安全漏洞库构建及应用研究综述

曹旭栋[1),2)] 黄在起[3)] 陈禹劼[3)] 王文杰[1)] 史慧洋[1)] 李书豪[2)] 张玉清[1),2),3),4)]

[1)](中国科学院大学国家计算机网络入侵防范中心 北京 101408)

[2)](中关村实验室 北京 100094)

[3)](西安电子科技大学杭州研究院 杭州 311231)

[4)](海南大学网络空间安全学院 海口 570228)

摘 要 在以计算机和网络为基础的信息社会中,计算机和网络系统中存在的漏洞给网络信息安全带来了巨大挑战,大部分网络攻击往往都是基于漏洞发起的,并且随着近些年来漏洞数量的急剧增加以及发现速度的加快,收集、整理和利用已有漏洞就变得越来越重要.而漏洞库作为信息安全基础设施中重要的一环,不仅能够保存各类漏洞的基本信息、特征、解决方案等属性,还能快速响应漏洞信息并及时进行传播,提高公众应对信息安全威胁的能力.同时,随着机器学习、自然语言处理等技术的发展,越来越多的工作开始关注人工智能技术在智能化漏洞信息处理中的应用,漏洞库能作为其中的一个重要数据基础,在计算机领域中发挥着越来越重要的作用.漏洞库研究已成为计算机领域的一个研究热点和重点.本文首次从基础知识、背景、理论方法和创新等方面对近些年来围绕漏洞库的研究进行了全面调查,具体包括以下内容:(1)回顾了漏洞及漏洞库的背景知识,包括定义及分类;还阐述了漏洞发布与漏洞库之间的关系;(2)对漏洞库的发展现状进行介绍,同时介绍了漏洞库建设的相关标准;(3)归纳并总结了已有研究围绕漏洞库建设在漏洞信息收集、管理、字段补全以及质量评价等方面的进展;(4)归纳并总结了已有研究基于漏洞库数据分别在漏洞预测与扫描、漏洞修补、软件安全性及成分分析、网络攻击建模、安全态势分析以及漏洞特征的规律及关联性挖掘等方向的应用;(5)讨论了漏洞库研究存在的挑战和未来的研究方向.

关键词 安全漏洞;漏洞报告;漏洞数据库;漏洞自动化评估;漏洞生命周期

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2024.01082

An Overview of Research on Vulnerability Database Construction and Application

CAO Xu-Dong[1),2)] HUANG Zai-Qi[3)] CHEN Yu-Jie[3)] WANG Wen-Jie[1)] SHI Hui-Yang[1)]
LI Shu-Hao[2)] ZHANG Yu-Qing[1),2),3),4)]

[1)](*National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing* 101408)

[2)](*Zhongguancun Laboratory, Beijing* 100094)

[3)](*Hangzhou Institution of Technology, Xidian University, Hangzhou* 311231)

[4)](*College of Cyberspace Security, Hainan University, Haikou* 570228)

Abstract In the information society based on computers and networks, vulnerabilities in computer and network systems have brought great challenges to network information security. Most network attacks are launched based on vulnerabilities, and with the sharp increase in the number of vulnerabilities and the speed of discovery in recent years, it is becoming more and more important to collect, manage and exploit existing vulnerabilities. On this basis, as an important part of

收稿日期:2023-06-30;在线发布日期:2024-01-25.本课题得到国家重点研发计划项目(2023YFB3106400,2023QY1202)、国家自然科学基金重点项目(U2336203,U1836210)、海南省重点研发计划项目(GHYF2022010)、北京市自然科学基金(4242031)资助.**曹旭栋**,博士研究生,主要研究方向为网络与系统安全.E-mail:caoxd@nipc.org.cn.**黄在起**,硕士研究生,主要研究方向为人工智能与信息安全.**陈禹劼**,硕士研究生,主要研究方向为人工智能与信息安全.**王文杰**,博士,副教授,主要研究方向为信息安全与智能信息处理.**史慧洋**,博士,高级工程师,主要研究方向为网络与系统安全.**李书豪**,博士,正高级工程师,主要研究方向为威胁检测与信息对抗、网络攻防技术.**张玉清**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为网络与系统安全.E-mail:zhangyq@ucas.ac.cn.

information security infrastructure, vulnerability database can not only store basic information, characteristics, solutions and other attributes of various vulnerabilities, but also quickly respond to vulnerability information and disseminate it in a timely manner to improve the public's ability to deal with information security threats. At the same time, with the development of machine learning, natural language processing and other technologies, more and more researchers are paying attention to the application of artificial intelligence technology in intelligently processing vulnerability information. The vulnerability database can serve as an important data foundation and play an increasingly important role in the field of computers. Vulnerability database research has become a popular research topic in the field of computer science. This paper is the first comprehensive survey of research on vulnerability databases in recent years, from multiple perspectives including basic concepts, background knowledge, theoretical frameworks, and innovation points. The specific contents include the following: (1) Reviewed the background knowledge of vulnerabilities and vulnerability databases, including definitions and classifications, and also elaborated on the relationship between vulnerability publication and vulnerability databases; (2) Introduced the development status of vulnerability databases, and also discussed standards related to vulnerability database construction; (3) Classified and summarized the existing research progress in vulnerability information collection, management, filling of incomplete fields and quality evaluation around the construction of vulnerability databases; (4) Classified and summarized existing research on the application of vulnerability database data in vulnerability prediction and scanning, vulnerability repair, software security and component analysis, network attack modeling, security situational analysis, and vulnerability feature regularity and correlation mining; (5) Discussed the challenges and future research directions of vulnerability database research.

Keywords　vulnerability; vulnerability report; vulnerability database; automated assessment of vulnerabilities; vulnerability lifecycle

1　引　言

漏洞是指开发人员在信息技术、产品及信息系统的需求、设计、实现、配置、运行等阶段，有意或无意地引入的安全性缺陷. 这种缺陷往往以不同形式存在于信息系统的各个层次和环节之中，一旦被恶意主体利用，就会对信息系统的安全造成损害，从而影响构建于信息系统之上正常服务的运行，危害信息系统及信息的安全.

近年来，随着计算机网络技术的快速发展及物联网、云计算等新一代网络信息技术的普及应用，全球互联网体量急速膨胀，网络空间呈现出更加智能化、复杂化的趋势. 与此同时，网络空间安全威胁也更加严峻，安全漏洞数量呈现出快速增长的趋势，图 1 展示了美国国家漏洞库每年披露的漏洞数量情况(截至 2023 年 6 月). 可以看到，2022 年的漏洞记录数量已经接近 2016 年全年的四倍. 除此以外，基于

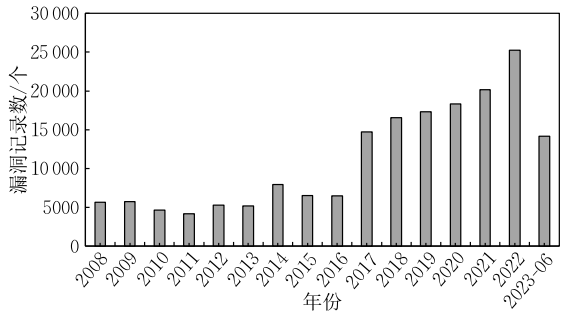

图 1　美国国家漏洞库(NVD)披露的历年漏洞数量

漏洞的安全事件层出不穷，绝大多数的安全威胁都是通过安全漏洞来实现破坏系统、窃取机密信息等目的，这对漏洞的预防和治理都提出了更加严格的要求. 比如 2021 年末被爆出的堪称互联网史上破坏力最惊人的漏洞之一的 Log4j 漏洞[1]，由于其源于被各在线应用程序、开源软件等广泛使用的知名开源日志组件 Apache Log4j 且能够导致远程代码执行，因此对业界产生了极大影响. 漏洞波及面和危害程度都堪比 2017 年在短时间内就感染了超过全

球 150 多个国家的近 20 万台计算机、造成超 80 亿美元直接经济损失的“永恒之蓝”漏洞. 如何在尽可能短的时间内对漏洞进行响应并及时披露、以最大程度地减小漏洞利用对厂商和用户带来的影响成了各国政府和安全从业者迫切需要解决的问题.

