安全专家根据自己的认知建立的,难以挖掘出通用规则。因此,如何自动构建威胁识别规则模型是一个挑战。在人工辅助数据准备的基础上,本研究首次实现自动化方法的探索性尝试,并提供常见数据集。

3　基础威胁识别规则库的制定

本节主要介绍基础威胁识别规则库的制定过程,如图 2 所示,包括:(1)收集和处理基础威胁和组件数据;(2)制定威胁识别规则模型。制定规则模型的过程包括制定类型规则和制定交互规则。

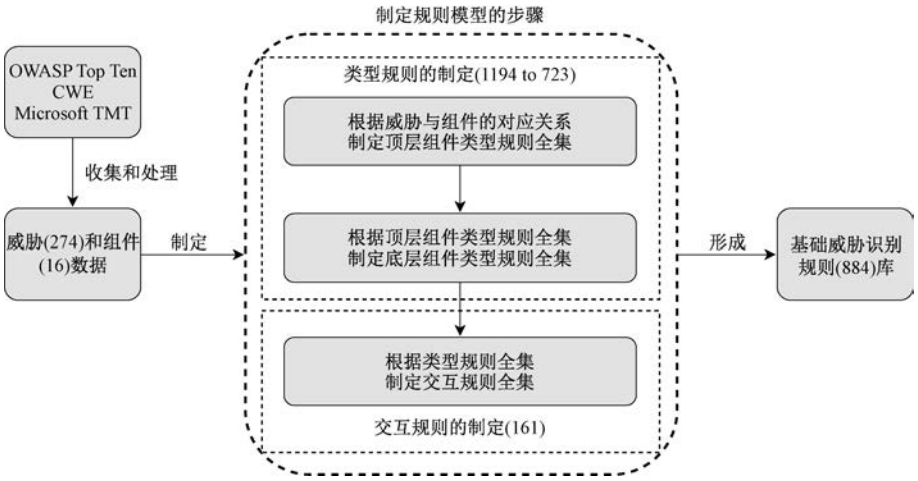

图 2　基础威胁识别规则库的制定过程

3.1　基础数据库构建

本文主要构建了 Web 安全领域的规则模型。考虑到与 Web 领域的相关性、数据的数量和全面性,我们使用了 OWASP Top Ten① 的 2021 版本、Common Weakness Enumeration (CWE)②中的软件开发视图(Software Development View, SDV)和微软 TMT 工具中的 Azure 云服务(Azure Cloud Services)模板作为主要威胁和组件数据源。

对于 SDV 数据,我们对该视图内包含的弱点进行分析和父子关系提取,整个弱点列表呈现 5 层(第 0～4 层)树状结构,整体抽象程度随层数的增加而降低,过于抽象与过于具体的弱点均不适合作为威胁条目,因此确定选取中间层节点作为待选条目进行威胁收集。威胁的收集工作将主要聚焦于第 2 层节点,以及部分的第 1 和第 3 层节点。将其中抽象程度适中、适合作为设计阶段分析的威胁进行汇总:一方面,不过于抽象,易于理解和后续指导开发;另一方面,不拘泥于代码、框架、语言层次的细节,避免导致细节爆炸。收集到的数据不包含 STRIDE 类型,对此进行 STRIDE 类型标注。由于同一个弱点可能产生多种后果,因此,一个原始威胁条目可能成为多个 STRIDE 威胁。STRIDE 类型标注后,最终汇总得出来自 SDV 的 STRIDE 威胁列表,其数量为 426 条。

微软 TMT 中的威胁自带 STRIDE 类型,但并不是以单独的威胁列表形式存在,而是以嵌入在规则中的形式存在。由于一个 STRIDE 威胁可以关联于多个组件,从而产生多条威胁规则,因此多个威胁规则其实对应于同一个威胁,所以需要对这种重

① https://owasp.org/www-project-top-ten/。
② https://cwe.mitre.org/index.html。

复进行去重,去重后将获得来自微软 TMT 的威胁列表。将此种去重后的威胁列表再与来自 CWE 中 SDV 的威胁列表进行合并去重,共得到 171 条威胁。从微软 TMT 中还收集了 DFD 组件数据。其原始包含两个层级共 39 个组件,存在三类情况:对于常见组件可以直接进行保留;而部分过于细粒度划分的组件,对其进行合并处理;至于微软自己旗下的软件组件,则在该过程中进行去除。处理后最终得到两个层级共 16 个组件。

OWASP Top Ten 作为 Web 领域的权威数据库,是本文的核心数据源之一。在过去的近二十年间,发布了六个版本的榜单。对六个版本迭代关系进行提取,版本之间存在映射和包含关系,2021 版本涵盖范围最为全面。由于该数据源抽象程度较高,需要结合与之对应的 CWE 列表进行收集。首先通过对 2021 版对应的 CWE 列表进行汇总,将所有 CWE 条目的 CWE ID、标题、描述进行收集和提取,以便后续进行数据的合并与处理。在收集的过程中对过于抽象的条目进行寻子搜索,找寻更为具体,更具有指导性的子条目作为替代;对于过于具体的条目,如关注于具体框架,具体语言的条目,进行寻父分析,找寻可以脱离具象的技术,表达指导性的威胁的父条目作为替代。收集完毕后,对重复或相似条目进行去重与合并之后确定了在 OWASP Top Ten 威胁下的 167 个 CWE 弱点条目。与 CWE 类似也需要进行 STRIDE 类型标注。一个传统威胁根据其后果影响,可能对应着多个 STRIDE 威胁,故而转换后的 STRIDE 威胁数目达到了 242 个。

本文将收集到的威胁数据的描述转换为统一的格式描述,转换格式为:由于{威胁},攻击者可以{攻击手段},导致{STRIDE 类型}问题。此外,还添加了 CWE 链接,便于开发者到原始页面详细了解更多细节信息(包括案例、消减措施等),以增强威胁的可理解性。在进行格式化描述转换后,对三方数据源数据进行整合去重处理,将 OWASP Top Ten 的数据与已先一步完成合并的两方数据逐一比照,将其中相同、相似或包含的条目合并为同一个条目,并对其余条目进行剔除。最终获得去重完毕后的威胁列表,共 274 条威胁。

3.2 威胁识别规则模型制定

在收集和处理的数据的基础上,通过充分的威胁分析,同时参考业内开源规则模型的制定,我们从组件类型关联威胁和组件交互关联威胁两个角度,从威胁与组件的关联关系全集中筛选制定两种规则,合并汇总得到一个覆盖面更广、可阅读性更强的规则模型。威胁识别规则模型制定过程如图 2 所示。规则制定从两个角度出发,第一个角度是组件的类型和威胁之间的关联,不同类型的组件可能面临不同类别的 STRIDE 威胁;第二个角度是组件的交互和威胁之间的关联,部分威胁将依赖于特定组件之间产生数据流而出现,该角度规则制定时需要基于第一个角度的结果。从这两个角度制定的规则分别是类型规则和交互规则。

类型规则的制定分为两个步骤,由上至下进行。第一步是顶层组件(即外部实体(EE)、处理过程(P)和数据存储(DS))类型规则的制定。类型规则的处理逻辑主要基于 STRIDE 方法定义的组件与威胁之间的关联关系,如表 1 所示。通过表 1 中的对应关系,我们确定了不同类型组件可能关联的威胁集合。结合前文收集到的数据,依据关联关系将顶层组件集合和 STRIDE 类型集合进行乘积操作,得到待定规则全集。再对其进行遍历筛查和分析,最后得到顶层类型对应的规则全集,共有 293 条规则。第二步是底层组件类型规则的制定。在前面顶层类型规则的基础上,将每个顶层组件集替换为其从属的底层组件,例如将顶层组件的外部实体(EE)替换为底层组件浏览器(Browser),然后执行筛选操作。通过此流程,完成底层组件的类型规则制定,总量为 901 条。总共为顶级组件和底层组件确定了 1194 条类型规则。表 3 显示了规则模型中的类型规则示例。

交互规则聚焦于数据流(DF),通过数据流串起的组件之间的数据流动,即交互,将会产生特定的威胁。换言之,部分威胁的产生,不仅依靠单一组件,需要与特定组件之间产生数据流才可。因此,交互规则的制定将主要依托于,对拥有同样关联威胁的组件之间的数据流进行判断得出。在制定完毕的类型规则基础上,根据顶层组件之间潜在数据流的分析,确定交互规则关系的可能集合,对每种集合进行分析,筛选得到绑定于数据流的交互规则。根据分析,外部实体之间不会直接产生数据流,它们直接与处理过程进行交互;数据存储也与处理过程产生数据流,处理过程是核心;处理过程之间也存在数据流交互。在这种交互关系分析的基础上,将作为发生者即源(Source)组件和被作用者即目标(Target)组件在前一步中制定得到的类型规则,进行合并分析,将其中重复且存在数据流关联的威胁提取出来,形成交互规则。例如 Browser 和 Web Application 关联相同的威胁“由于通过捕获-重放绕过身份验证,攻击者可以绕过身份验

证,导致仿冒问题”。通过对威胁进行分析,捕获-重放是指攻击者对 Browser 和 Web Application 之间的数据流进行嗅探,并在 Web Application 的身份验证存在缺失时重放攻击,以绕过身份验证。可以明确得知,此威胁需要 Browser 和 Web Application 之间产生数据流才可以出现,因此该规则应该从类型关联规则提取为交互规则。该条规则触发仿冒威胁,其数据流是从 Browser 到 Web Application,即 Browser 应为 DF 的 Source,而 DF 的 Target 应为 Web Application。表 4 显示了规则模型中交互规则的示例。交互规则最终数量为 161 条,提取交互规则后类型规则数减少为 723 条。共有 884 条规则构成了基本的威胁识别规则库。

如以上所描述,本文从组件类型关联威胁和组件交互关联威胁两个角度进行规则的制定,得到的所有规则最终汇总为本文构建的规则模型,用于支持自动威胁识别。表 3 和表 4 分别为规则模型中类型规则和交互规则的示例。类型规则包含威胁的信息以及对应的组件,在识别威胁的过程中使用类型规则可以为每个出现的组件匹配到可能出现的威胁。交互规则包含威胁的信息以及发生交互的组件,在识别威胁时如果同时出现交互规则库中包含的发生者(Source)和被作用者(Target)两种组件,且存在数据流关联,则匹配出对应可能出现的威胁。

表 3　类型规则示例

DFD 组件类型	威胁类型	三元表达式	威胁描述内容	组件
GE. EE	S	由于渲染 UI 层或帧的不适当限制,攻击者可以绕过身份验证,导致仿冒问题	Web 应用程序不会限制或错误地限制属于另一个应用程序或领域的框架对象或 UI 层,这可能会导致用户混淆用户正在与哪个界面交互。一个 Web 应用程序应该对是否允许在框架、对象、嵌入或 applet 元素中渲染设置限制。如果没有这些限制,用户可能会在他们不打算与应用程序进行交互的时候被欺骗。	Browser

表 4　交互规则示例

DFD 组件类型	威胁类型	三元表达式	威胁描述内容	发生者(Source 组件)	被作用者(Target 组件)
GE. DF	S	由于通过捕获-重放绕过身份验证,攻击者可以绕过身份验证,导致仿冒问题	当软件的设计使恶意用户有可能嗅探网络流量,并绕过身份验证,将其重放到有问题的服务器,达到与原始消息相同的效果(或稍作修改)时,就会存在捕获-重放缺陷。捕获-重放攻击是常见的,如果不使用密码技术很难被击败。它们是网络注入攻击的一个子集,依赖于观察以前发送的有效命令,然后在必要时略微更改它们,并将相同的命令重新发送到服务器。	Browser	Web Application

4　威胁识别规则模型自动构建方法

本节详细阐述了本文提出的自动构建 STRIDE 威胁识别规则模型和更新规则库(ACUTIRule)的方法,总体流程如图 3 所示。所提出的 ACUTIRule 方法的整个过程包括五个步骤,与图 3 中的编号相对应,包括:(1)数据获取和处理;(2)STRIDE 威胁的自动分类;(3)关系三元组表达式的提取;(4)生成类型规则和交互规则;以及(5)规则库的自动更新。

4.1　数据获取和处理

为了解决 STRIDE 标签数据有限的挑战,我们基于国内外的网络威胁(Cyber threat)数据质量确定 CNNVD 为主要数据收集平台。CNNVD 是基于恶意软件信息共享平台(Malware Information Sharing Platform, MISP)收集数据的综合性、权威性信息平台。目前,CNNVD 已经整合了 220251 条国内外主流软件产品和服务产生的漏洞信息,并且每日持续更新。此外,CNNVD 平台还标注了所收录的每条漏洞信息的漏洞类别。其他常用的数据平台,如 CWE (Common Weakness Enumeration),已经收录了 1137 个威胁,数据量过少,极易导致过拟合的现象。CVE (Common Vulnerabilities and Exposures)收录的数据包含大量产品名称不同但威胁描述完全一致的条目,包含的属性较少、描述粒度过细,没有类别标注。来自 MISP 的数据在内容和数

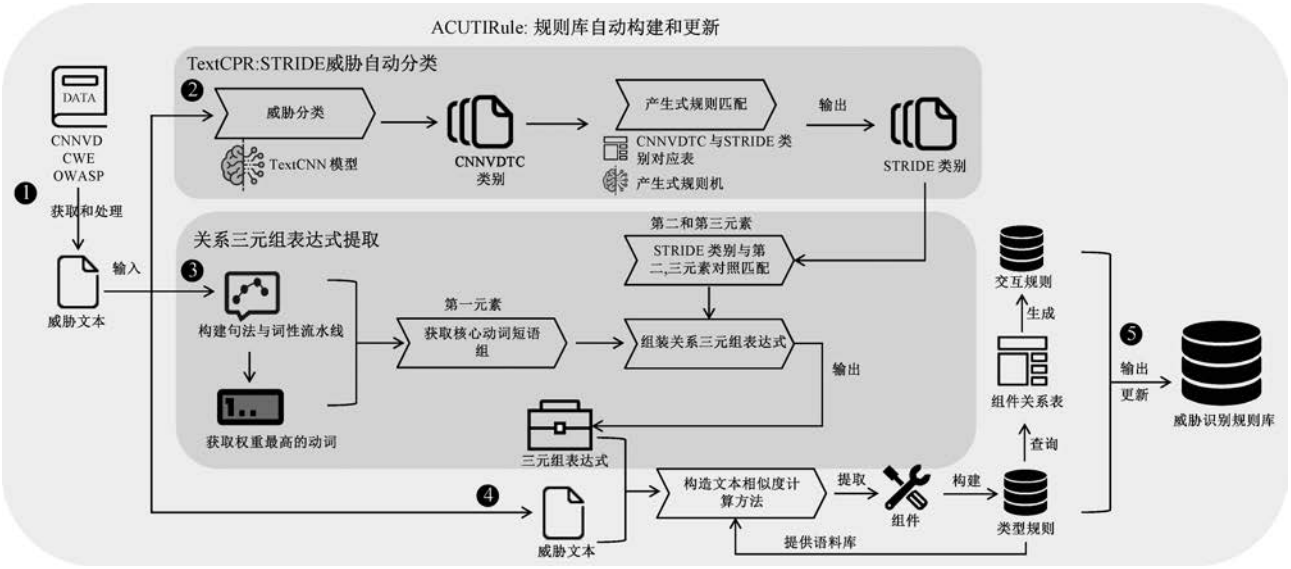

图 3 自动构建和更新规则库的方法

量上与 CNNVD 数据高度相似,但缺少部分国内企业相关的威胁内容,而且缺少具有详细描述的类别标注。因此,我们主要使用 CNNVD 数据进行 STRIDE 威胁自动分类并进一步自动构建规则库。此外,在规则库的更新中,我们还加入了获取到的 CWE 和 OWASP Top Ten 数据,其中,获取到的 CNNVD 数据需要进行数据处理。