漏洞库作为对安全漏洞数据的收集和发布机构,可以全面收纳和总结漏洞内容,协调利益各方积极应对网络风险. 高质量的漏洞库不仅可以帮助计算机安全工程师及时获取有价值的漏洞并迅速做出响应,同时加快厂商和用户对漏洞的处理效率;还能够帮助政府从整体上把握网络安全的发展态势,并支持计算机领域研究人员围绕漏洞生命周期等的科研工作,推动行业的有序发展. 虽然近些年随着各国政府和公众对网络安全的逐步重视,计算机领域已经产生了许多颇具影响力的漏洞库,然而这些来自不同政府、企业和研究机构的漏洞库大都各自为战,并且在建设中普遍面临诸多难题和困境,例如不能及时有效地收集漏洞数据、对漏洞的处理和评估依赖专家经验和繁重的人力成本等等. 这些问题从一定程度上影响了漏洞库的建设效率和数据质量,也限制了围绕漏洞数据的一系列科研工作. 为此,越来越多研究人员开始将目光投向漏洞库建设,如何构建一个全面、高效和准确的漏洞数据库成了科研人员和漏洞库建设者的共同问题.

随着人工智能技术的快速发展,越来越多的研究开始将机器学习、深度学习技术应用于漏洞库建设中的各个环节,从漏洞报告的识别与收集、多源异构漏洞数据的分析和处理、漏洞数据的维护和管理,到漏洞数据中关键字段和信息的补全,最后到漏洞关系的补全与展示研究,已经形成了许多富有意义和创造性的成果. 除此以外,还有一些研究致力于漏洞库整体建设质量的提升,将人工智能技术应用于漏洞库质量评估过程,并发现了当前主流漏洞库建设所存在的问题,为未来漏洞库的质量提升和改进提供了重要参考. 可以看到,将机器学习等技术应用于漏洞库建设已经成为安全领域的一个热点工作,因此,对已有研究工作的总结和分析也具有重要意义.

另外,由于漏洞库中往往记录了漏洞从产生到消亡的整个过程,并对海量漏洞数据进行了细致准确的分析、评估和归类工作. 因此,漏洞库中数量庞大的漏洞报告可以很好地支撑围绕漏洞生命周期的研究,比如漏洞的检测与发现、漏洞利用和修补技术等. 漏洞报告中所包含丰富的漏洞特征也极大地推动了机器学习尤其是深度学习技术在这些方向中的应用. 除此以外,基于漏洞库中规模庞大的漏洞信息,也可以很好地开展软件安全评估以及网络安全态势感知等工作,并且还可以支持我们分析和探索不同字段特征间的联系和规律,从一定程度上能够促进网络安全并为漏洞治理工作带来一些灵感和启发.

据我们所知,目前国内还没有漏洞库相关的综述,而围绕漏洞库建设和应用研究近些年来已经逐渐成为该领域的热点方向[2-3]. 图 2 统计了该领域每年发表的论文数量,从图中可以看出,与漏洞库相关的研究工作呈总体上升的趋势. 在 2018 年至 2022 年,这种上升趋势尤其明显,这表明学术界对于漏洞库研究越来越关注. 因此,本文首次全面、系统地归纳并介绍了近些年来围绕漏洞库构建与应用的代表性成果,发现该领域研究存在的挑战并总结了我们认为未来可能的研究趋势. 具体地,本文的主要贡献可以总结如下:

(1) 围绕漏洞库构建研究在数据收集、处理及漏洞管理等各个阶段中遇到的问题,分别介绍了已有工作在这些方面所取得的成果,除此以外,还介绍了漏洞库质量评估的相关工作,总结了人工智能技术在漏洞库构建中所发挥的重要作用;

(2) 对漏洞库在计算机领域的应用研究进行了分类和总结,具体包括漏洞预测与扫描、漏洞修复等围绕漏洞生命周期研究以及软件安全评估、安全态势分析和基于漏洞报告的探索性研究,并分别围绕这些方面对漏洞库的应用现状进行了介绍. 这些工作一方面可以帮助读者了解漏洞库在该领域的具体应用场景,另一方面也能为漏洞库自身的建设带来一些启发;

(3) 分析并总结了已有研究在漏洞库构建及应用中还存在的不足和挑战,并展望了未来可能的研究方向,以便为后续学者了解或研究漏洞库及漏洞数据提供参考指导.

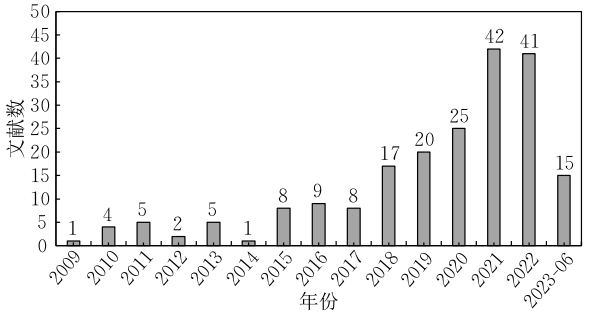

图 2 漏洞库相关论文统计结果(截至 2023 年 6 月)

本文组织结构如图 3 所示,第 2 节和第 3 节分别回顾漏洞及漏洞库的基本概念,包括它们的分类及

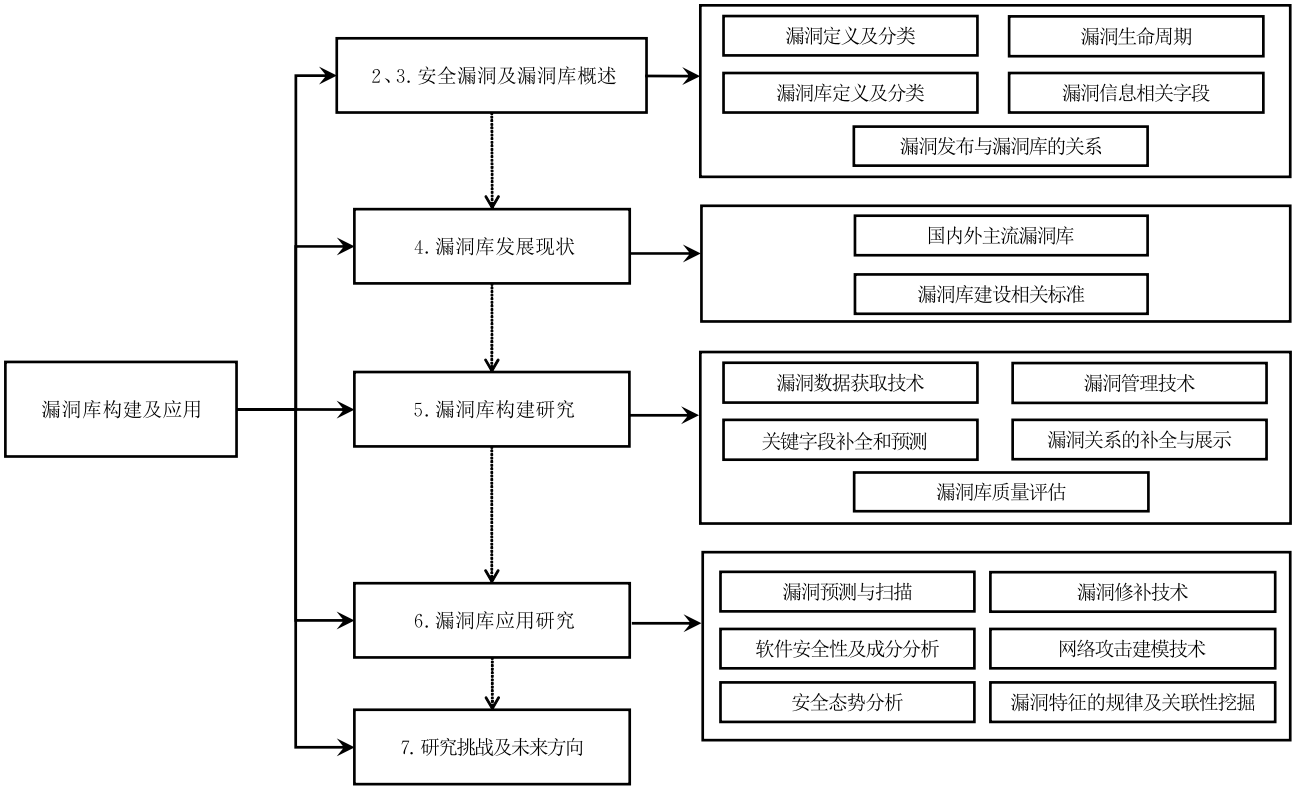

图 3　本文的组织结构图

关系；第 4 节介绍漏洞库的发现现状，并对目前主流漏洞库及广泛使用的漏洞库建设标准进行介绍；第 5 节介绍目前围绕漏洞库构建的相关研究进展；第 6 节则介绍围绕漏洞库应用的研究；并在第 7 节对现有研究中的不足和挑战进行总结，给出漏洞库研究趋势展望；最后，第 8 节对本文工作进行总结.