CNNVD 收录的漏洞信息有着分类、分级和内容描述等标准规范,其中本文重点关注在分类和内容描述两类标准规范上。CNNVD 将信息安全漏洞划分为 5 个层次 26 种类型(CNNVD Threat Category, CNNVDTC),判别威胁类型时可根据 CNNVDTC 的层次关系进行辅助。CNNVD 的数据内容都是以 XML 文件的形式呈现,但是在实际爬取解析的过程中发现,CNNVD 数据的 XML 文件大多没有按照国际通用的 XML 标准格式,导致无法使用已有的 XML 解析工具获取数据。CNNVD 平台的多数 XML 文件虽然在威胁内容书写方面没有严格遵守通用的 XML 标准格式,但是严格执行了《CNNVD 漏洞兼容性描述规范》规定的 XML 标签格式,所以本文利用正则表达式对于特定内容查询的特性,构造了一种 CNNVD 特定的 XML 解析器来帮助获取数据。所构造的 XML 解析器的核心处理逻辑是:首先将 XML 文件按行读取,然后根据“<entry>”标签将每条漏洞信息进行分离,最后再根据本文所需的漏洞名称、漏洞编号、漏洞类别以及漏洞描述所对应的 XML 标签获取对应的内容。

通过数据处理将原始数据转化为质量高的可用数据。由于从 CNNVD 平台获取到的网络威胁信息包含与威胁无关的软件名称、版本等无用信息,所以首先需要经过数据清洗去除掉重复数据和空值。然后通过基础类别的可用性和包含关系将数据项进行合并删除,将与软件安全和网络安全不相关的漏洞类别进行删除,将父类和子类的漏洞类型描述相似且漏洞行为相似的部分进行合并,便于通过对漏洞进行自动分类实现威胁 STRIDE 类别的自动分类。经过处理最终确定了 11 个 CNNVDTC(CNNVD Threat Category)类别,包括信息泄露、跨站脚本、注入、跨站请求伪造、路径遍历、授权问题、后置链接、资源管理错误、加密问题、代码问题、权限许可和访问控制问题。本文进一步利用正则表达式去除威胁描述中的无用信息,即漏洞的具体产品信息,同时保留威胁描述中的有用信息。本文收集的漏洞数据符合 CNNVD 漏洞内容描述规范中规定的统一格式,包括受影响实体和漏洞内容两部分。受影响实体部分描述了漏洞所在软件或产品的基本信息,漏洞内容部分由漏洞类型、原因、利用方式和受影响版本组成。首先,我们将漏洞描述文本分割成句子列表,然后通过正则表达式查询并筛选出漏洞类型环节内容。然后,我们将这个句子和后面的句子进行连接,形成保留的有用信息的描述。因此,去除漏洞类型环节前置的受影响实体部分,保留漏洞内容描述。考虑到授权问题和跨站脚本这两种威胁类型的数量与其余威胁类型的数量之比超过 10∶1,而且因为威胁种类较多而多数类样本只有两类,因此本文选

择欠采样中的 EasyEnsemble 方法[71]对样本数据进行优化。利用集成学习机制,将反例划分为若干个集合供不同学习器使用,这样对每个学习器来看都进行了欠采样,但在全局来看却不会丢失重要信息。

4.2 STRIDE 威胁自动分类

本节详细介绍了 TextCNN 文本分类模型与产生式规则相结合的 STRIDE 威胁自动分类方法(TextCPR)。

(1) 自动威胁分类

用于文本分类算法的深度学习模型的主流选择包括 TextCNN[72-73]、TextRNN[73]、BERT[74]、Transformer[75]等模型。其中,TextCNN 模型相对轻量,具有高效、灵活、易于调整等优势。该模型可以更快的训练速度提供理想的效果[72],实际应用时适用性更强。其他模型如 BERT、Transformer 模型调参过程相较复杂,由于模型结构不公开、复杂性高、专业知识高等因素,调整难度大,在实际应用中的灵活性、适用性相对较弱。TextRNN 模型更适合超长文本的训练任务。由于威胁分类可以看作是文本多分类任务,并且数据处理步骤获取的威胁文本长度通常会在 50 字符以内,所以本文提出了基于 TextCNN 的威胁分类模型,针对 CNNVDTC 威胁描述的短文本进行分类。威胁分类模型分为 3 个模块,包括词向量预处理模块、包含卷积层和池化层的特征提取模块和回归分类模块。TextCNN 漏洞分类流程如图 4 所示。

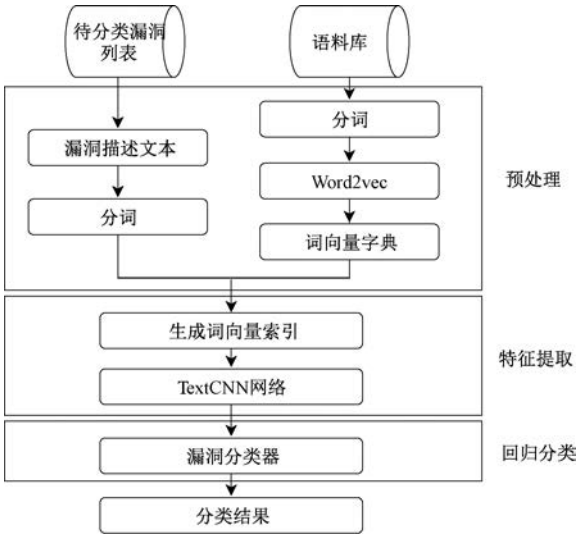

图 4　TextCNN 漏洞分类流程

首先是词向量预处理模块,对语料库中的处理后的漏洞数据进行分词和去除停用词的预处理,再通过 Word2vec 算法对漏洞描述语料进行词向量预训练,构建词向量字典。将分词后的漏洞描述文本通过词典映射转换为词向量索引,进而提取每个词的特征向量,作为 TextCNN 模型的输入层。

然后是特征提取模块,将输入层得到的固定维度词向量输入到特征提取模块,经卷积层后送入激活层、池化层和全连接层。TextCNN 的数据只有一维卷积,卷积公式为

$$L_{\text{out}}=\left[\frac{L_{\text{in}}+2\times\text{padding}-\text{dilation}\times(\text{kernelsize}-1)}{\text{stride}}-1\right],$$

其中,L_{out} 和 L_{in} 分别为输入序列长度和输出序列长度,padding 对输入的每一条边补充 0 的层数,kernelsize 为卷积核的尺寸,卷积核窗口在句子长度的方向上滑动,进行卷积操作[72]。

最后是分类模块,首先对特征提取模块获得的结果进行最大池化处理,主要是从多个值中取一个最大值。目的是在保持主要特征的前提下减少参数数量,在做到加速计算的同时防止模型过拟合。然后本文使用 softmax 层对威胁文本特征进行预测分类。在进行数据训练时采用交叉熵函数来计算威胁分类损失,公式为

$$L=-\sum_{i}^{n}\hat{y}\ \ln f(z_i)=-\ln f(z_k),$$

$$f(z_k)=\frac{e^{z_k}}{\sum n}e^{z_i}=\frac{e^{z_{k-m}}}{\sum n e^{z_{k-m}}},$$

$$m=\max z_i。$$

式中 $\hat{y}$ 为标签值,k 为输入文本标签所对应的神经元,$f(z)$ 为 softmax 回归函数,其中 m 为输出的最大值,即威胁分类输出概率。虽然 STRIDE 方法需要通过分析威胁影响范围进行分类,与威胁文本的语义环境存在差异,但是 CNNVDTC 威胁类别与威胁文本的语义环境相符。所以通过使用基于 TextCNN 实现的威胁分类模型,可以得到通过输入的威胁文本相对应的 CNNVDTC 威胁类别。

(2) 产生式规则匹配

由于目前网络威胁数据内容描述的重点是攻击者的行为,而 STRIDE 类别是从仿冒、篡改、抵赖、信息泄露、拒绝服务和特权提升六个维度对威胁的影响进行定性分析和分类。这就导致很难甚至无法直接从威胁的描述文本中直接获取到 STRIDE 类别,从而造成了通过分类器获取威胁 STRIDE 类别效果不理想的现象。CNNVD 公布的漏洞分类指南中包含对每种漏洞类别的常见后果和影响范围的描述[76],本文根据 CNNVD 漏洞分类指南和 STRIDE

的影响范围属性得出如表 5 所示的对应关系。然后利用 CNNVDTC 类别与 STRIDE 类别对应关系建立产生式规则，实现通过 TextCNN 威胁分类模型得到的 CNNVDTC 类别获取对应的 STRIDE 类别。通过算法 1 实现从 CNNVDTC 类别到 STRIDE 类别转换的产生式规则匹配方法。算法的输入分别是 STRIDE 的 6 种类别、CNNVDTC 的 11 种类别和根据表 5 生成的 CNNVDTC 类别与 STRIDE 类别对应关系列表。输出是输入威胁对应的 STRIDE 类别。具体实现为输入的威胁 CNNVDTC 类别通过“if...then...”形式的产生式规则机查询到对应的 STRIDE 类别。

表 5 CNNVD 漏洞类别与 STRIDE 类别对应关系

CNNVDTC 类别	STRIDE 类别
信息泄露(Information disclosure)	I
跨站脚本(Cross site scripting)	S,I
跨站请求伪造(Cross site request forgery)	S,T
注入(Injection)	T,I
路径遍历(Path traversal)	I
授权问题(Authorization issues)	E
后置链接(Link following)	T
资源管理错误(Resource management errors)	D
加密问题(Encryption issues)	I
代码问题(Code issues)	D,R
权限许可和访问控制问题 (Permission, privilege and access control issues)	S

算法 1. CNNVDTC 类别与 STRIDE 类别产生式规则机

输入：*stride*=['S','T','R','I','D','E'] //STRIDE 类别

输入：*base _ threat* = ['*s*'] //威胁分类步骤中生成的 CNNVDTC 类别

输入：*shine _ upon* = 'S':[...],'T':[...],'R':[...],'I':[...],' D': [...],' E': [...] //CNNVDTC 与 STRIDE 类别对应关系列表

输出：*res*//对应的 STRIDE 类别

1. WHILE *type*, *content* in *shine_upon*. *items* DO //对 *shine_upon* 进行遍历，*type* 和 *content* 分别对应 *shine_upon* 中的 STRIDE 类别以及 STRIDE 类别对应的 CNNVDTC 类别
2. IF *s* in *content* THEN//如果生成的 CNNVDTC 类别 *s* 在 *shine_upon* 中存在
3. *res*. *append*(*type*) //将 CNNVDTC 类别对应的 STRIDE 类别 *type* 追加到 *res* 中
4. ENDIF
5. ENDWHILE
6. RETURN *res* //返回 *res*

4.3 关系三元组表达式提取

类型规则的自动构建需要提取威胁和组件之间的联系，然而这种联系很难从获取的威胁数据描述中直接提取。本文在构建规则模型时采用了三元表述法来描述威胁内容，通过突出威胁的攻击行为和影响，清晰地表达了 STRIDE 威胁，易于理解，进而体现出威胁与组件的关系，便于建立威胁和组件之间的联系。采用的三元表述方式定义了三个主要元素，第一是采取行为的主体，第二是主体采取的行为，第三是行为造成的后果。第一元素描述了造成威胁的原因，第二元素和第三元素与威胁的 STRIDE 类别关联形成如表 6 所示的特定形式和表达，如：“攻击者可以{绕过身份验证}，导致{仿冒}问题”关联类别 S，该形式为后续类型规则匹配提供实现基础。其中第二元素是对根据不同的 STRIDE 类别威胁攻击者会采取宏观层次攻击行为的描述，第三元素是对根据不同的 STRIDE 类别威胁导致的不同问题描述。

表 6 STRIDE 类别与第二元素和第三元素对照表

STRIDE 类别	第二元素	第三元素
S	攻击者可以绕过身份验证	导致仿冒问题
T	攻击者可以进行数据篡改或代码执行	导致篡改问题
R	攻击者可以否认恶意行为	导致抵赖问题
I	攻击者可以读取敏感信息	导致信息泄露问题
D	攻击者可以发起特定攻击	导致拒绝服务问题
E	攻击者可以进行未授权的访问	导致越权问题

本文通过构建一个威胁描述内容关键主体提取的方法来实现三元表达式的自动生成。该方法提取的是威胁描述内容中能对关键行为进行描述的句子或者短语组，与文本摘要任务相似[77]。本文规定的三元组表达式中第一类元素是行为主体，即威胁文本中所关注的攻击行为。根据 Li 等人[78]提出的生成语法的最简方案框架，肯定了汉语存现句中动词短语组能够更好地对内容进行概括。因此，本文基于提取式摘要方法[77]，采用动词短语组对威胁文本重点内容提取作为三元组表达式的第一类元素。

如图 3 所示的步骤 3 为本文构建关系三元组表达式提取的流程。(1)首先通过 HanLP① 工具中的短语句法树、词性标注算法构建句法与词性流水线。(2)然后将经过数据处理后的 CNNVD 漏洞情报信息作为基础语料库。采用词频-逆文档频率(TF-IDF)加权技术评估漏洞内容词汇在语料库中的重要性，计算威胁语句分词后所有词语的 IDF 值，生成由每个单词 IDF 值组成的文档。然后计算生成

① https://www.hanlp.com。

每个词语对应的 TF 值。再计算 TF-IDF 值后将所有词语进行倒序排序。(3)接下来根据第一步构建的句法与词性流水线,以权重最高的动词为基础从下向上遍历,遍历到动词短语组后放入相关动词短语组列表中。因为权重最高的动词可能会在文本中出现多次,并且动词短语组中也可能会包含动词短语组,所以会出现多个结果。为了在多个相关动词短语组中获取核心动词短语组,本文通过第二步生成的词汇权重表,采用权重叠加赋值的方法将每个相关动词短语组进行排序,获取权重值最高的核心动词短语组。(4)最后将第三步获取的核心动词短语组作为第一类元素,将通过 TextCPR 方法获取的威胁 STRIDE 类别按照如表 6 所示的特定表达对照匹配第二类元素和第三类元素,最终组装出关系三元组表达式。