2　安全漏洞概述

本节主要介绍安全漏洞的基本概念及分类，围绕漏洞生命周期对漏洞从产生、利用到大规模危害直至逐渐消失的过程进行介绍，总结了对漏洞进行有效管理的积极意义.

2.1　漏洞定义及分类

安全漏洞，又称为安全脆弱性，是计算机信息系统在需求、设计、实现、配置、运行等过程中，被有意或无意引入的缺陷[4]. 这些缺陷一旦被黑客或恶意机构所利用，就会对系统或其应用数据造成严重损害，从而干扰系统的正常运行.

安全漏洞的分类是漏洞管理过程中的重要内容，对于漏洞的分类可以按照形成原因、利用位置、威胁类型等多个方面进行[4].

(1) 根据形成原因. 基于漏洞产生或触发的技术原因，漏洞可以分为代码问题、配置错误、环境问题以及其他这几类[4]. 代码问题包括资源管理错误、输入验证错误、数字错误、竞争条件问题、处理逻辑错误、加密问题、授权问题、数据转换问题以及未声明功能等. 其中，输入验证错误还可以细分为缓冲区错误、注入、路径遍历、后置链接和跨站请求伪造；授权问题还可以细分为信任管理问题以及权限许可和访问控制问题. 环境问题包括信息泄露和故障注入.

(2) 根据利用位置. 基于漏洞的利用位置，可以分为本地利用漏洞和远程利用漏洞这两类. 其中，本地利用是指需要操作系统级的有效账号登录到本地才能进行利用，远程利用是指无需系统级的账号验证即可通过网络访问目标进行利用.

(3) 根据威胁类型. 根据漏洞的威胁类型可以分为获取控制、获取信息和拒绝服务这三大类. 其中获取控制可以导致攻击者得到控制应用系统或操作系统的控制权，威胁最大；获取信息可以导致劫持程序访问预期外的资源并泄露给攻击者；拒绝服务则会导致目标应用或系统失去响应正常服务的能力.

2.2　漏洞生命周期

由于漏洞的生命周期描述了安全漏洞从产生到消亡整个过程中所表现出的不同状态，因此，可以将

其生命周期分为以下几个阶段：

(1)漏洞发现. 即通过人工或自动的方法分析、挖掘漏洞的过程，并且该漏洞可以被验证和重现，该阶段将一直持续到 Exploit 代码出现，即确认了可利用性.

(2)漏洞利用. 利用漏洞对计算机信息系统的保密性、完整性和可用性造成损害的过程，该阶段从 Exploit 代码出现时开始，将一直持续到该漏洞被完全修复.

(3)漏洞修复. 通过补丁、升级版本或配置策略等对漏洞进行修补的过程，使得该漏洞不能够被恶意主体所利用，漏洞被彻底修复后，其生命周期也将结束.

如图 4 所示，从漏洞产生到其被公开发现或公开披露的阶段被称为“0-day”. 在发现缺陷后，测试人员还需要对其做进一步的分析，以确认是否为安全漏洞，并为该漏洞开发概念验证性的攻击代码(Proof Of Concept，POC)以及确认可利用性. 在发现安全漏洞并给出渗透攻击代码后，负责任的发现者会首先通知厂商进行修补，等厂商给出补丁后再进行公布，而“黑帽子”与“灰帽子”可能会在封闭小规模团队中进行秘密地共享，以充分利用这些安全漏洞和渗透攻击代码所带来的价值，因此，从漏洞被发现到官方发布修补方法的阶段被称为“1-day”. 之后，随着用户逐渐地安装补丁和新漏洞的产生，该漏洞的危害也就变得越来越小，并逐渐走向消亡.

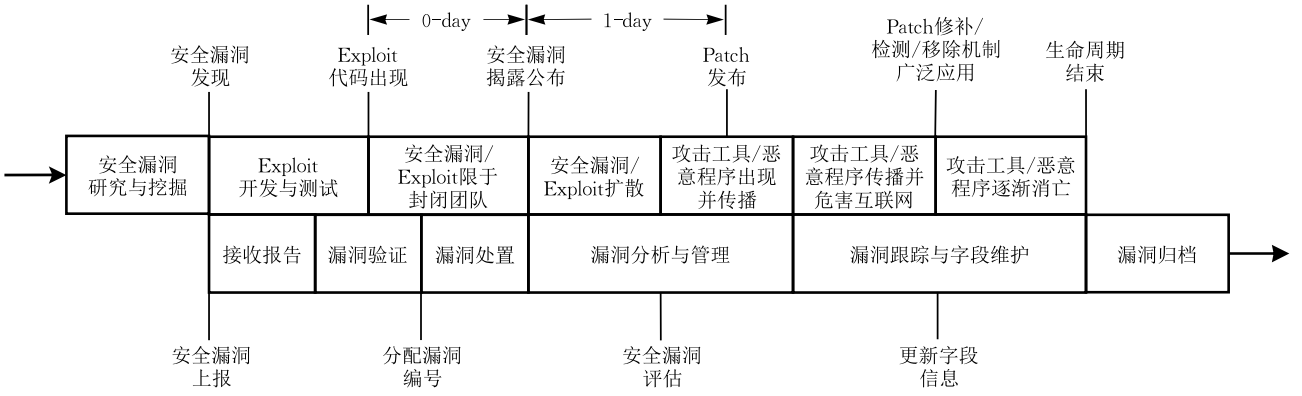

图 4 安全漏洞生命周期

在以上漏洞从产生开始到传播利用、修复并消亡的整个过程，漏洞管理均起到了十分重要的作用. 漏洞库通过收集和整合最新的漏洞信息并进行快速响应和即时传播，从而加快漏洞修复进度以尽可能地缩短漏洞生命周期，保障厂商和用户的信息安全.

2.3 小 结

第 2 节介绍了安全漏洞的基本定义、分类及生命周期，可以看到，漏洞是普遍存在并且难以避免的，其时效性往往取决于厂商修补效率及用户的响应，因此为了尽量减少漏洞被恶意利用对系统造成的损害，需要进行有效且高效的漏洞管理，而漏洞库在解决这一需求中则发挥着重要作用.

3 漏洞库概述

本节介绍了漏洞库的概念及分类，并围绕漏洞生命周期介绍了漏洞发布与漏洞库的具体关系，除此以外，也对漏洞信息的相关字段进行了介绍.

3.1 漏洞库定义及分类

安全漏洞数据库(又称漏洞库)是指收集和管理国内外网络安全缺陷及漏洞资料以进行存储、共享和发布的平台[5]. 作为信息安全基础设施中的重要组成部分，漏洞库在安全预警和应急响应领域发挥着重要作用.

国内外安全漏洞数据库数量繁多，对于当前漏洞库的分类可以围绕机构设置、数据特点、运营模式等方面进行，下面将进行详细介绍：

(1)依据机构设置. 可以按照建设方的不同将漏洞库分为政府漏洞库和民间漏洞库两类. 其中，政府漏洞库大都面向公众和社会，其发布的漏洞信息也更具全面性和权威性，例如中国国家漏洞库 CNNVD[6]、国家信息安全漏洞共享平台 CNVD[7] 以及美国国家漏洞库 NVD[8] 等. 而民间漏洞库则相较而言针对性更强，例如 SecurityFocus[9]、Secunia[10] 等安全企业漏洞库主要关注影响力大、通用性较广的产品漏洞，类似的还有 IBM X-Force Exchange[11]、Seebug[12] 等.

(2)依据数据特点. 可以按照描述语言的不同，分为中文漏洞库、英文漏洞库等；也可以按照所侧重数据功能上的不同，分为漏洞描述库、漏洞利用库等.

ExploitDB[13]更专注于发布漏洞利用信息，CVE[14]中则主要包含描述信息.

(3) 依据运行模式. 可以将漏洞库分为开源漏洞库和商业漏洞库. 其中，NVD、CVE 等政府漏洞库大多是以开源模式提供数据服务，因此，也是目前围绕漏洞库领域的学术研究中广泛使用的数据来源.

3.2　漏洞发布与漏洞库的关系

漏洞库作为安全漏洞的主要发布平台，为漏洞数据提供了全面、准确且标准的披露与共享服务. 漏洞库对于漏洞的管理与发布过程也遍历了整个漏洞生命周期，并且同时受到漏洞报告者、厂商及用户等的影响[15]. 下面以 CVE 漏洞的发布过程为例进行介绍.