4.4　类型规则和交互规则的生成

基于威胁 STRIDE 类别以及关系三元组表达式进一步生成威胁识别规则,包括类型规则和交互规则。

(1) 类型规则匹配生成

本文将制定的威胁识别规则模型形成的基础规则库作为语料库,首先通过 STRIDE 类别与组件间关联规则确定组件范围,然后本文分别将通过关系三元组表达式提取模块生成的表达式和输入的威胁文本内容与威胁识别类型规则中的对应信息进行语义相似度匹配,再根据两者相似度值中最大的结果匹配对应类型规则。最后将匹配到的类型规则中的组件进行提取,然后将提取后的组件与威胁内容、生成的三元组表达式、威胁的 STRIDE 类别进行结合形成完整规则的格式,最终将新生成的类型规则填充入类型规则库中。如图 3 中步骤 4 所示为类型规则匹配生成的流程。

本文借助 Gensim[①] 模块辅助计算威胁描述文本和生成的三元表达式与基础类型规则库内容相似度。该相似度计算算法的伪代码描述如算法 2 所示。首先借助 jieba[②] 库进行中文分词,然后通过哈工大构建的停用词文档去除停用词。然后遍历分词后的结果集,计算每个词的频率。再根据词频结果进行排序编号,创建单词与编号之间的映射字典。接着通过词袋表示法对词表中的每一个词在该文本出现的频次进行记录,将待比较的文档转换为向量。然后将每一条相关威胁识别规则通过词袋表示法转换为向量建立漏洞相关语料库。因为词袋表示法只能表示每个词特征在当前文本中的重要程度,所以使用 TF-IDF 对词特征加权,以表示词特征在整个语料中的重要程度,最终将整个语料库转为 TF-IDF 表示方法。再使用上一步得到的带有 TF-IDF 值的语料库建立索引。最后将待比较文档转换为 TF-IDF 表示方法,然后使用余弦相似性度量计算要比较的文档与语料库中每篇文档的相似度,最后返回相似度最大值的文档。

算法 2. 威胁文本相似度计算核心算法

输入:类型规则库

输出:*res* //按相似度顺序排列的文档列表

```
join(jieba.cut(line)).split()
frequency = defaultdict(int)
WHILE text in res DO
  WHILE word in text DO
    frequency[word]+ = 1
  END WHILE
END WHILE
dictionary = corpora.Dictionary(texts)
NewVec = dictionary.doc2bow(res)
corpus = [dictionary.doc2bow(text) for text in texts]
index = similarities.MatrixSimilarity(corpus)
sims = index[NewVec]
WHILE i in range(length) DO
  res.append(sims[i])
END WHILE
RETURN res
```

(2) 交互规则提取生成

交互规则聚焦于数据流,通过数据流串起的组件之间的交互而产生的特定威胁。换言之,部分威胁的产生是基于特定组件之间产生的数据流。因此,交互规则的制定将主要依托于对拥有同样关联威胁的组件之间的数据流进行判断得出,在类型规则的基础上提取出交互规则。本文提取出类型规则库中重复出现的威胁所对应的组件,对通常会有数据流交互的组件进行分析整理,根据交互的组件以及数据流的方向,形成源(Source)组件和对应的目标(Target)组件集合的关系表,如表 7 所示。在交互规则自动构建过程中,首先提取选择类型规则库中的不同组件重复出现的威胁及其对应的规则,然后查询提取出的重复出现的威胁所对应的组件是否存在于组件集合关系表中,若存在则可以提取出交互规则中的 Source 组件和 Target 组件。将提取的

① https://github.com/RaRe-Technologies/gensim。
② https://github.com/fxsjy/jieba。

类型规则中重复关联的威胁与提取的 Source 组件和 Target 组件相结合形成交互规则。最后将提取生成的交互规则收录到交互规则库中，与类型规则库共同构成完整的威胁识别规则库。

4.5 规则库自动更新

由于威胁的不断更新迭代，本文通过实现类型规则库和交互规则库的自动更新，来保持构建的威胁识别规则模型的时效性。CNNVD、CWE 和 OWASP Top Ten 是目前能够保持更新且都得到大量知名互联网企业支持的网络威胁信息平台，而且三个平台都有 XML 文件格式的数据，便于统一处理，所以本文以 CNNVD、CWE 和 OWASP Top Ten 平台的数据作为规则库更新的数据源。

OWASP Top Ten 列出了最常见的 Web 应用安全风险，可以通过对 OWASP Top Ten 对应的 CWE 列表进行汇总。OWASP Top Ten 和 CWE 都涵盖了软件安全和网络安全的相关数据。CNNVD 包含 Web 安全漏洞信息，提供了高质量、全面的软件安全和网络安全威胁数据，其描述包含威胁类别和影响信息。这三个数据源相结合可以构建更全面的 Web 安全领域的威胁数据，其包括软件安全和网络安全威胁数据。它们之间相互依赖、相互影响，存在互补性。Web 安全主要关注保护 Web 应用程序和服务的安全性，防止如跨站脚本(XSS)、SQL 注入等 Web 攻击，确保用户交互和数据交换的安全性。软件安全数据主要涉及应用程序尤其是 Web 应用程序的安全性，网络威胁数据关注的是网络环境中可能存在的安全威胁和攻击行为。Web 应用攻击和网络威胁的攻击手段相互影响和依赖。Web 应用攻击如 SQL 注入、XSS 等主要是由于应用程序的安全漏洞导致的，这些攻击可能导致网络威胁如未授权访问、数据泄露等严重后果；网络威胁如恶意软件、钓鱼攻击、DDoS 攻击等可能为 Web 应用攻击创造条件。软件安全数据关注应用层，网络安全数据关注网络层，两者共同保障 Web 安全。这些数据源适合作为规则库更新的基础数据源，从多个数据源可以获取更丰富、更全面的数据。

表 7 组件关系表

发生者(Source 组件)	被作用者(Target 组件)
Browser	Traffic Manager, Identity Server, App Server, Web Application, Database, SQL Database, File System, NoSQL Database
PC Client	Traffic Manager, Host, Identity Server, Redis Cache, Cache, Database, SQL Database, File System, NoSQL Database
Mobile Client	Traffic Manager, App Server, Web Application, Database, SQL Database, File System, NoSQL Database
Traffic Manager	Redis Cache, Cache
Host	Redis Cache, Cache, File System
Identity Server	Database, SQL Database, NoSQL Database
App Server	Redis Cache, Cache, Database, SQL Database, File System, NoSQL Database
Web Application	Database, SQL Database, NoSQL Database

CNNVD 平台数据保持每日更新，分批提供下载以年、月、日为单位的 XML 数据文件，每个月、每年都会提供新的漏洞整合版本(每月月初的第一个工作日生成上一个月新生成的漏洞整合版本，当生成 12 个以月为单位的 XML 数据文件后整合为本年度的漏洞 XML 数据文件)。通过使用本文构建的特定的 CNNVD 内容 XML 解析器获取数据。CWE 平台每 3 个月更新一次，每年会在 4 月、6 月、10 月、12 月发布最新版本，但是该平台每次发布最新版本时会包含收录的全部漏洞条目，所以本文在下载 XML 文件后会将筛选出的重复内容进行排除。OWASP Top Ten 更新频率为 3 至 4 年，每次发布的新版本会包含近年来影响最大的十个安全风险漏洞。

本文根据三个数据源平台的更新频率，对更新的数据进行定时获取。新获取的数据通过前文所描述的各个步骤成功匹配到相应的组件后形成统一格式化处理的类型规则条目或交互规则条目，经过专家审核，然后填充入威胁识别规则库，保证了规则库的时效性。

5 实验评估

本节通过实验对提出的威胁自动分类方法 TextCPR 和规则库自动构建方法 ACUTIRule 的有效性进行了评估。

5.1 威胁自动分类方法的实验评估

(1) 研究问题

为了评估本文提出的 STRIDE 威胁自动分类方法 TextCPR 的有效性，本文提出以下两个研究问题：

RQ1：与其他基线方法相比，本文所提出的 TextCPR 方法的有效性如何？RQ1 旨在通过将本文所提出的自动分类方法 TextCPR 与基准方法进行对比，验证 TextCPR 方法的有效性。本文将 Is-

lam 等人[66]提出的 SmartValidator 框架作为基线方法,该框架用于网络威胁数据自动识别和分类。SmartValidator 方法利用 ML 技术从攻击、威胁类型、威胁名称和威胁级别四个属性方面对威胁进行自动识别和分类。选择其作为基线出于以下考虑:在威胁类型分类上,一方面,该方法研究了如何实现威胁的自动分类,这与我们的研究目标是一致的;另一方面,该方法在开源威胁信息平台 MISP 上平均准确率达到 87%,被认为是目前最先进的方法。

RQ2:本文所提出的 TextCPR 方法中各部分对方法整体性能产生什么影响? RQ2 旨在评估 TextCPR 方法中的各个部分即 TextCNN 威胁分类模型、产生式规则匹配对方法整体性能的影响。

(2) 评估指标

本文将威胁 STRIDE 类别自动化分类视为一个多分类问题,方法的目的是识别威胁文本描述属于的 STRIDE 类别。在分类领域中,有诸多模型评价指标,避免单一指标带来的评估偏差。分类模型的评估指标主要包括:准确率(Accuracy)、精确率(Precision)、召回率(Recall)、$F1$-score 等。

(3) 实验设置

本文选取归纳出的 11 种最常见的 CNNVDTC 类型进行实验,分别是:信息泄露、跨站脚本、注入、跨站请求伪造、路径遍历、授权问题、后置链接、资源管理错误、加密问题、代码问题、权限许可和访问控制问题。其中每个漏洞信息包含以下属性项:漏洞名称、CNNVD 编号、漏洞类型、漏洞描述。本文采用 CNNVD 的漏洞描述文本作为输入,然后按照 5∶1∶1 的比例分配训练集、验证集、测试集。数据集划分的具体信息如表 8 所示。由于类别不平衡的问题,在实验过程中采用欠采样方法进行优化。

本文还采用 CWE 和 CVE 平台的开源威胁数据作为实验的测试数据集,旨在验证不同描述粒度数据下的实验效果。这两种类型的数据对攻击方式描述的粒度不同,相较于 CWE 平台的数据而言,CVE 平台的数据对攻击者的行为描述粒度更细,详细到攻击者使用的参数或者具体攻击位置。其中 CWE 平台有 1137 条数据,剔除硬件相关漏洞数据后为 642 条数据。CVE 平台有 16.5 万条数据。由于这两种数据都没有与 CNNVDTC 类别或 STRIDE 类别相关的属性,因此需要人工标注 CNNVDTC 类别和 STRIDE 类别。考虑到两种数据的数据量和人工标注的工作量,最终确定了分别将标注后的 600 条 CWE 漏洞信息和 1000 条 CVE 漏洞信息作为测试数据集。

实验采用 Google 开源框架 TensorFlow 搭建 TextCNN 网络进行验证漏洞分类效果,在 10 核 3.70GHz Intel(R) Core(TM) i9-10900X CPU 和 NVIDIA GeForce RTX 3090 GPU 的服务器上运行所有的实验。同时为了验证 TextCNN 模型对威胁按照 CNNVDTC 类别分类的有效性,本文还将比较与使用传统 SVM 分类方法的效果差异。实验过程中的超参数如下表 9 所示。超参数 embedding 决定了每个词向量的表示空间大小,影响模型对文本

表 8　CNNVD 数据集划分情况

CNNVDTC 类别	训练集	验证集	测试集
信息泄露(Information disclosure)	5000	1000	1000
跨站脚本(Cross site scripting)	10000	2000	2000
跨站请求伪造(Cross site request forgery)	3352	670	670
注入(Injection)	4200	840	840
路径遍历(Path traversal)	4510	902	902
授权问题(Authorization issues)	5000	1000	1000
后置链接(Link following)	3000	600	600
资源管理错误(Resource management errors)	5000	1000	1000
加密问题(Encryption issues)	2485	497	497
代码问题(Code issues)	45000	9000	9000
权限许可和访问控制问题(Permission, privilege and access control issues)	2440	488	488

表 9　实验的超参数设置

参数名称	参数含义	取值
embedding	词向量维度	64
SeqLength	序列长度	600
NumClasses	威胁类别数	11
NumFilters	卷积核数目	256
KernelSize	卷积核尺寸	5
VocabSize	词汇表大小	5000
HiddenDim	全连接层神经元	128
DropoutKeepProb	dropout 保留比例	0.5
LearningRate	学习率	1e-3
BatchSize	每轮训练大小	64
NumEpochs	总迭代轮次	20
PrintPerBatch	每多少轮输出一次结果	100
SavePerBatch	每多少轮存入 tensorboard	10

语义的理解和捕捉能力。SeqLength 设定输入序列的固定长度。NumClasses 用于定义分类任务的类别数。NumFilters 决定了每个卷积层提取的特征数量。KernelSize 定义卷积核的大小，影响局部特征的捕捉范围。VocabSize 决定了词汇表的大小，直接影响嵌入层的参数量，进而影响模型的复杂度和性能。HiddenDim 决定了全连接层的神经元数量，直接影响模型的表达能力。DropoutKeepProb 用于控制训练过程中每个神经元被保留的概率，通过在训练中随机丢弃部分神经元以防止过拟合。LearningRate 控制参数更新步长，影响模型收敛速度和稳定性。BatchSize 决定了每次训练时输入的数据量，影响模型的训练速度、内存占用和性能。NumEpochs 决定模型训练的轮数，用于优化模型训练，确保其在合理时间内充分学习数据特征，达到最佳性能。PrintPerBatch 控制训练过程中每隔多少个批次(batch)打印一次训练信息，用于监控训练进度、调试模型或分析性能。SavePerBatch 控制将训练过程中的信息保存到 tensorboard 的频率，影响训练过程的灵活性和资源使用效率。

(4) 实验结果与分析

RQ1 方法有效性分析：为了探究方法的有效性，将本文提出的 TextCPR 方法在 CNNVD 和 CWE 两个不同的数据集上与基线方法进行了对比实验，得到的结果如表 10 所示。

从表 10 可以看出，TextCPR 在 CNNVD 数据集上的精度达到 0.925，召回率达到 0.876，$F1$-score 达到 0.893。与基线方法 SmartValidator 相比，TextCPR 方法在精度、召回率和 $F1$-score 上分别提高了 0.112、0.082 和 0.092。同时，在 CWE 数据集上各项评价指标的结果也优于 SmartValidator 方法。实验结果表明本文提出的 TextCPR 方法在威胁分类任务上的有效性。即使在不同粒度的威胁描述数据集(如 CWE)上，TextCPR 方法的精度也达到 0.873，召回率和 F1-score 均超过 0.8。