如图 4 所示，当测试人员发现新的漏洞时，可以通过提交漏洞报告的形式向 CVE 编号机构 CNA (CVE Numbering Authority) 请求唯一的 CVE 编号(CVE-ID). MITRE 作为最重要的 CNA，负责接收报告、验证漏洞并对 CVE 信息进行编辑. 当该漏洞通过验证并确认是首次发现后，将进入漏洞处置阶段，MITRE 将为其分配一个 CVE-ID 作为漏洞编号，同时与相关提供者、网络运营商协同开展漏洞分析与处理工作. 相应地将该漏洞信息通过 CVE 列表公开发布，美国国家标准与技术研究院的 NVD 团队将对已在 CVE 列表中更新的漏洞进行同步更新，并对漏洞的 CVSS (Common Vulnerability Scoring System)[16]、影响范围等进行评估，以完善各字段信息. 同时每个 CVE 条目还包含一个外部参考链接的列表，以追踪厂商修补信息、第三方技术报告或与该漏洞相关的博客/论坛的链接，直到厂商补丁程序、安全公司提供的检测和移除机制得到广泛应用后，该漏洞生命周期结束，漏洞条目也将以最近更新版本的完整形式记录存储在 NVD 和 CVE 数据库中.

3.3　漏洞信息相关字段

为了尽可能清楚和详细地描述漏洞条目，以增强研究者对于漏洞原理的理解和对漏洞整体情况的掌握，漏洞库针对每个漏洞都会增加统一的字段信息. 如图 5 所示，每个漏洞库中包含的字段都有所不同[17]. 其中，漏洞库涵盖的字段主要包括：

(1) 标题. 漏洞的简要概括，一般包括软件名称和漏洞类型；

(2) 标识号. 包含 CVE-ID 以及不同漏洞库可能分配的不同标识号；

(3) 漏洞描述. 有关漏洞原理、触发方法和漏洞类型等的描述；

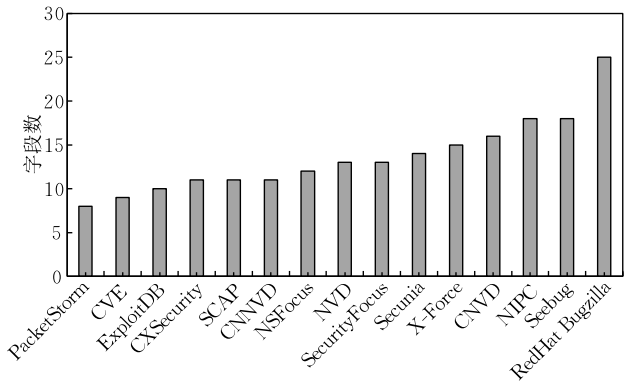

图 5　主流漏洞库用于描述漏洞信息的字段数

(4) 漏洞类型. 即 CWE(Common Weakness Enumeration)类型[18]，或各库自己定义的漏洞类型；

(5) 受影响厂商、产品和版本. CPE(Common Platform Enumeration)标准[19]提供了对厂商、产品(软硬件)、版本进行标识的方案，受影响的产品也包括了受影响的软件依赖和操作系统等；

(6) 漏洞危害性. 即 CVSS 分值，或各库自己定义的危害评级；

(7) 漏洞利用信息. 例如 POC 代码，方便用户进行漏洞重现；

(8) 参考链接. 其他可供参考的索引，如漏洞相关厂商给出的补丁链接；

(9) 发布时间. 漏洞信息发布日期；

(10) 更新时间. 该漏洞信息最后一次更新时间；

(11) 漏洞提交者. 漏洞的提交者或发布者.

3.4　小　结

第 3 节不仅介绍了漏洞库的概念及分类，也介绍了漏洞库在漏洞生命周期中所发挥的重要作用. 对于已解决的漏洞，其信息将会以格式化或半格式化的形式记录在漏洞库中，作为围绕漏洞研究的重要数据支撑. 因此，漏洞库在学术研究中扮演着重要角色.

4　漏洞库发展现状

第 4 节将首先介绍国内外主流漏洞库的发展现状，接着对当前国际广泛使用的 CWE、CPE、CVSS 等漏洞库建设相关标准进行介绍，最后对当前漏洞库的发展进行总结.

4.1　国内外主流漏洞库

图 6 显示了国际主流漏洞库的成立时间，可以看到，欧美等发达国家在安全漏洞库建设方面起步

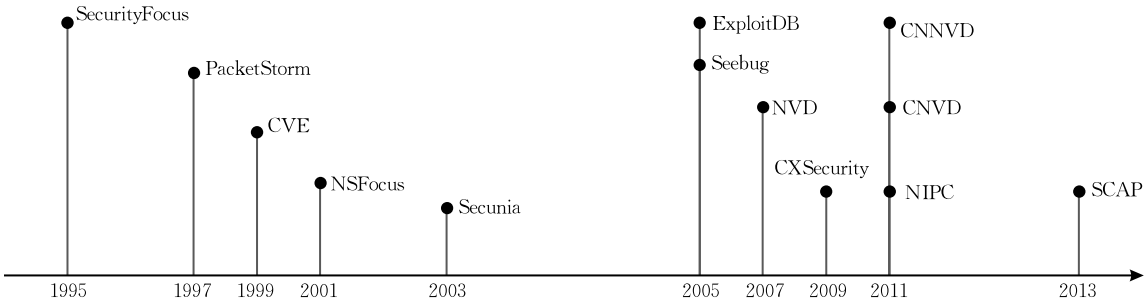

图 6 主流漏洞库上线时间

较早,其政府和民间机构已经建设完成了具有一定影响力的漏洞库,且目前在国际上广泛使用的主流漏洞库也主要集中在欧美国家的英文漏洞库中,例如政府支持的 NVD、CVE 库,以及安全组织承办的 Secunia、CXSecurity[20]、ExploitDB、PacketStorm[21]、SecurityFocus 等[22]. 同样,随着漏洞数量的激增以及漏洞库的作用不断凸显,我国的漏洞库建设也取得了迅速发展,先后成立了中国国家漏洞库 CNNVD、国家信息安全漏洞共享平台 CNVD、国家安全漏洞库 NIPC[23] 和国家工业信息安全漏洞库 CICSVD[24],逐渐形成较为完善的漏洞管理体系. 除此以外,清华大学也推出了 SCAP 中文[25],集成了信息安全相关标准和各种漏洞数据,一些国内安全公司也根据自身需求成立了 NSFocus[26]、Seebug 等民间漏洞库.

表 1 统计了截至 2023 年 6 月,当前主流漏洞库的漏洞收录情况,其中独立性反映了漏洞数据来源是否主要依赖其他漏洞库,从整体情况来看,欧美国家的漏洞库独立性普遍较国内更强. 在主流漏洞库中,目前最受工业界和学术界欢迎且影响最大的是隶属于美国国家标准与技术研究院的美国国家漏洞库 NVD. 作为国际上安全预警和漏洞发布的重要平台,NVD 漏洞库除了拥有丰富的漏洞数据且结构规范以外,还严格按照 CVE 标准对漏洞进行编号命名[27],并且漏洞的评级和分类也严格按照 CVSS 和 CWE 标准进行. 对于受影响的软件和版本,NVD 使用 CPE 规范的语言进行描述,使得漏洞条目可以很好地支持自动化分析和处理.

表 1 主流漏洞库数据统计

漏洞库名	语言/地区	披露数(万)	字段数	数据独立性
PacketStorm	英/美国	12.7	8	√
CVE	英/美国	20.6	9	√
ExploitDB	英/美国	4.6	10	√
NVD	英/美国	20.6	13	√
SecurityFocus	英/美国	10.2	13	√
CXSecurity	英/美国	3.9	11	√
X-Force	英/美国	11.9	15	
Secunia	英/丹麦	8.7	14	
CNNVD	中/中国	21.7	11	
CNVD	中/中国	18.4	16	√
NIPC	中/中国	42.6	18	
SCAP	中/中国	40.6	11	
NSFocus	中/中国	8.1	12	
Seebug	中/中国	5.9	18	

除此以外,ExploitDB、Secunia、SecurityFocus 和 X-Force 等漏洞库也都在工业界和学术界取得了很好的应用[28]. ExploitDB 作为一个非营利性安全漏洞库,由于其提供了大量漏洞验证代码,因此在安全领域具有非常大的影响力;Secunia 漏洞库由 IT 安全厂商 Secunia 于 2002 年组织成立,其数据一部分来自 Secunia 公司获取的漏洞信息,一部分则主要通过对主流 IT 企业的安全公告进行收集获得,由于该库收录的漏洞数据量庞大且数据来源权威严谨,因此受到了业界的广泛关注;SecurityFocus 漏洞库同样包含了大量的漏洞数据,并且其除了会对漏洞信息进行简要描述,也会发布攻击方法、脚本实例等内容,能为安全工作者分析漏洞提供很好的便利;X-Force 库则在保证所收录漏洞全面且权威的基础上,支持 CVSS 漏洞危害评估标准,并且应用性强,其数据已经被应用于 ISS 开发的漏洞扫描器等产品中.