SmartValidator 通过使用标签编码的方式对分类变量进行编码，然后使用计数向量化和 TF-IDF 方法作为将文本编码为数值的特征工程方法。其中通过标签编码的方式很好地学习了威胁文本的局部特征，但是却忽略了威胁文本的全局语义信息，而且通过手动对分类变量进行标签编码的方式也会造成丢失关键信息的情况。以上的局限性使得最终的性能没有 TextCPR 方法优异。而本文所提出的 TextCPR 方法综合考虑了全局和局部的威胁文本语义信息，首先因为使用的数据具有高质量的类别标注，所以能够充分学习威胁文本的局部特征。而且本文通过对 CNNVDTC 类别定义的分析，建立了类别与威胁影响间的联系，借助该联系能够充分地学习全局的威胁语义信息。

RQ1 的回答：与基线方法相比，本文提出的 TextCPR 方法在 CNNVD 和 CWE 两种不同的数据集上显著提高了威胁分类的性能，并实现了 STRIDE 威胁的自动识别和分类。

RQ2 方法各部分作用影响分析：为了验证 TextCPR 方法中各部分对方法整体性能的影响，本文通过对比结合了 TextCNN 威胁分类模型和产生式规则方法的 TextCPR 方法与单独使用 TextCNN 威胁分类模型的效果差异，验证产生式规则匹配部分对方法整体性能的影响。再通过对比 TextCNN 分类模型与 SVM 分类方法在威胁分类任务下的效果差异，验证 TextCNN 威胁分类模型部分对方法整体性能的影响，得到的结果如表 11 和表 12 所示。

表 10 TextCPR 与基线方法的对比实验结果

方法	CNNVD			CWE		
	Precision	Recall	$F1$-score	Precision	Recall	$F1$-score
SmartValidator	0.813	0.794	0.801	0.783	0.768	0.772
TextCPR	**0.925**	**0.876**	**0.893**	**0.873**	**0.849**	**0.861**

表 11 TextCPR 中产生式规则匹配部分的效果

方法	CNNVD			CWE	CVE
	Precision	Recall	$F1$-score	Accuracy	Accuracy
TextCNN	0.51	0.388	0.536	0.487	0.366
TextCPR	**0.925**	**0.876**	**0.893**	**0.907**	**0.826**

表 12 TextCPR 中 TextCNN 部分的效果

方法	CNNVD			CWE	CVE
	Precision	Recall	$F1$-score	Accuracy	Accuracy
SVM	0.53	0.442	0.474	0.374	0.297
TextCNN	**0.933**	**0.751**	**0.813**	**0.922**	**0.873**

表 11 展示了单独使用 TextCNN 分类模型和 TextCPR 方法在 CNNVD、CWE、CVE 三个数据集上进行实验的结果。TextCNN 分类模型在 CNNVD 数据集上的 Precision 为 0.51，Recall 为 0.388，$F1$-score 为 0.536，TextCPR 方法相比于单独使用 TextCNN 分类模型在 Precision、Recall、$F1$-score

上分别提高了 0.415、0.488 和 0.357。同时在 CWE 和 CVE 数据集上 TextCPR 方法对于威胁 STRIDE 类别分类任务的 Accuracy 也明显高于单独使用 TextCNN 分类模型的方法。该对比实验结果表明了产生式规则部分明显提升了 TextCPR 方法整体的性能。产生式规则匹配能够解决 STRIDE 类别与威胁的语义环境缺乏直接联系的问题,选择与威胁语义存在联系的 CNNVDTC 类别作为过渡,从而避免了通过 TextCNN 直接进行 STRIDE 威胁类别自动分类任务的局限性。因此,TextCPR 方法可以在 STRIDE 威胁标签数据不足的挑战下,通过产生式规则匹配实现 CNNVDTC 类别与 STRIDE 类别的自动转换,实现 STRIDE 威胁自动分类以及提高分类性能。

TextCNN 模型将输入文本经过 embedding 操作之后映射成对应的词向量,实现从语义空间到向量空间的映射。这种方式能够有效地学习威胁文本语义的局部和全局特征,同时尽可能在向量空间保持原样本在语义空间的关系,但是当分类任务中的类别与语义环境缺乏对应关系时便不能得到有效地学习。以上的局限性使得直接使用 TextCNN 模型对标注 STRIDE 类别的数据进行学习的效果较差,导致最终的性能没有 TextCPR 方法优异。而本文所提出的 TextCPR 方法选择与威胁语义存在联系的 CNNVDTC 类别作为过渡,使得 TextCNN 模型发挥其在文本分类任务中的优势。再经过产生式规则匹配实现 CNNVDTC 类别与 STRIDE 类别的自动转换,补充了单独使用 TextCNN 模型的 STRIDE 威胁分类任务中缺失的语义连接,最终避免了用 TextCNN 模型进行 STRIDE 威胁自动分类任务的局限性。

表 12 则展示了 TextCNN 分类模型和 SVM 分类模型在 CNNVD、CWE、CVE 三个数据集上进行实验的结果。TextCNN 分类模型在 CNNVD 数据集上的 Precision 为 0.933,Recall 为 0.751,F1-score 为 0.813。相较于 SVM 分类器,TextCNN 分类模型在 Precision、Recall、F1-score 上分别提升了 0.403、0.309、0.339。同时在 CWE 和 CVE 数据集上 TextCNN 方法对于威胁 CNNVDTC 类别分类任务的 Accuracy 也明显高于 SVM 分类模型。该对比实验结果表明了 TextCNN 威胁分类部分在一定程度上提高了 TextCPR 方法的整体性能。

本实验中对比的两种不同的方法和两种不同的模型在 CVE 数据上的表现都明显不如 CWE 数据上的表现,这是因为 CVE 的描述信息中包含大量产品信息等无用信息描述,造成了数据干扰降低了准确率,这也表明本文在数据处理阶段将 CNNVD 数据中的产品信息等无用信息去除的有效性。

TextCNN 威胁分类模型在 CNNVD、CWE 和 CVE 数据集上的性能均优于 SVM 分类算法。产生式规则匹配能够解决 STRIDE 类别与威胁的语义环境缺乏直接联系的问题,从而避免了通过 TextCNN 直接进行威胁 STRIDE 类别自动分类任务的局限性,因此 TextCPR 方法可以在威胁 STRIDE 标签数据不足的挑战下,通过产生式规则匹配实现 STRIDE 威胁分类以及提高分类性能。方法的各部分的性能优势说明了 TextCPR 方法对于 STRIDE 威胁自动分类任务是一个很好的选择。

RQ2 的回答:产生式规则匹配可以帮助解决 STRIDE 类别与威胁的语义环境之间缺乏直接联系的问题,并相应地提高整体方法性能。与 SVM 分类算法相比,TextCNN 威胁分类模型在 CNNVD、CWE 和 CVE 数据集上提高了威胁分类的性能。TextCPR 方法将 TextCNN 分类模型与产生式规则匹配相结合,可以有效地对 STRIDE 威胁进行自动化分类,并在 STRIDE 标签威胁数据不足的挑战下提高分类性能。

5.2　规则模型自动构建方法的实验评估

(1) 研究问题

为了验证所提出的威胁识别规则模型自动构建方法 ACUTIRule 及其各部分在自动构建威胁识别规则模型方面的有效性,本文提出以下研究问题:

RQ3:采用不同的词类作为三元组表达式核心元素对威胁文本重点内容提取是否有影响? RQ3 旨在验证将动词词组作为三元组表达式的第一元素是否可表达威胁的重点内容,通过将核心动词词组提取的三元组表达式与人工提取的三元组表达式对比来验证该方法的实际效果。同时,本文还将设计一组对照实验,使用名词词组作为三元组表达式的核心元素,以进一步说明采用动词词组作为核心元素的有效性。

RQ4:相比于单一的类型规则或交互规则,本文提出的结合类型规则和交互规则的威胁识别规则是否更全面地涵盖威胁? RQ4 旨在说明本文提出的结合类型规则和交互规则的威胁识别规则相较于单一的类型规则或者交互规则能够更加全面地覆盖威胁。该研究使用结合类型规则和交互规则的威胁识别规则在开源威胁数据集上进行归纳分类,并对比

单一的类型规则或者交互规则在相同数据集下归纳分类的效果，来验证本文提出的威胁识别规则在涵盖威胁方面的有效性和全面性。

RQ5：相比于人工构建威胁识别规则模型的方法，本文提出的 ACUTIRule 方法是否具有优势？RQ5 旨在验证本文所提出的 ACUTIRule 方法相较于人工构建威胁识别规则模型的方式是否具有优势。该研究通过对比 ACUTIRule 方法和人工构建方法在构建规则模型的速度和人力消耗等指标上的差异，体现 ACUTIRule 方法相较于人工构建威胁识别规则模型的方式在自动化方面的优势，并通过将同类型规则库数据集作为测试集验证产出结果的准确性。

（2）评估指标

本文在验证使用核心动词短语组作为三元组表达式核心元素的有效性时，采用文本相似性度量和人工判定可用率相结合的方法，使用了平均相似率和可使用率作为方法的评估指标。此外，本文为了验证所提出的规则构建方法的有效性，使用准确率（Accuracy）这个评估指标评估构建规则的准确性。

（3）平均相似率（Average Similarity）

相似率（Similarity）是通过比较文本间的文本相似度来体现两者的语义相似度。而平均相似率是将每组文本间的相似率进行叠加后计算平均值，用来体现方法与数据间的整体匹配度。其计算公式如下：

$$AS=\frac{\sum_{i=1}^{n}\text{Similarity}}{n}。$$

（4）可使用率（Usability）

可使用率是指由领域专家人工判定可使用的数据占整体数据的比例，用来直观体现方法的效果。其计算公式如下：

$$\text{Usability}=\frac{\text{EJT}}{\text{EJT}+\text{EJF}}。$$

其中，EJT 为经过判定为可使用的数据数量，EJF 为经过判定为不可使用的数据数量，两者相加为总体数据数量。

（5）实验设计

本节分别根据提出的三个研究问题设计对比实验对 ACUTIRule 方法的有效性进行验证。

RQ3：本文通过提取威胁描述文本中的攻击行为组成三元表达式，与自动摘要任务相似。评估自动摘要的方式有多种[79]，其中通过句子选择进行评估的方法主要是人工选择提取，需要大量人力成本，基于任务的评估方法适用于特定领域，需要提前建立分类语料库。所以本文选择通过基于内容的相似性度量方式在更细粒度的级别上计算两个文档之间的相似性。

本文将自动生成的三元组表达式与基础规则库中人工总结的三元组表达式做文本相似度匹配，采用 Word2vec 将词转化为向量，使用余弦相似性度量来计算其相似性。生成的三元组表达式的可使用率由领域专家人工判定。同时本文还将比较与使用核心名词词组作为三元组表达式第一元素的效果差异。本实验数据来自人工构建的基础威胁识别规则库，将根据威胁描述文本自动生成的三元组表达式和基础规则库中人工总结的三元组表达式作为输入进行实验。

RQ4：为了验证结合类型规则和交互规则的威胁识别规则的有效性，对于本文所收集的威胁数据集进行随机抽取，然后分别单独使用交互规则和类型规则进行归纳，再使用结合类型规则和交互规则的威胁识别规则进行归纳，最终对比三者能够包含以及未能包含的威胁数目。本实验数据来自 CNNVD、CWE 和 CVE 开源项目，分别随机抽取 1000 条总共 3000 条数据作为测试数据集，通过人工划分的方式对数据进行划分。

RQ5：为了验证本文提出的 ACUTIRule 方法在威胁识别规则模型自动构建方面的有效性，本文首先对采用的基础类型规则库进行拓展作为测试规则集来验证准确率。我们从 CVE 网站随机获取 1000 条 Web 领域相关的威胁数据，分析确定威胁类别，将威胁描述以三元组表达式的格式进行描述。根据前文所描述的威胁识别规则模型制定方法，我们将这些威胁数据手动拓展为经过专家审核且符合使用标准的规则集，并将这 1000 条规则集作为测试数据。本实验将拓展的测试规则集中的威胁描述文本作为输入，然后通过 ACUTIRule 方法自动生成对应的 1000 条规则结果，并与手动拓展的正确结果进行对比，实际等同于将根据原始规则集作为语料库匹配出的组件与拓展规则集手动判定的组件进行对比，如果同一条威胁文本经过两种方法对应的组件相同则记为正确结果，反之记为错误结果，最终得出准确率。

最后将 ACUTIRule 方法构建规则模型的过程和结果通过构建速率、构建准确率和构建成本等量化指标与人工构建规则模型的方式进行对比。本文在 CVE 网站随机爬取 1000 条 Web 相关的漏洞信息，然后通过上述的两种不同的规则模型构建方式将 1000 条漏洞信息制定为符合使用标准的规则，并

记录各自使用的时间、人力成本,并将最后结果进行人工审核验证得出各自的准确率。

(6) 实验结果与分析

RQ3:为了验证所提出的采用动词词组作为三元组表达式第一要素对威胁文本重点内容提取的有效性,本文对以名词词组作为三元组表达式第一要素和以动词词组作为三元组表达式第一要素进行威胁文本重点内容提取这两种方式进行了实验结果对比,得到的结果如表 13 所示。

表 13 展示了通过两种不同方式在三元组表达式提取任务上的实验结果。通过动词词组作为三元组表达式第一要素构建的三元组表达式与现有的类型规则库中的三元组表达式的平均相似率达到 83%,由专家人工判定的可使用率达到了 92%,整体达到了良好的效果,相比于使用核心名词词组作为三元组表达式第一要素的方法在平均相似率、可使用率上分别提升了 21%、20%。实验结果表明了本文所提出的采用动词词组作为三元组表达式第一元素对威胁文本重点内容提取的有效性。

表 13　三元组表达式提取结果

方法	平均相似率 (Average Similarity)	可使用率 (Usability)
核心动词词组	**83%**	**92%**
核心名词词组	62%	72%

RQ3 的回答:由动词短语组组成的三元组表达式与基础类型规则库中三元组表达式的平均相似率达到 83%,专家人工判定的可使用率达到 92%。与使用核心名词短语组形成三元组表达式的方法相比,平均相似率和可使用率分别提高了 21% 和 20%。与名词短语组相比,由动词短语组组成的三元组表达式可以更好地表达威胁文本的关键内容。

RQ4:为了验证所提出的结合类型规则和交互规则的威胁识别规则的有效性,本文以单独使用交互规则和类型规则和使用结合类型规则和交互规则的威胁识别规则这 3 种方式对威胁进行划分,并将实验结果进行了对比。由表 14 的结果可知,当单独使用类型规则对威胁进行划分时,3000 条威胁中有 218 条不能成功划分,单独使用交互规则则有 1655 条威胁不能成功划分,而使用类型规则结合交互规则的方式则只有 27 条威胁未能成功划分。实验结果表明本文所提出的结合类型规则和交互规则的威胁识别规则的有效性,能够有效地对威胁进行划分,并且相比于单一的类型规则或交互规则,所提出的规则能够更全面地覆盖威胁。该方法能够有效地划分绝大部分威胁内容,并且具有高可解释性和全面性。

表 14　威胁划分结果

方法	成功划分数量	未成功划分数量
类型规则	2782	218
交互规则	1345	1655
类型规则结合交互规则	**2973**	**27**

RQ4 的回答:使用结合类型规则和交互规则的威胁识别规则成功划分威胁的数量大于使用单一类型规则或交互规则成功划分威胁的数量。本文提出的将类型规则和交互规则相结合的威胁识别规则可以有效地对威胁进行划分,与单一类型规则或交互规则相比,本文提出的规则可以更全面地覆盖威胁。

RQ5:为了验证 ACUTIRule 方法的有效性,本文首先通过对比在测试数据集上用 ACUTIRule 方法生成的规则与正确结果的差异来体现该方法的准确率,实验结果显示 ACUTIRule 方法在 100 和 1000 个漏洞数量下生成的规则与正确结果对比的准确率分别达到了 88%和 89.5%,体现出该方法可以有效地进行威胁识别规则模型自动构建。

然后将 ACUTIRule 方法与人工在构建规则模型的效率上进行对比,进一步说明本文提出的 ACUTIRule 方法能够明显提升工作效率,实验结果如表 15 所示。表 15 展示了 ACUTIRule 方法在构建规则模型的时间和人力成本方面的优势,能够明显提升工作效率,体现了 ACUTIRule 方法的自动化优势。就构建相同条目的漏洞为可使用的规则而言,ACUTIRule 方法所消耗的时间远少于人工构建方法,且不需要额外的人力成本。当输入的漏洞数量越多时该方法的优势越明显。虽然 ACUTIRule 方法在准确率方面的表现稍逊于人工构建规则的方式,但误差仅在 10%之内,且漏洞数量越多准确性差异越小。

表 15　ACUTIRule 方法与人工构建规则模型指标对比

方法	漏洞数量	消耗时间	人力规模	准确率
ACUTIRule 方法	100	29(s)	无需人工	88%
ACUTIRule 方法	**1000**	**64(s)**	**无需人工**	**89.5%**
人工构建	100	30(min)	两人	95.7%
人工构建	1000	6(h)	两人	94.9%

RQ5 的回答:与 100 条和 1000 条威胁的正确结果相比,ACUTIRule 方法的准确率分别为 88% 和 89.5%。该方法可有效地用于威胁识别规则模型的自动构建。在将相同条目的威胁构建为可使用

的规则方面，ACUTIRule 方法比手动方法花费的时间要少得多，并且不需要额外的人力成本。虽然 ACUTIRule 方法的准确性低于人工构建规则的方式，但误差仅在 10%之内。

6 应用实例

本节通过一个应用实例展示 ACUTIRule 应用的效果。通过将应用结果与权威工具微软 TMT 的结果进行对比，表明 ACUTIRule 应用能够有效支持威胁识别。

本文以基于 iNTegrity[5] 的文件完整性检查这个场景作为实例来进行论述。iNTegrity 是一款文件完整性检查工具，它可以读取资源（比如文件系统 filesystem 中的文件），确定自上次检查以来是否有任何文件或注册表项被更改。该场景中，Admin 通过前端页面向后端 Web Application 组件 INTegrity Admin Console 传递指令进行完整性检查。INTegrity Admin Console 一方面可以读取 Config data 和 Integrity files 中的数据，另一方面还可以通过传递命令给 Integrity host software，驱使其对 registry 和 filesystem 进行获取，从而将对应的资源完整性数据返还给 INTegrity Admin Console。INTegrity Admin Console 对数据进行判断后更新 Integrity files，并将完整性变动信息返回至 Admin。该 DFD 包含了最为常见、关联威胁最多的组件，如处理过程中的 Web Application 和 Host，数据存储中的 SQL Database 和 File System。同时，该项目包含 Browser 和 Web Application，也是十分具有典型性的架构。因此，其 DFD 具有典型代表性，该项目适合作为实例展示应用结果。

在图 5 中展示了 iNTegrity 的 DFD，可以看到该 DFD 主要包含的节点包括一个外部实体 Admin，两个处理过程 Integrity host software 和 INTegrity Admin Console，以及四个数据存储 registry、filesystem、Config data 和 Integrity files，而边一共包含 9 个数据流。实例通过对这些组件数据进行分析，为部分组件分配细粒度组件类型，如 INTegrity Admin Console 被分配为 Web Application，而 Integrity host software 则被分配为 Host 类型的处理过程。相应地根据描述，registry 可以认为存于关系型数据库中即 SQL Database，而 filesystem 则更适合为 File System 类型组件，其余的组件也被分配为较为合适的类型。将图 5 中的 DFD 转换为数据结构，转换后的数据用于匹配规则识别对应的威胁列表。

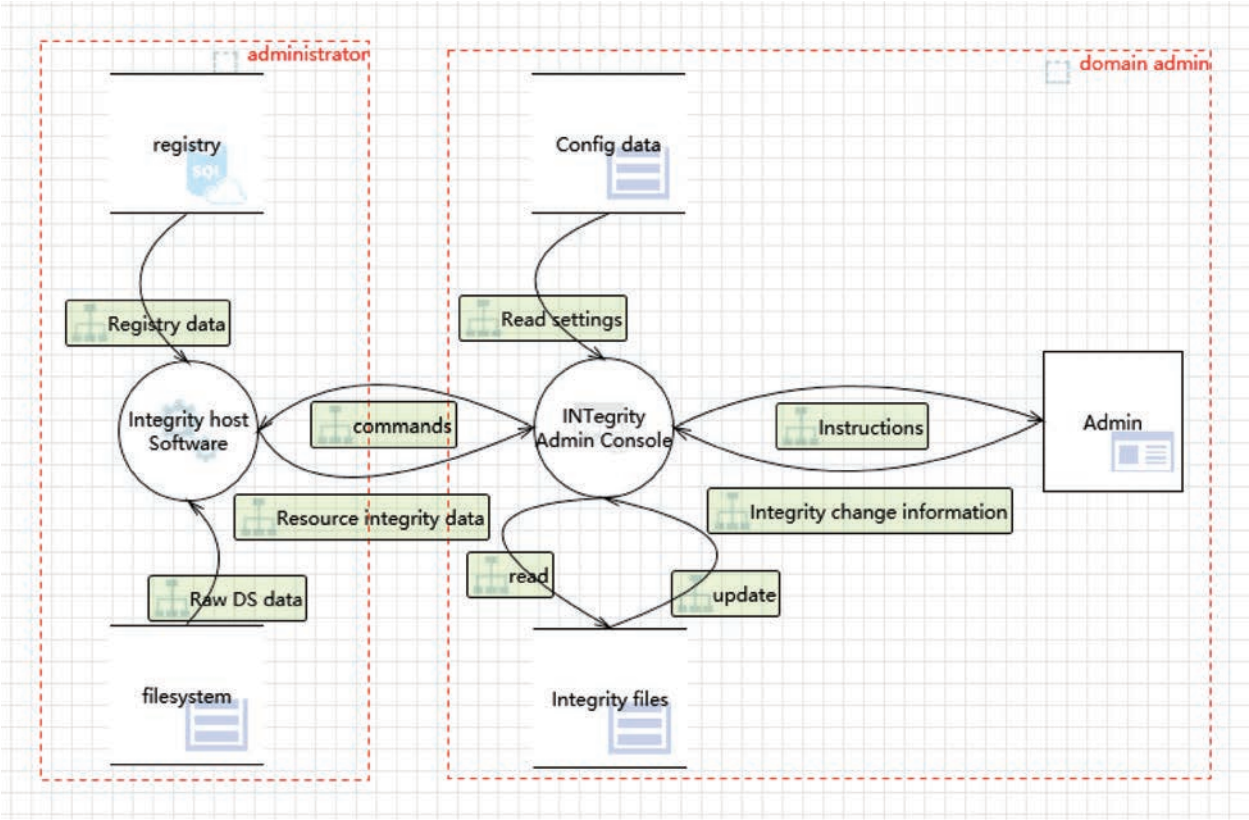

图 5 iNTegrity 实例的 DFD

将 ACUTIRule 应用识别出的威胁列表与业内权威工具微软 TMT 识别的威胁列表进行对比。ACUTIRule 应用基于所自动构建的全面的规则库来识别威胁,微软 TMT 基于人工构建的少量规则来识别威胁,因此主要从数量、描述等方面对比应用结果。

图 6 展示的是 ACUTIRule 应用中构建的规则,图 7 是 ACUTIRule 应用所识别的威胁列表。从图 7 中可以看到第 4 条威胁"由于渲染 UI 层或帧的不适当限制,攻击者可以绕过身份验证,导致仿冒问题"。该威胁作为前端开发中最为常见的风险之一,被成功识别得到。根据该条目,后续从业人员可以根据标题和描述对其进行理解。第 2 条威胁呈现的是"由于 cookie 属性不安全以及用户会话 cookie 可以被窃取,攻击者可以绕过身份验证,导致仿冒问题"。所提及的 cookie 问题,也是网页端中常见的攻击漏洞来源,在网页开发时需要切实注意此威胁。这两条威胁都是外部实体中的 Admin 组件可能面临的潜在的 Spoofing 类型的威胁。微软 TMT 的威胁列表的节选内容展示在图 8 中。图中可以明显看出存在大量条目是重复的,这是由于同一个威胁可以参与不同组件的规则中去。

0	1	2	3	4	5	6
GE.EE	S	39	322	由于无需实体身份验证的密钥交换，攻击者可以绕过身份验证，导致仿冒问题	该软件在不验证参与者身份的情况下与该参与者执行密钥交换。进行密钥交换将保持两个实体	
GE.EE	S	TH8	311	由于cookie属性不安全以及用户会话cookie可以被窃取，攻击者可以绕过身份验证，	会话cookies是服务器知道每个传入请求的当前用户身份的标识符。如果攻击者能够窃取用户令	
GE.EE	S	143	294	由于通过捕获-重放绕过身份验证，攻击者可以绕过身份验证，导致仿冒问题	当软件的设计使恶意用户有可能嗅探网络流量，并绕过身份验证，将其重放到有问题的服务器，	
GE.EE	S	108	1021	由于渲染UI层或帧的不适当限制，攻击者可以绕过身份验证，导致仿冒问题	web应用程序不会限制或错误地限制属于另一个应用程序或领域的框架对象或UI层，这可能会导	
GE.P	S	34	347	由于加密签名验证不当，攻击者可以绕过身份验证，导致仿冒问题	该软件不会验证或错误地验证数据的加密签名。	
GE.P	S	39	322	由于无需实体身份验证的密钥交换，攻击者可以绕过身份验证，导致仿冒问题	该软件在不验证参与者身份的情况下与该参与者执行密钥交换。进行密钥交换将保持两个实体	
GE.P	S	42	326	由于加密强度不足，攻击者可以绕过身份验证，导致仿冒问题	该软件使用理论上合理的加密方案存储或传输敏感数据，但不足以满足所需的保护水平。弱加	
GE.P	S	44	330	由于随机值不足的使用，攻击者可以绕过身份验证，导致仿冒问题	该软件在依赖于不可预测数字的安全上下文中使用了足够的随机数字或值。当软件在需要不可	
GE.P	S	48	759	由于使用不带Salt的单向哈希，攻击者可以绕过身份验证，导致仿冒问题	该软件对密码等不应可逆的输入使用单向加密散列，但软件也不使用salt作为输入的一部分。这	
GE.P	S	66	643	由于XPath表达式中数据中和不当（"XPath注入"），攻击者可以绕过身份验证，导	该软件使用外部输入动态构造用于从XML数据库检索数据的XPath表达式，但它不会中和或错误	
GE.P	T	1	22	由于将路径名限制在受限制的目录中("路径遍历")，攻击者可以进行数据篡改或代码	该软件使用外部输入构建一个路径名，该路径名旨在识别位于受限父目录下方的文件或目录，	
GE.P	T	8	352	由于跨站点请求伪造（CSRF），攻击者可以进行数据篡改或代码执行，导致篡改问	Web应用程序不能或不能充分验证格式良好、有效、一致的请求是否由提交请求的用户故意提	
GE.P	T	11	425	由于直接请求（"强制浏览"），攻击者可以进行数据篡改或代码执行，导致篡改问题	Web应用程序没有对所有受限的URL、脚本或文件充分强制执行适当的授权。易受直接请求攻击	
GE.P	T	55	89	由于SQL命令中使用的特殊元素的中和不当（"SQL注入"），攻击者可以进行数据篡	该软件使用来自上游组件的外部影响输入构建SQL命令的全部或部分，但它不会中和或错误地中	
GE.P	T	52	80	由于网页中脚本相关HTML标签的中和不当中和（基本XSS），攻击者可以进行数据	该软件接收来自上游组件的输入，但它不会中和或错误地中和特殊字符，如"<"、">"和"&"，	
GE.P	I	3	200	由于向未经授权的参与者暴露敏感信息，攻击者可以读取敏感信息，导致信息泄露	产品将敏感信息暴露给未明确授权访问该信息的行为者。引入信息暴露的错误有很多种。错误	
GE.P	I	4	219	由于在Web Root下存储具有敏感数据的文件，攻击者可以读取敏感信息，导致信息	该应用程序在Web文档根目录下存储敏感数据，访问控制不足，这可能会使不受信任的各方访问	
GE.P	I	37	319	由于敏感信息的清晰文本传输，攻击者可以读取敏感信息，导致信息泄露问题	该软件在通信通道中以清晰的文本传输敏感或安全关键数据，未经授权的行为者可以嗅探。在	

图 6　ACUTIRule 应用中构建的规则(节选)