4.2 漏洞相关标准

目前主流漏洞库在对漏洞字段进行处理过程中所广泛使用的标准主要包括 CWE、CVSS 和 CPE 标准,且 CVSS 和 CPE 均属于 SCAP 标准协议(Security Content Automation Protocol),作为当前比较成熟的信息安全评估标准体系,SCAP 为安全工具实现标准化、自动化提供了很好的解决方案,下面将分别对以上三个标准进行简单介绍:

(1) 通用平台枚举 CPE. 作为一种对应用程序、操作系统以及硬件设备进行描述和标识的标准化方案,它通过提供一个标准且机器可读的格式来实现对 IT 产品和平台的编码,CPE 枚举了漏洞影响的软件、软件版本号、系统平台和厂商等信息.

(2)通用缺陷枚举 CWE. 作为一个统一的、可度量的软件缺陷描述体系，CWE 通过创建软件缺陷和漏洞类别列表来帮助用户能够更好地理解软件缺陷，其作为目前最为权威和全面的漏洞分类标准，列举了目前已知的几乎所有漏洞类型. 为了便于识别，CWE 还为每一个缺陷类型分配了一个唯一的编号，即一个 CWE 编号只对应于一种类型的缺陷. NVD 采用部分 CWE 编号作为漏洞的分类依据.

(3)通用漏洞评分系统 CVSS. 作为一个开发共享的、用于计算机信息系统安全漏洞威胁严重性评估的系统，当前 CVSS 已经开发迭代到了 3. X 版本，其评分标准由基本分数指标、时间得分指标和环境得分指标三部分构成，并且 CVSS 会将漏洞严重程度划分为五个等级，还支持通过提供 0.0 到 10.0 之间的数字分数来评估缺陷，以此帮助相关人员确定修复和处理该漏洞的优先级.

4.3 小　结

国内外近些年来的漏洞库建设都取得了较快发展，并且已经逐渐得到了软件厂商、安全公司以及科研工作者的广泛关注. 各国政府及企业先后建设的包含多个软件厂商漏洞信息的安全漏洞库在一定程度上整合了漏洞的发布. 可以看到，为了便于进行数据分析以及自动化工具的开发，漏洞库的建设工作已经逐步向标准化迈进，例如 CPE、CVSS、CWE 等共同保证了 NVD 等漏洞库中数据字段的规范性，并被业界广泛接受. 与此同时，保持高效的漏洞披露效率、扩大漏洞数据的收集范围以及保证信息的准确性也已经成为了漏洞库重点考虑的建设方向.

5　漏洞库构建研究

对安全漏洞的及时收集和发布是漏洞库的一个重要职能，随着漏洞数量的急剧增加以及机器学习技术的不断发展，进行自动化和智能化的漏洞处理可以极大提高漏洞管理的效率，已经成为了漏洞库发展的一个重要趋势，因此本节旨在介绍已有研究围绕漏洞库建设过程中漏洞数据从自动化获取、管理、关键字段的补全和预测到关系补全与展示等各个环节开展的工作. 最后，本节也对漏洞库质量评估的相关研究工作进行了总结，介绍了漏洞库在建设中所存在的一些问题，下面将展开进行介绍.

5.1 漏洞数据获取技术

漏洞库发布了关于网络漏洞的开源信息，许多企业、政府部门以及网络防御工具都使用漏洞库作为数据来源来更新给定网络的风险和威胁. 因此，一旦发现漏洞，更新漏洞库至关重要，以便安全分析人员在对抗网络攻击者时具有优势. 然而，受厂商对产品维护方式和漏洞发现者主观因素等的影响，新产生的漏洞往往可能散布在软件缺陷管理系统、论坛等不易被公众获取的来源. 因此，如何收集这些未被披露的安全漏洞就成了学术界十分关注的问题[29-33]. 除此以外，相比于一些商业、特定类型或是小型漏洞库，公众往往对大型通用漏洞库如 NVD、CVE、CNNVD 等更加关注[34-36]. 如何收集不同来源的漏洞数据[37-47]，并对异构数据进行处理[48-50]，也是围绕漏洞库研究在学术领域较为关注的问题. 因此，本文将漏洞数据获取技术的研究进展分为多源数据获取、安全缺陷报告预测和补丁数据获取这三个方面分别进行介绍.

5.1.1 多源数据获取

越来越多的国家和企业都建立了自己的漏洞库，如法国 VUPEN、国家信息安全漏洞共享平台 CNVD 以及 IBM 公司的 ISS X-Force 等. 然而不同漏洞库所收集的漏洞类型以及披露漏洞的形式都可能会有所差别[36,46]. 因此，为及时获取到全面准确的漏洞信息，如何借助机器学习、深度学习等技术实现自动化的多源漏洞采集就成了数据库建设人员及基于漏洞数据的研究人员首先需要解决的问题. 在漏洞库中，漏洞信息往往会以自然语言文档的形式展示，并且呈现半结构化或非结构化的特征[44]. 而信息抽取作为搜索和分析非结构化数据以发现其中结构化任务[51]的技术在这一场景中就可以取得很好的应用，Bridges 等人[52]和 Russo 等人[53]通过词性标记和模式匹配等基于规则的技术来提取漏洞信息. 尽管此类模型在特定目标域中提取较为准确，但当文本的形式不符合规则中指定的模式时，就可能无法得到关键信息. 为此，Chaleshtori 等人[35]提出了一种基于 BERT(Bidirectional Encoder Representation from Transformers)的漏洞信息抽取模型，通过将预训练模型在领域数据集中进行微调，实现了安全领域内的命名实体识别(Name Entity Recognition，NER)，并且能够提取文本中的复杂特征，具有较好的泛化能力.

在数据检索与获取算法的改进方面，Li 等人[47]将垂直搜索技术应用于软件安全漏洞的获取. 首先使用关键字训练器来获取软件安全域中的域关键字；然后在分析获得的域关键字和 URL 拓扑结构后设计网页过滤器；最后，设计了一个基于网页过滤器的垂直搜索爬虫，用于搜索软件安全漏洞信息.

Pereira 等人[34]提出了一个自动化的软件漏洞采集流程,能够实现从开源漏洞库(如 CVE)、缺陷跟踪系统(如 Bugzilla)以及版本控制系统(如 GitHub)等多渠道收集漏洞信息. Armold 等人[36]对不同源漏洞库进行数据获取前,直接过滤掉了一些大量重复使用 NVD、CVE 中数据的开源漏洞库(如 OSVDB),并通过使用数据获取模块和关系数据库实现了漏洞数据的收集.

5.1.2 安全缺陷报告预测

漏洞在被漏洞库收集并披露前,很有可能就以缺陷的形式存在于 Bugzilla、JIRA、GitHub 等缺陷管理平台中[33,54-55],尤其是对于开源软件漏洞. 因此,如何在漏洞生命周期早期及时获取相关信息并做处理,是漏洞库建设人员、厂商以及用户都十分关注的问题. 为此,Zhou 等人[33]发现对于大多数安全漏洞,缺陷报告中丰富的上下文信息通常足以支持安全研究人员确定该条目是否与漏洞相关,并在此基础上分析了 GitHub、JIRA 和 Bugilla 上数千个开源项目的提交信息和缺陷报告,通过构建 K 折漏洞识别分类器,选择了六个常用的分类算法,接着对每个基本分类器的测试结果再进行逻辑回归以找到算法的最佳组合,最终分别实现了 70%和 71%的精确率和召回率.

除此以外,加入缺陷报告中的元特征和文本功能等信息已被证明能够很好地提升分类器的性能[56]. Behl 等人[57]采用了 TF-IDF 分析和识别漏洞报告的特征,并使用朴素贝叶斯作为其模型分类算法,实现了 93.99%的准确率. 在该分类器基础上,Yang 等人[58]提出了一种基于检索词频率的高影响缺陷报告识别模型. Cao 等人[59]发现安全缺陷报告的预测受到领域样本的稀缺性和不平衡性以及报告复杂特性的限制. 因此,提出了基于 BERT 的安全漏洞报告预测模型 SbrPBert,在测试数据集中获得了比基线模型更好的效果.