No.	Component type	Component name	Threat type	Title	Description	Associated CWE-ID	Operate
1	EE	Admin	S	由于无需实体身份验证的密钥交换，攻击者可以绕过身份	该软件在不验证参与者身份的情况下与该参与者执行密钥交换。进行密钥交换将保	322	Edit
2	EE	Admin	S	由于cookie属性不安全以及用户会话cookie可以被窃取，攻	会话cookies是服务器知道每个传入请求的当前用户身份的标识符。如果攻击者能够	311	Edit
3	EE	Admin	S	由于通过捕获-重放绕过身份验证，攻击者可以绕过身份验	当软件的设计使恶意用户有可能嗅探网络流量，并绕过身份验证，将其重放到有问	294	Edit
4	EE	Admin	S	由于渲染UI层或帧的不适当限制，攻击者可以绕过身份验	web应用程序不会限制或错误地限制属于另一个应用程序或领域的框架对象或UI层	1021	Edit
5	P	INTegrity Admin	S	由于加密签名验证不当，攻击者可以绕过身份验证，导致	该软件不会验证或错误地验证数据的加密签名。	347	Edit
6	P	INTegrity Admin	S	由于无需实体身份验证的密钥交换，攻击者可以绕过身份	该软件在不验证参与者身份的情况下与该参与者执行密钥交换。进行密钥交换将保	322	Edit
7	P	INTegrity Admin	S	由于加密强度不足，攻击者可以绕过身份验证，导致仿冒	该软件使用理论上合理的加密方案存储或传输敏感数据，但不足以满足所需的保护	326	Edit
8	P	INTegrity Admin	S	由于随机值不足的使用，攻击者可以绕过身份验证，导致	该软件在依赖于不可预测数字的安全上下文中使用了足够的随机数字或值。当软件	330	Edit
9	P	INTegrity Admin	S	由于使用不带Salt的单向哈希，攻击者可以绕过身份验证，	该软件对密码等不应可逆的输入使用单向加密散列，但软件也不使用salt作为输入的	759	Edit
10	P	INTegrity Admin	S	由于XPath表达式中数据中和不当（"XPath注入"），攻击	该软件使用外部输入动态构造用于从XML数据库检索数据的XPath表达式，但它不	643	Edit
11	P	INTegrity Admin	T	由于将路径名限制在受限制的目录中("路径遍历")，攻击者	该软件使用外部输入构建一个路径名，该路径名旨在识别位于受限父目录下方的文	22	Edit
12	P	INTegrity Admin	T	由于跨站点请求伪造（CSRF），攻击者可以进行数据篡改	Web应用程序不能或不能充分验证格式良好、有效、一致的请求是否由提交请求的	352	Edit
13	P	INTegrity Admin	T	由于直接请求（"强制浏览"），攻击者可以进行数据篡改或	Web应用程序没有对所有受限的URL、脚本或文件充分强制执行适当的授权。易受	425	Edit
14	P	INTegrity Admin	T	由于SQL命令中使用的特殊元素的中和不当（"SQL注入"）	该软件使用来自上游组件的外部影响输入构建SQL命令的全部或部分，但它不会中	89	Edit
15	P	INTegrity Admin	T	由于网页中脚本相关HTML标签的中和不当中和（基本XSS	该软件接收来自上游组件的输入，但它不会中和或错误地中和特殊字符，如"<"、"	80	Edit
16	P	INTegrity Admin	I	由于向未经授权的参与者暴露敏感信息，攻击者可以读取	产品将敏感信息暴露给未明确授权访问该信息的行为者。引入信息暴露的错误有很	200	Edit
17	P	INTegrity Admin	I	由于在Web Root下存储具有敏感数据的文件，攻击者可以	该应用程序在Web文档根目录下存储敏感数据，访问控制不足，这可能会使不受信	219	Edit
18	P	INTegrity Admin	I	由于敏感信息的清晰文本传输，攻击者可以读取敏感信息	该软件在通信通道中以清晰的文本传输敏感或安全关键数据，未经授权的行为者可	319	Edit

图 7　ACUTIRule 应用中识别的威胁列表(节选)

Id	Title	Category	Description
16	An adversary can create a fake website and launch phishing attacks	Spoofing	Phishing is attempted to obtain sensitive information such as
28	An adversary can create a fake website and launch phishing attacks	Spoofing	Phishing is attempted to obtain sensitive information such as
38	An adversary can create a fake website and launch phishing attacks	Spoofing	Phishing is attempted to obtain sensitive information such as
48	An adversary can create a fake website and launch phishing attacks	Spoofing	Phishing is attempted to obtain sensitive information such as
18	An adversary can deface the target web application by injecting malicious code or	Tamperin	Website defacement is an attack on a website where the attac
6	An adversary can gain access to certain pages or the site as a whole.	Informatic	Robots.txt is often found in your site's root directory and exist
20	An adversary can gain access to sensitive data by performing SQL injection throug	Tamperin	SQL injection is an attack in which malicious code is inserted i
30	An adversary can gain access to sensitive data by performing SQL injection throug	Tamperin	SQL injection is an attack in which malicious code is inserted i
40	An adversary can gain access to sensitive data by performing SQL injection throug	Tamperin	SQL injection is an attack in which malicious code is inserted i
50	An adversary can gain access to sensitive data by performing SQL injection throug	Tamperin	SQL injection is an attack in which malicious code is inserted i
7	An adversary can gain access to sensitive data by sniffing traffic to Web Applicatio	Informatic	An adversary may conduct man in the middle attack and dow
21	An adversary can gain access to sensitive data stored in Web App's config files	Tamperin	An adversary can gain access to the config files. and if sensitiv
31	An adversary can gain access to sensitive data stored in Web App's config files	Tamperin	An adversary can gain access to the config files. and if sensitiv
41	An adversary can gain access to sensitive data stored in Web App's config files	Tamperin	An adversary can gain access to the config files. and if sensitiv
51	An adversary can gain access to sensitive data stored in Web App's config files	Tamperin	An adversary can gain access to the config files. and if sensitiv
8	An adversary can gain access to sensitive information through error messages	Informatic	An adversary can gain access to sensitive data such as the foll
24	An adversary can gain access to sensitive information through error messages	Informatic	An adversary can gain access to sensitive data such as the foll
34	An adversary can gain access to sensitive information through error messages	Informatic	An adversary can gain access to sensitive data such as the foll

图 8　微软 TMT 原始列表展示(节选)

本文 ACUTIRule 应用识别的威胁列表和微软 TMT 识别的威胁列表在数量上的分布情况如表 16 所示。合并前的数量下的威胁列表以威胁和组件的成对关系为主键，即以(威胁，组件)的形式进行展示。将威胁列表切换至以威胁为主键，即以(威胁，[组件])的形式展示，对同一威胁关联的不同组件进行合并便可以得到纯威胁的数量。对 ACUTIRule 应用识别的威胁列表进行合并，合并后的威胁数量为 191 条。将微软 TMT 识别的威胁进行合并后，微软 TMT 威胁列表实际涉及的威胁仅为 24 条。从表 16 中可以看到，合并前与合并后，ACUTIRule 应用识别出的威胁的数量都大幅领先于微软 TMT 识别出的威胁的数量。

表 16 识别的威胁数量对比

方法	合并前							合并后						
	S	T	R	I	D	E	合计	S	T	R	I	D	E	合计
ACUTIRule 应用	**21**	**97**	**27**	**158**	**73**	**31**	**407**	**14**	**47**	**13**	**73**	**22**	**22**	**191**
微软 TMT	19	10	4	16	1	1	51	10	4	1	7	1	1	24

通过对微软 TMT 的 24 个威胁进行逐一分析，发现其相同、类似或拆分后的威胁均可在 ACUTIRule 应用识别的 191 个威胁中被找到。例如微软 TMT 的威胁列表中涉及的 Spoofing 威胁仅包含由于创建假网站(fake website)导致的攻击这条粗粒度的威胁，而 ACUTIRule 应用的威胁列表中包含了多条 Spoofing 威胁，涵盖了造成威胁影响的潜在威胁行为的各个方面。攻击者绕过认证进行欺骗攻击，可能是由多种攻击行为或原因造成的。例如“无需实体身份验证的密钥交换”、“cookie 属性不安全以及用户会话 cookie 可以被窃取”、“通过捕获-重放绕过身份验证”、“渲染 UI 层或帧的不适当限制”、“加密签名验证不当”、“随机值不足的使用”、“使用不带 Salt 的单向哈希”、“XPath 表达式中数据中和不当(‘XPath 注入’)”等。从威胁的覆盖范围而言，ACUTIRule 应用的覆盖范围相对更广。

除了威胁数量和覆盖范围外，将图 7 的内容与图 8 进行对比可以看到，ACUTIRule 应用还附带了关联的 CWE 条目。如果阅读威胁列表的开发人员存在一些疑惑，除了通过标题和描述中的文字内容进行理解，还可以通过超链接的方式跳转到对应的网页去获取更多细节的信息，包括案例、消减措施、关联的 CAPEC 和 CVE 链接等。有利于帮助后续使用该威胁列表的人员更好地理解威胁，以达到消减威胁的目的。

由于微软 TMT 中的威胁数据量小、威胁粒度大，导致识别的威胁少；其人工构建的规则不够全面，无法确保识别的威胁的全面性和准确性。本文 ACUTIRule 应用中基于自动构建的全面的规则库，对 Web 安全领域常见场景的威胁基本都有覆盖。我们对多个数据源的威胁数据进行了收集以及精细处理，对威胁描述的核心内容采用统一格式进行描述，将威胁行为与造成的影响进行关联，从业人员能够快速准确地掌握威胁的原因、行为、类别等信息。同时我们提供了详细的威胁描述以及相关的来源编号和链接，方便开发者对其进行详尽了解。增强了威胁的可理解性，便于对威胁进行分析、缓解以及对接后续安全测试生成。通过 iNTegrity 实例展示了 ACUTIRule 应用结果，表明本文提出的 ACUTIRule 的应用能够有效支持从业人员进行威胁识别。

7 讨 论

本节讨论了本研究工作的重要性、相关考虑和局限性。

7.1 重要性

由于知识产权的限制以及自动化程度、数据支持和分析质量的不足，现有的工具尚未被业界广泛使用。目前，业界主要依赖于人工威胁建模和分析，缺少自动化威胁建模技术，识别结果作为需求提供给测试人员手动设计测试用例来验证安全问题，这需要很高的专业知识。如引言所述，自动化威胁建模在工业中是迫切需要的。学术界[7-9]指出，目前的相关研究还不成熟，并呼吁进行这方面的研究。我们提出了一种新的方法来提高威胁建模的自动化程度和分析质量，解决痛点，满足学术界和工业界的需求，支持设计阶段的威胁分析和处理，以及安全实践的左移。

目前，已有的研究大多采用微软公司的 STRIDE 威胁建模方法，通过人工构建不完整的规则库来进行威胁识别。作为构建和更新规则库的第一个自动化方法，我们的工作能够支持基于全面规则库的自动威胁识别。通过输入特定于场景的 DFD，可以根

据规则库自动生成威胁列表,这使得开发人员能够在开发的早期阶段识别潜在的威胁,为威胁生成测试[13],指导后续的开发和测试,并更好地将安全问题嵌入到软件开发活动中。

7.2　相关考虑

本文提出的方法具有通用性,并以 Web 安全领域为典型应用领域构建规则库数据。由于难以从威胁数据中挖掘通用规则或直接提取组件以及组件与威胁之间的关系,本文中的规则自动构建和更新任务具有挑战性。Gao 等人[80]和 Kadhim 等人[69]分别通过建立知识图谱或者基于知识库的语义相似度匹配方法实现其他领域规则库的自动构建。因为威胁和软件组件的快速更新迭代,而通过构建知识图谱的方法需要先找出威胁内容与各个实体间的关系,这就需要专家频繁地对知识图谱间关系进行判定和填充。所以本文以成熟的类型规则库作为语料库进行语义相似度匹配的方式更符合威胁识别规则模型自动构建和更新的背景需求。但是这种方法存在一定的局限性,即只能自动构建已有组件的规则,完全新的组件相关的规则无法直接自动构建。当出现全新的组件内容时需要有专家填充与之关联的少数规则,后续即可自动添加。本文数据来源于多方数据源,包含了 Web 领域常见的组件和威胁数据,Web 领域的新威胁通常可以匹配到同组件关联的规则。此外,三元组表达式包含了攻击行为、后果等信息,即使威胁文本不完全相同,也可匹配到对应规则并提取该威胁关联的组件。规则更新入库之前会经过专家审核,如果出现相似度低的情况,可能是由于输入的威胁属于其他领域特有或罕见的组件关联的威胁。这种情况需要人工将新组件关联威胁,根据本方法制定规则并添加入库,后续可自动构建和更新。

STRIDE 是目前软件行业应用的主流方法。与其他方法相比,DFD 模型覆盖了大多数其他模型的各个方面,能够以相对较低的技术门槛支持更全面的威胁分析,因此在软件行业中更受欢迎,与需求、设计和实现活动的结合更紧密。由于安全问题经常出现在数据流中,所以数据流模型成为最理想的威胁模型[5]。然而,由于数据和规则不完整,自动化和准确性不够,软件行业仍然缺乏自动威胁建模技术。虽然微软的 TMT 工具有库支持,但由于它是人工构建的,无法更新,威胁的数量有限,规则也不全面,并未被业界所广泛使用。到目前为止,在这个前沿领域还没有相关的工作,也没有大型语言模型(LLMs)的应用。考虑到这种规则制定任务中分析所需要的专业知识和经验,本文提出的方法是一种从零开始的可行方案。而且由于相关数据的匮乏,目前的 LLMs 还没有准备好解决这个问题。

如引言和背景部分所述,业界现有的大多数工具不支持自动识别,没有数据支持或数据集小,规则库不全面,不支持规则自动构建和更新。CNNVD 数据更全面,数据量充足,粒度适当,描述了类别和影响,信息更全面。以 CWE 和 CVE 作为数据集进行验证,计算准确率表明 TextCPR 方法在不同粒度的威胁描述数据集上仍能表现良好,也表明本文在数据处理阶段去除 CNNVD 数据中无用信息的有效性。CNNVD、CWE、OWASP Top Ten 的数据源中,仅 CNNVD 需要处理,其他两个平台的数据仅用于更新,不包含产品信息等不必要的内容。其只需爬取或下载并直接输入,通过 XML 分析的威胁内容等属性将组成数据。其他平台的数据可以分成 CNNVDTC 类,用于匹配和识别 STRIDE 类。本文根据类别的定义匹配形成 CNNVDTC 与 STRIDE 类别的映射表,如代码问题的定义描述了攻击行为可以通过代码设计导致拒绝服务等问题。

虽然威胁数据可能涉及某些网络威胁数据,但威胁建模在开发初期就被视为实现安全的重要软件实践,无疑也是安全领域的一个重要课题。所提出的方法可以支持广泛的软件从业者,从威胁建模者、安全分析师和设计师,到软件开发人员和测试人员。然而,威胁建模的自动化水平仍然处于较低水平,这一领域的前沿研究进展在国内尚属空白。

在这项前沿工作中,首次实现了 STRIDE 威胁分类,结合 TextCNN 模型这种可用且适用效果理想的模型对威胁文本进行基础类别分类,然后结合产生式规则匹配生成 STRIDE 威胁类别。本研究中,文本分类模型的选择目标是选择可用的、适用于实际问题需求的高效模型,而非无止境地寻找最新、最准、最快、指标最高的模型。这里重点考虑了 TextCNN 模型的神经网络参数少、训练快、结构简单的模型,该模型具有灵活、可理解、高效等特点[72]。同时,该模型的结构公开,当数据集发生变化时,模型易于调整。另外,它对软硬件环境要求较低,更容易应用到开发实际场景。基于以上全面考虑,我们将 TextCNN 模型作为本研究的一种可用和轻量的文本分类模型。Qu 等人[68]基于 CNN 实现了漏洞的自动分类。从分类结果来看,使用 CNN 模型可以有效地对威胁数据进行基础类别分类,效

果好。此外，本文实验中还包括了对比 TextCNN 分类模型与传统的 SVM 分类模型在威胁分类任务下的效果差异，实验结果表明 TextCNN 模型威胁分类部分在一定程度上提高了 TextCPR 分类方法的整体性能，可以有效地用于对威胁描述文本 CNNVDTC 进行分类。