5.1.3 补丁数据获取

与漏洞描述、严重性、受影响软件等自然语言信息不同的是,漏洞修复补丁的产生存在一定的滞后性及不易收集等特点. 因此,尽管 NVD 等漏洞库会及时发布漏洞信息,但有相当多漏洞的安全补丁仍然不会及时披露[39,41,60]. 如何对安全补丁进行有效和及时的收集和管理已经成为学术界广泛讨论的问题[61].

Zhou 等人[41]首先指出该领域研究面临的几个挑战,如缺乏标记数据集、提交消息可能过于冗长和已有的代码嵌入方法(如通用的代码分布式表示[62])并不适用于进行安全漏洞的有效表示等. Wang 等人[63]通过使用人工提取的特征,例如内存运算符的数量、循环的数量等来识别提交之间的安全补丁. 然而,传统的机器学习方法可能无法充分探索和提交源代码的语义并依赖人工基于领域知识或消息提取的特征,使得模型泛化性较差,并可能带有一些领域专家的偏见. 因此,Zhou 等人[41]基于开源存储库中与安全相关的提交设计并实现了一个由两个复合神经网络组成的安全补丁识别系统,并通过实验证明模型效果明显优于 SVM 和 K 折堆叠算法. 为了提取更深层特征,Wang 等人[64]提出了一种基于排名的漏洞提交匹配方法 VCMATCH. 除了漏洞和补丁提交之间的浅层统计特征外,还提取了漏洞描述和提交消息等深层语义特征,并通过结合 XGBoost、LightGBM 和 CNN 等三种分类模型,实现了对开源软件补丁信息的有效识别. 文献[38,40,42-43,45]则更加关注于补丁数据集的构建工作. Wang 等人[40]和 Hong 等人[38]分别基于所提出的最近链接搜索和高覆盖率方法试图找到匹配的安全补丁以构建领域补丁数据集. Tan 等人[43]通过对代码提交进行排名以试图找到更多的安全补丁. Bhandari 等人[42]则更加关注于数据集中字段信息的扩充. 除此以外,Nguyen-Truong 等人[37]发现现有技术仅考虑从提交中获取数据,然而其中并不总是包含足够的区分性信息,因此通过结合缺陷跟踪器中的数据来进一步丰富数据来源,所提出的方法使得分类模型取得了更好的效果.

5.1.4 小 结

漏洞数据的获取是漏洞库建设中的重要一环,为了实现全面的漏洞数据收集,本节调研了围绕漏洞生命周期中从漏洞产生到补丁发布阶段. 现有工作是如何充分考虑数据来源并进行获取的. 在多源数据获取方面,由于传统的机器学习技术依赖于特征工程,在对不同来源进行数据提取时就无法很好地提取出关键信息. 因此,漏洞信息抽取技术转向使用 BERT 等大型预训练模型实现,能够实现对复杂特征较好的提取. 然而,由于漏洞修补信息往往存在较明显的滞后性,现有漏洞库无法及时全面地跟进并公开,因此,对于安全补丁的获取也成了一项热点工作. 由于不能直接依赖漏洞库数据,该项工作就首先面临如何在开源项目的缺陷信息中区分哪些是和安全相关的问题,同时,漏洞在被漏洞库收集并披露前,本身就很有可能以缺陷的形式存

在于缺陷管理平台中，因此，安全缺陷报告的预测成为研究人员普遍关注的问题，现有技术虽然很好地解决了领域样本存在稀疏性和不平衡性的问题，也在数据集中取得了不错的分类效果，但是，未能充分考虑模型在跨项目任务中的适用性与通用性，所选取的研究数据来源较为单一，并且数据集中的数据标注也受到提交报告用户专业知识和标注专家主观因素的影响，并非完全准确，这也影响了模型的训练效果，这些都是在漏洞收集的相关工作中还有待解决的问题.

5.2　漏洞管理技术

漏洞管理技术是一系列用于有效管理和维护漏洞的技术和方法. 在漏洞库的构建过程中，虽然已经解决了不同源信息的采集问题，但采集到的异构数据还存在冗余重复问题，并且在漏洞数据管理阶段，还存在披露的新数据更新不完整的问题以及新漏洞的审批流程缓慢导致数据更新存在延迟的问题. 除此之外，由于漏洞特征丰富且关系复杂，而传统的检索技术缺乏数据间关联性分析，且库中对于漏洞各字段的描述还可能并不准确完善等，因此，在查询漏洞时可能无法直接检索和展示出多条相关的数据信息，这也是围绕漏洞管理研究在学术领域较为关注的问题. 综上，本节将分别围绕异构漏洞数据融合、漏洞数据的维护以及漏洞展示与共享这三个方面对漏洞管理技术的研究进展进行介绍.

5.2.1　异构漏洞数据融合

由于现有安全漏洞库的漏洞数据来源不同，因此数据间可能存在异构和冗余，从而使得漏洞数据质量降低，并且对同一漏洞的统一描述和检索变得困难. 因此，如何对来源不同的异构漏洞数据进行对比融合处理就成为漏洞库构建工作中面临的问题. Guo 等人[49]提出了一种基于深度神经网络的定制 NER 方法，用于提取 CVE 和其他漏洞数据库的描述，并使用从其他漏洞库相应的安全漏洞描述中所提取的漏洞关键信息来补充 CVE 数据，最后通过设计实验证明了方法的有效性. 虽然 MITRE 已试图通过为每个漏洞分配唯一 ID 来缓解漏洞标识的问题，但 Sun 等人[48]发现与 NVD 漏洞库不同的是，大多数漏洞库（例如 ExploitDB、Openwall、IBM X-Force）并不会在所有报告中引用 CVE-ID. 比如对于 ExploitDB，有大约 52% 的条目信息都缺失 CVE-ID. 因此，异构漏洞数据的融合无法通过简单的编号索引匹配来实现.

为此，Li 等人[65]基于所设计的漏洞受影响软件名称及其版本相似性测量算法，提出了一种漏洞数据相似性测量智能融合框架 IFVD，来解决 NVD、Secunia 和 SecurityFocus 库中异构漏洞数据统一映射的问题，在扩充漏洞数量的同时，保证了数据质量. 在此基础上，Sun 等人[48]通过 BERT 的命名实体识别和 QA（Question Answering）问答模型自动从漏洞描述中提取关键漏洞信息，包括产品名称、版本、组件和漏洞类型等命名实体以及攻击媒介、根本原因和影响等短语，通过匹配这些提取的关键信息来链接来自不同数据库的报告，实现了较好的匹配效果.

5.2.2　漏洞数据的维护技术

从安全研究人员初步上报漏洞数据，到 CVE、CNNVD 等漏洞平台将其漏洞信息公布，需要经过一系列缓慢的人工审查过程. Rodriguez 等人[66]通过将 NVD 上的数据资源与其他来源比如暗网上的数据进行对比，发现这些来源的漏洞数据披露均会比 NVD 漏洞库提前 1～7 天，该研究指出了新漏洞数据存在的发布滞后问题. 当漏洞数据披露后，相关字段信息的发布也存在滞后性. 例如 Chan 等人[67]发现，在漏洞库中，从漏洞被公开披露到相关元数据（例如受影响的软件及其版本信息）发布的平均时间间隔是 35 天. 这一现象反映了新发布漏洞数据面临信息不完备的问题. 因此，如何解决新漏洞数据的发布滞后与披露后信息不完备的问题成为漏洞数据维护技术应当解决的两大难题.

最近的工作对解决新漏洞数据不完备且滞后问题进行了探索. 在如 NVD 等的漏洞数据库中，漏洞数据的关键字段通常包括漏洞描述字段、CWE 字段、CVSS 字段、CPE 字段等，其中在漏洞数据更新的初始阶段，仅包含漏洞描述，而其余的 CWE、CVSS、CPE 信息都存在滞后更新. 这是由于这些信息通常是由安全研究人员进行人工审核综合研判才能得出的. Chan 等人[67]认为导致当前研判分析效率较低的一个重要原因是不同库对 CVE 的记录方式并不统一，这给自动化处理 CVE 数据以及软件组成分析工具的有效性带来了挑战. 因此，为了确保漏洞数据更新的相对完备，文章指出当前开源软件的开发人员以及更新和维护 CVE 数据库的实体都应采用双方商定的 CVE 文档标准，以减少两者文档转换的流程，加快数据的更新. 他们还指出 CVE 漏洞库存在补丁信息缺失的问题，从而导致用户很难从 CVE 条目中识别出确切的修复版本，即使有些 CVE 包含了版本信息，但可能存在分支、子版本

等复杂情况,使得准确确定哪个版本是修复版变得困难.因此,应用一致且公认的 CVE 文档标准是十分必要的.除此以外,针对严重性较高的漏洞更新滞后的问题,Gong 等人[68]提出了利用深度学习技术和历史漏洞中嵌入的知识来帮助分析人员加快对 CVE 特征的分析方法,从而得出漏洞数据的特征信息,评判新漏洞数据处理的优先级.

5.2.3 漏洞展示与共享

为了解决漏洞数据的关联性信息检索展示问题以及漏洞数据中可能存在的信息孤岛问题,围绕漏洞数据的展示与共享,已有研究分别在提高漏洞检索效率以及构建范围更广的漏洞数据共享方面做出了一些探索.

从搜索引擎中搜索出来的常常是一条独立的漏洞数据,而无法查看与之相关联的所有漏洞信息.比如某一产品可能同时存在多个相互影响的漏洞,或者多个产品共用同一漏洞,而这些关联关系在常规搜索中较难被一次性展示.因此,如何检索并展示出多条信息相关的漏洞成为漏洞管理过程面临的问题.为了提高检索关联漏洞数据的效率,Tsutsui 等人[69]通过构建 CVE 及其关联信息的本体模型,利用本体中的预定义关系来自动抽取相关的多条漏洞信息,从而能够实现在单一搜索操作中获取到较全面的漏洞信息,而不必逐一查找每个单独的 CVE 或产品,并且避免了遗漏重要关联性信息.为了对漏洞数据进行过滤和排序,缩小研究人员感兴趣的漏洞数据范围,Pham 等人[70]基于 NVD 中的漏洞数据,构建了一个交互式的漏洞扫描可视化交互系统,它包含有多个链接视图,允许用户通过该系统查看扫描出的漏洞数据中各个维度的关系.在此基础上,Reynolds 等人[71]对漏洞数据的展示进行了进一步的研究,他们设计了两个面向开发人员与安全专家的漏洞扫描交互可视化系统,用以帮助开发人员与安全专家概览受影响软件的安全状态,查看漏洞的分布情况,比较受影响软件之间的版本差异.

在安全领域,存在着大量漏洞数据来源,如不同的漏洞数据库、安全厂商报告等,这些数据通常被存储在不同的系统和平台中,并且覆盖范围以及格式等方面存在明显的差异,容易导致信息孤岛的问题[72].现有漏洞数据共享方法缺乏语言以及数据格式之间的统一转化.为解决该问题,Zheng 等人[73]提出了国际漏洞数据库联盟(IVDA)的概念,它指的是由来自不同国家的安全组织组成漏洞数据联盟,提供系统的政策和标准来管理不同语言的软件漏洞,并与其成员达成协议,以加强国际合作和沟通.但 IVDA 距离实际采用仍有一段距离,因为非英语地区的漏洞披露仍处于不成熟状态,满足 IVDA 的实现要求需要时间.

5.2.4 小 结

本节围绕解决多源异构漏洞数据的融合冗余问题、新漏洞数据更新不完整和新漏洞的审批流程缓慢等问题,分别从异构漏洞数据的融合,漏洞数据的维护技术以及漏洞数据的展示与共享三个方面展开讨论.对于异构漏洞数据的融合问题,现有工作基于描述信息、软件名称及版本等来实现漏洞信息的关联开展了一系列研究并取得了较好的效果,但是更多是在考虑对于 CVE 漏洞条目信息的扩充.对于没有 CVE-ID 的漏洞,即未被 NVD、CVE 所引用的多源漏洞数据,仍然缺乏关联性分析,这个问题目前还有待解决.在漏洞数据的维护技术方面,现有研究针对漏洞数据的更新以及披露延迟进行了初步的探索,揭示了 NVD 中的漏洞披露延迟,但研究中尚缺乏合适的解决方案,仅指明了该问题.对于漏洞数据的展示与共享方面,现有研究针对来自 NVD,CVE 等漏洞平台的数据,设计了可视化交互系统,从整体层面查看漏洞数据之间的相互关系,还针对用户的不同,面向软件开发者和安全研究人员进行了不同的交互式系统设计,但不同的可视化系统采用的都是不一致的设计标准,这使得用户需要花费额外的时间和精力来适应和理解不同平台的展示方式.未来的研究可以考虑采用统一的漏洞可视化系统设计标准来解决这一问题.

5.3 关键字段补全和预测

漏洞不同字段分别描述了该漏洞的标识、类别、发布、影响系统、严重性以及解决方案等基本属性和关键信息.漏洞库的建设为了保证数据的易读性和全面性,往往会以结构化或半结构化文本的形式对漏洞的各个字段进行展示以实现漏洞披露,并且各字段遵循简明和客观的原则,为用户提供准确的数据服务.也正是基于此,漏洞库逐渐成了围绕漏洞研究以及政府部门、企业十分依赖的漏洞获取途径.然而,随着漏洞数量的不断增长以及漏洞分析、利用等技术的自动化和智能化,如何快速准确地对新产生的漏洞进行自动化地表征和展示,即各字段信息的生成和预测已经成为目前该领域又一个十分关注的问题.总的来看,目前的研究主要集中在对漏洞描述信息的补全和生成[49,72,74-77],对漏洞类型的预测[78-82],对漏洞影响的软件名称及版本的预

测[83-86]和对漏洞严重性的预测[87-102]这几个方面，下面将分别进行介绍.

5.3.1　漏洞描述生成

CVE 官方漏洞提交指南中明确要求用户在提交漏洞报告时应该详细描述漏洞类型、根本原因、受影响的产品、攻击者类型、攻击载体以及影响等至少六种关键信息[53]. 这些信息可以从多个角度描述漏洞，对于开发人员及安全研究人员而言，要想对漏洞有完整的了解，这些关键信息都必不可少，对于漏洞库建设而言，清晰准确的漏洞描述可以帮助用户更好地区分相似度较高的漏洞. 但是，Guo 等人[49]通过对漏洞描述的拆分和统计，发现 CVE 中存在严重的关键信息缺失问题，这也激发了研究人员对描述信息自动生成的兴趣.

Tian 等人[72]也同样指出目前来自不同来源的漏洞信息通常是模糊的文本描述，无法在自动化过程中得到有效共享和利用. 因此，从漏洞描述格式的角度出发，提出了一种基于 XML 的通用漏洞标记语言 CVML，以更结构化的方式描述漏洞，CVML 除了包含当前大多数漏洞数据库的常规信息外，还支持分类、评估、存在性检查和攻击生成等信息. 从丰富描述信息来源的角度，Guo 等人[49]从其他漏洞库的相应安全漏洞描述中提取漏洞关键方面来补充 CVE 数据. 而 Sun 等人[75]则发现实际上有超过 73%的漏洞利用程序比相应的 CVE 更早出现，还有大约 40%的利用程序甚至仍没有 CVE. 因此，提出了一种基于 NER 和 QA 的信息提取方法，通过从 ExploitDB 条目中提取漏洞组件、版本、漏洞、根本原因等 9 个关键的漏洞信息并按 CVE 描述模板自动化地生成 CVE 描述.

机器学习技术需要带有标记的实体和概念以支持模型训练. 然而与大量标记的通用文本不同，目前仍没有用于训练信息提取模型的大型标记漏洞描述数据集，尤其是对于基于短语的概念，如根本原因、攻击向量和影响[76]. 由于漏洞描述充满了特定于软件的名称、编号规则和特定于域的术语，仅使用通用文本而不是漏洞描述训练的模型性能急剧下降[103]. 然而，手动标记是一项劳动密集型任务. 基于此，Yitagesu 等人[77]提出了一种在漏洞描述文本中标记和提取基于短语的概念的无监督方法，能够准确标记根本原因、攻击方法和影响等信息，很好地用于丰富漏洞信息以及下游分类任务. 然而，漏洞描述信息具有时间属性，一些新的漏洞特征、概念的出现可能会导致概念漂移问题，从而影响训练效果，为此，Le 等人[74]在选择最佳模型上做了一些尝试，提出了一种基于时间的交叉验证方法，以选择用于漏洞评估的最佳机器学习模型.

5.3.2　漏洞类型预测

当前对漏洞类型的分类工作大多都是依据 CWE 标准进行. 2006 年，通用脆弱枚举 CWE 作为安全漏洞分类的标准被美国国家标准技术研究院(NIST)正式提出. 作为一种面向软硬件的系统安全弱点分类准则，其很快被业界广泛接受. CWE 分析和总结了已存在安全漏洞分类标准的不足，吸纳了各个标准的优势，共枚举了数百种漏洞的类型，包含缓冲区溢出、代码注入、认证和授权错误等等，NVD 漏洞库就将 19 种最为常用的 CWE 类型作为其收录漏洞的分类标准. 目前 CWE 分类标准已经被公认为具备权威性和影响力的分类标准，其包含类别被认为是较为客观和合理的类别组成.