需要强调的是，本文的主要贡献在于整体首创性的威胁建模自动化方法，整体方法首次有效地实现了 STRIDE 威胁自动分类以及威胁识别规则自动构建和更新。对于单个模型如 TextCNN 模型，可以根据人工智能技术的发展，与威胁分析的领域及应用场景，后续根据综合评估替换为更先进的模型。TextCPR 方法中，TextCNN 的核心作用是通过 CNNVD 威胁分类为产生式规则匹配提供 CNNVDTC 类别输入，其相比传统方法在威胁描述变体和多粒度数据上具有更强的泛化能力。TextCPR 方法的核心创新在于有效地将 TextCNN 分类结果与定义的映射规则相结合。其针对 STRIDE 标注数据不足的问题，通过预定义的映射规则，利用 CNNVDTC 的丰富样本训练模型，并结合产生式规则匹配，实现 STRIDE 威胁分类。这种协同方法提升了小样本场景下的模型泛化能力。当威胁类别体系扩展时，该方法仅需扩展映射规则而无需重新训练模型，但需人工维护规则。此外，TextCNN 对 CNNVD 威胁分类的正确性直接影响后续 STRIDE 类别的输出，可能存在误差传播风险。

本自动化方法是在人工辅助准备数据的基础上实现的自动化，且本文已提供了常见数据集。为了实现目标任务，手工完成的任务如整理相对全面的数据以制定规则库是一次性行为，其成果可支持自动构建规则。与现有威胁建模工具对比，本方法在自动化程度、数据与规则的完整性等方面存在优势，并且本方法支持 STRIDE 威胁分类和规则构建，支持自动威胁识别。与人工方式相比，所提出的方法实现了自动化构建和更新规则。方法中数据获取为自动，由于 CNNVD 数据有产品信息等不必要的内容，数据处理中数据项合并这一步需要人工处理，然后自动去除无用信息和欠采样处理。自动构建方法中的主体部分如威胁分类、三元组表达式提取、规则匹配、规则生成、规则库更新都是自动化进行。其中，需要人工辅助的方面包括：自动威胁分类需要基于数据类别标注进行模型训练，交互规则提取需要基于整理的组件关系表，以及规则更新入库时经过专家审核。对于人工辅助环节，后续可利用 LLMs 智能解析非结构化数据，提取关键信息并转化为结构化/半结构化数据，从而提升数据处理效率，降低人工成本。此外，可基于迁移学习技术，利用 CNNVD 已标注数据预训练模型，通过特征迁移和模型微调策略适配跨平台标注任务，降低新数据源的标注需求。

两个系统文献综述都得出了类似的观察结果[8]，即“[…]现有技术缺乏产品质量保证。此外，这些技术缺乏成熟度、有效性和工具支持”[9]和“大多数威胁建模工作仍然是手工完成的，并且对其验证的保证有限”[7]。目前威胁建模领域的大多数论文都是通过理论例子或实证案例研究来验证的，这些研究大多用于验证威胁模型，或者没有进行任何验证[7,9]。我们在现有条件下根据验证需求，在最大限度的努力下通过对比实验对提出的方法进行验证，这在威胁建模相关研究的验证中是不多见的。我们将分类方法与基准方法进行对比，将自动构建的规则与人工构建的经过专家审核的规则进行对比，验证了方法的有效性。

7.3 局限性

本文所提出的构建规则的方法也存在一定的局限性，即只能自动构建现有组件的规则，不能直接自动构建与全新组件相关的规则。当出现新的组件内容时，专家需要填充少量与之相关的规则。未来工作将探索使用少量样本学习（Few-shot Learning）、领域自适应（Domain Adaptation）和整合外部知识源来扩展所提出的 ACUTIRule 方法，以处理新组件和动态威胁。首先通过异常检测（如 Isolation Forest）识别系统中新组件。随后基于少样本学习技术，采用元学习框架（如 MAML 或 Prototypical Networks），通过多任务学习方式在已有组件规则集上训练元模型，使模型获得快速适应新组件的能力。当新组件出现时，仅需少量样本即可通过模型生成初始规则。然后基于领域自适应技术，将已有组件的规则知识迁移到新组件。基于特征级自适应技术（如 DANN 或 MMD）对齐新旧组件的特征分布，构建共享威胁特征空间。将组件的行为映射到共享特征空间，使其与已知相似威胁行为的特征距离最小化。通过图神经网络（GNN）建模组件关系，若新组件与已知组件存在依赖或功能重叠，则自动迁移威胁识别规则。最后通过实时监控新组件的威胁数据并整合外部知识源补充信息[81]，利用在线学习（如 Bandit 算法）对规则进行细粒度调整，持续优化规则质量。通过知识沉淀技术将验证后的规则反

馈至元学习模型,形成闭环优化。新组件通过置信度验证后,将自动触发组件关系图的拓扑扩展,从而实现规则库的自主演进,逐步降低对人工维护的依赖。本文提出的自动识别和分类 STRIDE 威胁的方法有助于支持威胁识别规则的自动构建和更新。这种方法可以用于在安全实践中自动识别威胁,尤其是在软件开发的早期阶段。现阶段的工作仍然仅限于基于 STRIDE 这一事实上的主流威胁建模方法,即基于 STRIDE 类别的分类和基于 STRIDE 的威胁识别规则的构建,并且对于描述过于冗长的威胁文本,处理效果可能有限。当前探索阶段,我们采用 EasyEnsemble 优化样本分布后,重点验证模型在全局分布上的可行性,可能无法完全反映少数类性能,后续将开展细粒度类别分析及鲁棒性验证。

在规则库定期更新场景下,规则库规模可能呈现阶梯式增长。在规则更新维护中,随着核心攻击模式的覆盖完善,可根据需求比如规则库内存余量,选择不更新具有相同攻击特征的重复威胁对应的规则,优先更新具有新特征且威胁等级较高的规则,确保规则库全面性的同时保持规模受控。为提高规则库的可扩展性,采用分层存储与增量编译机制,通过规则匹配引擎与规则库管理的解耦设计,实现规则字段(如威胁描述、类别等)的按需加载与动态读取。引入基于组件类型的层级索引优化,确保匹配效率,同时通过模块化存储设计支持灵活扩展。此外,更新机制中设定了规则更新入库时经过专家审核,以提供更多保障,但可能需要投入专家人力和时间等成本。对于高风险核心规则,可根据更新频率定期进行全量审核。对于常规规则更新,可采用低频抽样审核。同时可引入自动化工具(如规则优先级评估、规则冲突检测)辅助人工审核,以优化成本效率。

对这项工作的有效性可能存在一些威胁。1)内部有效性:在威胁分类的对比实验中,本文选取了传统 SVM 方法作为对比对象,SVM 分类器对数据处理方面的要求更高。实验中采用的预测测试数据集相同,可能会影响 SVM 分类方法的有效性。实验使用从 CVE 平台爬取的威胁数据作为测试数据集。然而,CVE 平台收纳的威胁数据可能会存在内容相似的情况,这可能会影响实验对比效果。由于威胁识别规则的制定暂未有统一的标准,因此在本文自动和人工构建规则的方法的对比实验中,对产出的规则的正确性判断依赖于专家经验,可能存在人为主观性。后续可通过引入更多量化指标(如规则覆盖率、冲突率、历史数据验证准确率等),并结合多专家交叉验证机制,以降低个体主观偏差的影响。2)外部有效性:CNNVD、CWE 等平台的威胁数据多来自知名机构。然而,目前还没有记录所有潜在威胁的数据平台。尚未包括的威胁数据可能会影响结论的普遍性。

8　总结与展望

为了解决目前威胁建模质量较低、自动化程度不足的问题,本文基于 STRIDE 方法提出了完整的威胁识别规则模型,提供了完善的规则库数据。本文通过 NLP 技术提出了自动构建规则库的方法,以及提出了 STRIDE 威胁自动分类方法。通过 TextCNN 分类模型确定威胁内容的漏洞基础类别,然后通过产生式规则匹配获取对应其漏洞基础类别的威胁 STRIDE 类别,解决了 STRIDE 类型标注数据不足的挑战。通过所提出的分类方法获取威胁的 STRIDE 类别,结合获取到的核心动词短语组生成关系三元组表达式。通过文本相似度匹配获得威胁的组件,构建出类型规则。基于重复的类型规则以及数据流交互的组件关系表,构建出交互规则。最后将类型规则和交互规则整合为完整的威胁识别规则。最后本文设计了规则库自动更新机制,以确保规则库的时效性。本文通过实验验证了方法的有效性。一方面,以 CNNVD、CVE 和 CWE 等主流漏洞平台的威胁数据为样本,通过对比实验评估了所提出的方法在 STRIDE 分类任务上的有效性。结果表明该方法提高了分类的性能和自动化程度,精度达到 92.5%。另一方面,对威胁识别规则库的自动构建方法进行了对比实验。自动构建规则的准确率达到 89.5%。与人工构建方法相比,该方法提高了规则构建的自动化程度和效率,误差在 10%以内。评估结果表明本文所提出的方法的有效性。未来工作计划探索研究如何自动更新组件以及新组件相关的规则,以进一步提升自动化构建和更新规则库的效果,提高规则库的可扩展性,以及在更广泛的数据集上评估所提出的方法。

参　考　文　献

[1] McKee D, Crean M. 2024 SonicWall Mid-Year Cyber Threat Report. SonicWall, 2024. https://www.sonicwall.com/threat-report (https://siliconangle.com/2024/07/25/sonicwalls-cyber-threat-report-reveals-alarming-cybersecurity-trends-cubeconversations/)

[2] Nazah S, Huda S, Abawajy J, et al. Evolution of dark web threat analysis and detection: a systematic approach. IEEEAccess, 2020, 8: 171796-171819

[3] Myagmar S, Lee AJ, Yurcik W. Threat modeling as a basis for security requirements//Proceedings of theSymposium on Requirements Engineering for Information Security (SREIS 2005). LasVegas,USA, 2005: 1-8

[4] Howard M, Lipner S. Inside the windows security push. IEEE Security and Privacy, 2003, 1(1): 57-61

[5] Shostack, A. Threat modeling: designing for security. John Wiley & Sons, 2014

[6] Kohnfelder L, Garg P. The threats to our products. Microsoft Interface, April 1, 1999. http://blogs. msdn. com/sdl/attachment/9887486. ashx (http://www. cnetsec. com/article/32430. html)

[7] Xiong W, Robert L. Threat modeling- a systematic literature review. Computers & Security, 2019, 84:53-69

[8] Yskout K, Heyman T, Landuyt DV, et al. Threat modeling: from infancy to maturity//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER 2020). New Ideas and Emerging Results, Seoul, Republic of Korea, 2020: 9-12

[9] Tuma K, Calikli G, Scandariato R. Threat analysis of software systems: a systematic literature review. Journal of Systems & Software, 2018, 144: 275-294

[10] Al-Aswadi FN, Chan HY, Gan KH. Automatic ontology construction from text: a review from shallow to deep learning trend. Artificial Intelligence Review, 2020, 53(6):3901-3928

[11] Lipner S. The trustworthy computing security development lifecycle//Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC 2004). Tucson, USA, 2004: 2-13

[12] Williams I, Yuan X. Evaluating the effectiveness of Microsoft threat modeling tool//Proceedings of the 2015 Information Security Curriculum Development Conference (InfoSecCD 2015). Kennesaw, USA, 2015:1-6

[13] Fu CL, Zhang H, Li FL, and Kuang HY. Threat model-based security test case generation framework and tool. Journal of Software, 2023, 34(9): 4573-4603 (in Chinese)
(付昌兰, 张贺, 李凤龙, 匡宏宇. 一种基于威胁模型的安全测试用例生成框架和工具. 软件学报, 2023, 34(9): 4573-4603)

[14] Sasnick O, Rosenstatter T, Schafer C, et al. STRIDE-based methodologies for threat modeling of industrial control systems: a review//Proceedings of the 2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS 2024). Chongqing, China, 2024

[15] Lohmann PA, Albuquerque C, Machado R. Systematic literature review of threat modeling concepts//Proceedings of the 9th International Conference on Information SystemsSecurity and Privacy (ICISSP2023). Lisbon, Portugal, 2023: 163-173

[16] Hanvey S, Catano N. Identifying transitivity threats in social networks//Proceedings of the 1st IEEE/ACMInternational Workshop on TEchnical and LEgal aspects of data pRIvacy and SEcurity (TELERISE2015). Florence, Italy, 2015: 14-19

[17] Zou Q, Singhal A, Sun X, et al. Automatic recognition of advanced persistent threat tactics for enterprise security//Proceedings of the Sixth International Workshop on Security and Privacy Analytics (IWSPA 2020). New Orleans, USA, 2020: 43-52

[18] Dev J, Akhuseyinoglu N B, Kayas G, et al. Building guardrails in AI systems with threat modeling. Digital Government: Research and Practice, 2025, 6(1). DOI:10. 1145/3674845

[19] Liu F, Wen Y, Zhang D, et al. Log2vec: a heterogeneous graph embedding based approach for detecting cyber threats within enterprise//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS 2019). London, UK, 2019: 1777-1794

[20] Alidoost M, Bahrak B, Kargahi M, et al. Detecting new generations of threats using attribute-based attack graphs. IET Information Security, 2019, 13(4): 293-303

[21] Rouland Q, Hamid B, Jaskolka J. Specification, detection, and treatment of STRIDE threats for software components: modeling, formal methods, and tool support. Journal of systems architecture, 2021, 117(5): 102073

[22] Torkura KA, Sukmana MIH, Cheng F, et al. Continuous auditing & threat detection in multi-cloud infrastructure. Computers & Security, 2020, 102(1/2): 102124

[23] Joshi C, Aliaga JR, Insua DR. Insider threat modeling: an adversarial risk analysis approach. IEEE Transactions on Information Forensics and Security, 2021, 16:1131-1142

[24] Vanlanduyt D, Joosen W. A descriptive study of assumptions in STRIDE security threat modeling. Software and systems modeling, 2022, 21(6): 2311-2328

[25] Srikumar K, Kashish K, Eggers K, et al. STRIPED: a threat analysis method for IoT systems//Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 2022). Vienna, Austria, 2022: 96:1-96:6

[26] Wilhjelm C, Younis AA. A threat analysis methodology for security requirements elicitation in machine learning based systems//Proceedings of the20th IEEE International Conference on Software Quality, Reliability and Security Companion (QRS Companion 2020). Macao, China, 2020: 426-433

[27] Hacks S, Katsikeas S, Ling E, et al. Towards a systematic method for developing meta attack language instances//Proceedings of theEnterprise, Business-Process and Information Systems Modeling- 23rd International Conference, and 27th International Conference, (BPMDS 2022, EMMSAD 2022). Leuven, Belgium, 2022, 450: 139-154

[28] Johnson P, Lagerström, Robert, Ekstedt M. A meta language for threat modeling and attack simulations//Proceed-

ings of the 13th International Conference on Availability, Reliability and Security, (ARES 2018). Hamburg, Germany, 2018:1-8

[29] Xiong W, Legrand E, Berg O, et al. Cyber security threat modeling based on the MITRE enterprise ATT&CK matrix. Software and Systems Modeling, 2021, 21(1): 157-177

[30] Katsikeas S, Buhaiu A, Ekstedt M, et al. Development and validation of coreLang: a threat modeling language for the ICT domain. Computers & Security, 2024, 146: 104057

[31] Katsikeas S, Ling ER, Johnsson P, et al. Empirical evaluation of a threat modeling language as a cybersecurity assessment tool. Computers& Security, 2024, 140(000):16

[32] Rodrigues AADO, Villela MLB, Feitosa EL. PTMOL: a privacy threat modeling language for online social networks// Proceedings of the XXIII Brazilian Symposium on Human Factors in Computing Systems (IHC 2024). Brasília, Brazil, 2024: 46,1-14

[33] Mahmood S, Nguyen HN, Shaikh SA. Systematic threat assessment and security testing of automotive over-the-air OTA updates. Vehicular Communications, 2022, 35: 100468

[34] Marksteiner S, Ramler R, Sochor H. Integrating threat modeling and automated test case generation into industrialized software security testing//Proceedings of the Third Central European Cybersecurity Conference (CECC 2019). Munich, Germany, 2019: 25:1-25:3

[35] Assen J, Sharif J, Feng C, Bovet G, Stiller B. Asset-driven threat modeling for AI-based systems. Computing Research Repository (CoRR), 2024, abs/2403.06512

[36] Jamil AM, Khan S, Lee JK, et al. Towards automated threat modeling of cyber-physical systems//Proceedings of the 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM). Gambang, Malaysia, 2021:118

[37] Khan R, Mclaughlin K, Laverty D, et al. STRIDE-based threat modeling for cyber-physical systems//Proceedings of the 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe 2017). Torino, Italy, 2017: 1-6

[38] Khalil SM, Bahsi H, Dola HO, et al. Threat modeling of cyber-physical systems-a case study of a microgrid system. Computers & Security, 2023: 124

[39] Li K, Rashid A, Roudaut A. Vision: security-usability threat modeling for industrial control systems//Proceedings of the European Symposium on Usable Security 2021 (EuroUSEC 2021). Karlsruhe, Germany, 2021: 83-88

[40] Abbas SG, Vaccari I, Hussain F, et al. Identifying and mitigating phishing attack threats in IoT use cases using a threat modelling approach. Sensors, 2021, 21(14): 4816

[41] Al Asif MR, Fida Hasany K, Zahidul Islamz M, et al. STRIDE-based cyber security threat modeling for IoT-enabled precision agriculture systems. Computing Research Repository (CoRR), 2022, abs/2201.09493

[42] Damianou A, Khan MA, Angelopoulos CM, et al. Threat modelling of IoT systems using distributed ledger technologies and IOTA//Proceedings of the 17th International Conference on Distributed Computing in Sensor Systems (DCOSS 2021). Pafos, Cyprus, 2021: 404-413

[43] Kazim M, Evans D. Threat modeling for services in cloud// Proceedings of the 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE 2016). Oxford, UK, 2016: 66-72

[44] Pell R, Moschoyiannis S, Panaousis E. Multi-stage threat modelling and security monitoring in 5GCN. Computing Research Repository (CoRR), 2021, abs/2108.11207

[45] Pell R, Moschoyiannis S, Panaousis E, et al. Towards dynamic threat modelling in 5G core networks based on MITRE ATT&CK. Computing Research Repository (CoRR), 2021, abs/2108.11206

[46] Sattar D, Vasoukolaei AH, Crysdale P, et al. A STRIDE threat model for 5G core slicing//Proceedings of the 4th IEEE 5G World Forum (5GWF 2021). Montreal, Canada, 2021: 247-252

[47] Yan ZP, Gu CL, Huang HJ: Analysis for threat models and improvement scheme of 5G AKA protocol based on petrinet//Proceedings of the 21st International Conference on Communication Technology (ICCT 2021). Tianjin, China, 2021: 11-17

[48] Al-Hadhrami N, Collinson M, Oren N. Security analysis using subjective attack trees//Proceedings of the Innovative Security Solutions for Information Technology and Communications-13th International Conference (SecITC 2020). Bucharest, Romania, 2020, 12596:288-301

[49] Dillon-Merrill RL, Parnell GS, Buckshaw DL. Logic trees: fault, success, attack, event, probability, and decision trees. Wiley Handbook of Science and Technology for Homeland Security, 2009

[50] Sindre G, Opdahl AL. Eliciting security requirements with misuse cases. Requirements Engineering, 2005, 10(1): 34-44

[51] Abe T, Hayashi S, Saeki M. Modeling security threat patterns to derive negative scenarios//Proceedings of the 20th Asia-Pacific Software Engineering Conference (APSEC 2013). Bangkok, Thailand, 2013, 1: 58-66

[52] Robles-González A, Parra-Arnau J, Jordi Forné. A LINDDUN-based framework for privacy threat analysis on identification and authentication processes. Computers & Security, 2020, 94: 101755

[53] Gangavarapu A, Daw E, Singh A, et al. Target privacy threat modeling for COVID-19 exposure notification systems. Computing Research Repository (CoRR), 2020, abs/2009.13300

[54] Pape N, Mansour C. PASTA threat modeling for vehicular networks security//Proceedings of the 7th International Con-

ference on Information and Computer Technologies (ICICT 2024). Honolulu, USA, 2024: 474-478

[55] Saitta P, Larcom B, Eddington M. Trike v. 1 methodology document [Draft]. 2005

[56] Caralli RA, Stevens JF, Young LR, et al. Introducing OCTAVE allegro: improving the information security risk assessment process. 5 Eglin Street, Hanscom AFB: Software Engineering Institute, technical report: CMU/SEI-2007-TR-012 ESC-TR-2007-012, 2007

[57] Fl LH, Borgaonkar R, Tondel IA, et al. Tool-assisted threat modeling for smart grid cyber security//Proceedings of the International Conference on Cyber Situational Awareness, Data Analytics and Assessment. Dublin, Ireland, 2021: 1-8

[58] Casola V, Benedictis AD, Mazzocca C, et al. Toward automated threat modeling of edge computing systems//Proceedings of the IEEE International Conference on Cyber Security and Resilience. Rhodes, Greece, 2021: 135-140

[59] Silva MD, Puys M, Thevenon PH, Mocanu S, Nkawa N. Automated ICS template for STRIDE Microsoft threat modeling tool//Proceedings of the 18th International Conference on Availability, Reliability and Security. Benevento, Italy, 2023: 118:1-118:7

[60] Marksteiner S, Vallant H, Nahrgang K. Cyber security requirements engineering for low-voltage distribution smart grid architectures using threat modeling. Journal of Information Security and Applications, 2019, 49:102389

[61] Casola V, Benedictis AD, Rak M, et al. Toward the automation of threat modeling and risk assessment in IoT systems- ScienceDirect. Internet of Things, 2019, 7: 100056-100056

[62] Välja M, Heiding F, Franke U, et al. Automating threat modeling using an ontology framework. Cybersecurity, 2020, 3(1): 20

[63] Li YL. An approach towards standardising vulnerability categories. Standardising Vulnerabililty Categories, 2008, 371-382

[64] Chen Q, Bao L, Li L, et al. Categorizing and predicting invalid vulnerabilities on common vulnerabilities and exposures//Proceedings of the 25th Asia-Pacific Software Engineering Conference. Nara, Japan, 2018: 345-354

[65] Ayoade G, Chandra S, Khan L, et al. Automated threat report classification over multi-source data.//4th IEEE International Conference on Collaboration and Internet Computing. Philadelphia, PA, USA, 2018: 236-245

[66] Islam C, Babar MA, Croft R, et al. SmartValidator: a framework for automatic identification and classification of cyber threat data. Journal of Network and Computer Applications, 2022, 202: 103370

[67] Liao XF, Wang YJ, Fan XB, et al. Security vulnerability classification based on LDA topic model. Journal of Tsinghua University (Science and Technology), 2012, 052(010): 1351-1355 (in Chinese)
(廖晓锋, 王永吉, 范修斌等. 基于 LDA 主题模型的安全漏洞分类. 清华大学学报(自然科学版), 2012, 052(010): 1351-1355)

[68] Qu LY, Jia YZ, Hao YL. Automatic classification of vulnerabilities based on CNN and text semantics. Transactions of Beijing Institute of Technology. 2019, 39(7): 738-742 (in Chinese)
(曲泷玉, 贾依真, 郝永乐. 结合 CNN 和文本语义的漏洞自动分类方法. 北京理工大学学报, 2019, 39(7): 738-742)

[69] Kadhim MA, Alam MA, Kaur H. A multi-intelligent agent system for automatic construction of rule-based expert system. International Journal of Intelligent Systems Technologies & Applications, 2016, 8(9): 62-68

[70] Belabed I, Alaoui M T, Miloud J E, et al. Association rules algorithms for data mining process based on multi agent system//Proceedings of the Machine Learning for Networking-Second IFIP TC 6 International Conference (MLN 2019). Paris, France, 2019, 12081: 431-443

[71] Liu TY. EasyEnsemble and feature selection for imbalance data sets//Proceedings of the International Joint Conferences on Bioinformatics, Systems Biology and Intelligent Computing. Shanghai, China, 2009: 517-520

[72] Kim, Y. Convolutional neural networks for sentence classification//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014: 1746-1751. 2014

[73] Cai J, Li J, Li W, et al. Deeplearning model used in text classification//Proceedings of the 15th International Computer Conference on Wavelet Active Media Technology and Information Processing. Chengdu, China, 2018

[74] Devlin J, Chang MW, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, USA, 2019, 1: 4171-4186

[75] VaswaniA,ShazeerN, ParmarN, UszkoreitJ, JonesL, GomezAN, KaiserL, PolosukhinI. Attention is all you need//Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017. Long Beach, USA, 2017: 5998-6008

[76] Jin R, Nan J. Combining sources from CVE and CNNVD: data analysis in information security vulnerabilities. Journal of Physics: Conference Series, 2021, 1800(1): 012004

[77] Widyassari AP, Rustad S, Shidik GF, et al. Review of automatic text summarization techniques & methods. Journal of King Saud University- Computer and Information Sciences, 2022, 34(4): 1029-1046

[78] Li S, Rui X. Syntactic analysis of Chinese existential sen-

tences "LP + VP + NP" under the framework of minimalist program. Modern Linguistics, 2023, 11(2): 622-628 (in Chinese)
(李水,芮旭东. 最简方案框架下"LP + VP + NP"类存现句的句法结构分析. 现代语言学, 2023, 11(2): 622-628)

[79] Steinberger J, Jezek K. Using latent semantic analysis in text summarization and summary evaluation//Proceedings of the 7th International Conference on Information Systems Implementation and Modelling. Roznov pod Radhostem, Czech Republic, 2004: 93-100

[80] Gao P, Liu X, Choi E, et al. A system for automated open-source threat intelligence gathering and management//Proceedings of the 2021 International Conference on Management of Data. Virtual, China, 2021: 2716-2720

[81] Ma BQ, Zhou YH, Wang ZY, Tian ZH. A LLMs-based method for threat intelligence information extraction. Journal of Cybersecurity. 2024, 2(2):36-46 (in Chinese)
(马冰琦, 周盈海, 王梓宇, 田志宏. 一种基于大语言模型的威胁情报信息抽取方法. 网络空间安全科学学报, 2024, 2(2):36-46)

FU Chang-Lan, Ph. D. candidate. Her main research interests include software security, threat modeling, software engineering methods and theories.

ZHANG He, Ph. D., professor, Ph. D. supervisor. His main research interests include software development productivity, DevOps, software security, software process, software architecture, AI system engineering, empirical software engineering research and practice.

GUAN Xing-Zheng, master. His main research interests include software security and threat modeling.

LI Feng-Long, master. Her main research interests include cybersecurity, cloud security, software engineering, and threat modeling.

Background

The STRIDE method has become the de facto mainstream threat modeling technology in practice, which can be used to identify security threats to systems in the early phases of software development. At present, the analysis of STRIDE threats and the construction of the rules for threat identification largely rely on manual expertise, resulting in incomplete rules for threat identification and data volume of threat modeling as well as insufficient analysis accuracy and efficiency. The problem studied in this paper belongs to the construction of threat identification rules in the field of software security threat modeling, which can be used to support automatic threat identification based on STRIDE method.

At present, the international research on threat modeling mainly includes SLR, threat identification, detection and analysis, threat modeling language, security testing based on threat modeling, etc. In addition, some studies applied threat modeling to CPS, ICS, IoT, cloud services, and 5G mobile communication, mainly using the STRIDE method. Until now there is no related work on the automatic construction of threat identification rules in this frontier area. Threat modeling is currently at a very low level of maturity, both in terms of research, tool support, and in practice. Academic efforts have not been put into practice, failing to meet the practical requirements of the industry.

In this paper, we propose a threat identification rule model based on the STRIDE method, providing comprehensive rule base data. Then, we propose an automated approach for classifying STRIDE threats, and further propose an automated approach for constructing the rule base. In addition, we design an automatic update mechanism for the rule base to ensure its effectiveness. The precision of the proposed classification approachon the CNNVD dataset reaches 92.5%, the recall at 87.6%, and the F1-score at 89.3%. Compared with the baseline method, our classification approach significantly improve precision, recall, and F1-score by 11.2%, 8.2%, and 9.2% respectively. The accuracy of the automatically constructed rules reached 89.5%. Compared with manual construction approach, our approach improves the automation level and efficiency of rule construction. As the first automated approach for construction and update of the rule base, our work is able to support automatic threat identification based on the comprehensive rule base.

This paper was sponsored by the Natural Science Foundation of Jiangsu Province (BK20241195), the Key Research and Development Program of Jiangsu Province (BE2021002-2), the National Natural Science Foundation of China (62202219, 62302210), the CCF-Huawei Populus Euphratica Innovation Research Funding (CCF-HuaweiSE2021003), the Innovation Project of State Key Laboratory for Novel Software Technology (Nanjing University) (ZZKT2025A12, ZZKT2025B18, ZZKT2025B20, ZZKT2025B22), and the Overseas Open Project (KFKT2025A17, KFKT2025A19, KFKT2025A20, KFKT2024A02, KFKT2024A13, KFKT2024A14, KFKT2023A09, KFKT2023A10). The project researches security threat identification and test generation technology based on STRIDE method, and improves the quality and

speed of threat modeling through automatic identification of security risks and test generation. In the early design phase of software development, conducting security threat analysis and further generating test cases can guide subsequent development and testing. The security problem can be better embedded insoftware security design and development, thereby enhancing the security of the software. The research achievements of our research group in this field include the published paper entitled "Threat model-based security test case generation framework and tool".

The achievements of this paper is to solve the former part of the whole subject, which is in the upstream phase of the software development lifecycle. The upstream threat modeling work, i. e. , the analysis and identification of system threats based on the threat identification rules constructed in this paper, has a docking relationship with the downstream test generation work. The two interact to form a complete whole, which plays an important role in software security practice. The achievements of this paper, including the complete rule model based on the STRIDE method, the approach of automatic construction and update of the threat identification rule model, and the STRIDE threat automatic classification approach, are the core of the automatic threat identification part, which is used to support automatic threat identification.