作为漏洞条目中的一个重要字段，CWE 可以帮助各个安全角色更好地理解漏洞所属的安全弱点种类及影响，指导开发人员更好地识别和修复漏洞. 因此，对新入库的漏洞及时准确地识别出 CWE 种类是十分重要的. 然而，目前 CWE 类别的生成过程完全是通过手工确定[79,82]，这使得这一字段的生成时间变得不可预测，用户也无法及时根据 CWE 弱点信息准确了解漏洞. 因此，Aghaei 等人[78]首先将自适应分层神经网络应用于 CWE 类型预测任务，并根据文本分析得分和分类误差调整其权重. Wang 等人[79]在此基础上采用 Transformer 的编码—解码框架，首先对 CVE 输入条目进行编码以了解代表性特征，然后对其进行解码以执行有关 CWE 标准的漏洞类型分类，在测试数据集上取得了 90.74%的准确率. Das 等人[80]则通过提出一种基于 Transformer 的框架 V2W-BERT 并结合链接预测和迁移学习，实现了 97%的更高预测准确率. 借助于 V2W-BERT，Panchal 等人[81]也很好地实现了 CVE 的映射和表示，并在此基础上进一步提出了无监督学习算法 SOM，以实现数据压缩并应用于聚类分析，最终实现 CVE 和 CWE 间关系的分析.

5.3.3　影响软件及版本预测

为了进一步完善漏洞信息，NIST 安全专业人员采用了 MITRE 发布的 CVE，并将一个或多个通用平台枚举(CPE)链接到每个 CVE 条目中，以指定哪些软件和版本易受此漏洞影响. 虽然漏洞描述中通常也包括有关于受影响的产品和版本信息，但 CPE 列表的出现促进了此类信息的正式化、标准化

并支持以机器可读的格式提供服务.因此,在漏洞识别和评估自动化时,CPE 是 CVE 信息的重要补充.然而,文献[84,85]的工作都指出,并非 NVD 数据库中维护的所有 CVE 都能正确地链接到 CPE.此外,如 Elbaz 等人[104]所述,从第一次披露 CVE 到将 CPE 添加到漏洞中存在明显的时间滞后,2018 年,正确分配 CPE 元数据的中位时间长达 35 天.因此,对影响软件及版本进行准确收集和识别是十分重要的.由于 NVD 等漏洞库中有关受影响软件版本等信息并不总是以结构化形式提供,因此 Glanz 等人[85]使用了一种基于规则的方法从漏洞描述中提取这些信息. Wåreus 等人[84]通过 NER,从摘要文本中自动构建 CPE 和 CPE 列表,以补充受影响软件和版本信息.

然而,值得关注的是,即使软件信息以结构化的形式呈现,其结果也并非完全准确.围绕该问题,He 等人[83]就以 Linux 内核为例,通过量化 NVD 报告易受攻击版本的错误率来进行了验证.除此以外,他们还发现已有的检测工具通常会对软件的各个版本分别进行检测,而当检测结果相互冲突时,对检测结果的验证将会变得困难.因此,为解决该类挑战,他们使用开发人员日志和补丁来自动识别每个 CVE 真正影响的易受攻击的源代码版本,具体实现为首先将软件的所有版本组织到版本树中,并在版本树主干和分支中识别第一个易受攻击的版本和最后一个易受攻击的版本,从而更好地补全软件和版本信息.

5.3.4 漏洞严重性预测

随着研究人员对软件漏洞认识的不断深入,漏洞库会对漏洞信息进行进一步的分析和完善,同时也会提供严重性等级等漏洞相关信息,这可能会意味着一个不确定的等待过程.因此,许多工作都试图基于漏洞已公布的字段信息来预测其严重性并分配等级.已有研究表明[29],软件漏洞描述中包含了丰富的语义信息,可用于训练机器学习模型,以准确预测软件漏洞的严重性.此外,漏洞描述会随着软件漏洞的发布而公开,更加容易获得,所以对漏洞描述的分析和挖掘一直是软件漏洞严重性预测的重要手段.近些年来也有一些工作在严重性预测任务中结合了对漏洞类型、资产、攻击等的分析,下面将分别进行介绍.

Spanos 等人[87]通过采用文本分析和三种不同的分类方法(决策树、神经网络和支持向量机)实现了准确率为 80%的分类效果,同时也证明了漏洞描述本身是确定漏洞严重性的可靠且高度准确的信息来源.在此基础上,文献[29,68]通过基于漏洞描述训练 TextCNN 和 BiLSTM 分类模型来分别预测严重性.然而,当遇到语料库未包含的输入时,模型效果就无法保证,因此,Li 等人[102]引入了一种基于预训练模型的提示学习方法来预测漏洞特征,进一步提高了严重性预测的性能.在数据集的构建方面,大多数研究[29,68,88,90-91,93,96]均基于 CVSS 的评级结果来进行漏洞数据的严重性标注.除此以外,Malhotra 等人[99]还深入研究了不同的词嵌入方法对严重性预测任务可能造成的影响.

有趣的是,Allodi 等人[105]通过实验发现结合有关资产、攻击和漏洞类型的其他信息能够有助于提高严重性评估的准确性,而关于已知威胁的信息反而会误导评估者并降低漏洞评估的准确性.除此以外,还值得注意的是,漏洞严重性预测的准确率除了取决于特征提取和分类算法,还十分依赖数据标记的质量.尽管 CVSS 是量化漏洞严重性最广泛使用的标准,但 Holm 等人[94]就通过对 384 位安全专家的意见进行调查分析,发现现存 CVSS 对漏洞分级结果的准确性还与漏洞类型息息相关,并对 CVSS 的改进提出了一些建议.与前述基于现有的严重性标记数据进行研究都不同的是,Lee 等人[89]从漏洞利用性的角度出发,认为如果一个漏洞具有和其他漏洞更多的共同特征,则可能会吸引更多的攻击,应该分配更高的严重等级.因此,作者将漏洞严重性评级问题转化为了漏洞相似度检测问题.

5.3.5 小 结

随着漏洞数量的急剧增长,对漏洞进行的分析与字段生成已经成为漏洞库建设人员十分重要和耗时的工作,近些年来机器学习和深度学习技术的发展也大大推进了这一任务的自动化进程.其中,描述信息作为了解漏洞产生环境和机理,以及影响等的最直接和重要的依据,对该字段的生成与补全已经取得了较好的效果.然而,受到机器学习算法应用时概念漂移的影响,并伴随软件的迭代和漏洞数量的急剧增加,现有模型存在预测精度下降的问题,并难以对新出现的术语进行理解和特征表示.如何使模型更加适用于复杂漏洞文本的场景,是一个值得考虑的问题.另外,对于漏洞分类现有工作所采用的 CWE 弱点类型,由于 CWE 只关注安全弱点本身,而忽略了漏洞的具体环境.因此,未来研究还应考虑漏洞被利用时的具体场景,以进行更加细致准确的分类.对于漏洞严重性的研究,接下来的工作还应在

基于描述信息进行预测的基础上，结合更加细粒度的指标，比如缺陷代码等，同时减少无关信息可能对预测造成的干扰.

5.4 漏洞关系的补全与展示

漏洞库虽然包含丰富的漏洞信息，但是数据之间缺乏关联，这就导致很多隐藏信息无法被发现；其次，漏洞信息会由于描述不清晰、一词多义等原因出现歧义；不仅如此，传统漏洞库通过关键字匹配搜索指定漏洞并将漏洞信息以报告的形式展示，这也导致了无法检索包含复杂条件的漏洞并且无法将漏洞间的关系展示给用户. 已有研究已经证明知识图谱可以很好地解决上述存在的问题. 作为语义网的数据支撑，知识图谱以图的形式表示客观世界中存在的概念、实体及其之间的复杂关系，能处理数量庞大的漏洞数据并建立漏洞知识关联[106]，并在此基础上，因其所特有的关系型网络数据结构而具备关系分析以及漏洞数据可视化的能力. 目前有关漏洞关系的研究主要围绕关系补全、语义消歧和知识检索与展示这三个方面，下面将分别进行介绍.

5.4.1 漏洞关系补全

漏洞知识补全是指在发现或者披露一个软件、系统或者网络的安全漏洞后，分析该漏洞相关信息，并推断和补充漏洞之间的关联关系的过程. 目前针对漏洞关系补全的研究主要围绕 CVE、CWE 和 CAPEC 三个数据库. 这些数据库维护的条目可以视为安全实体，“CWE”和“CAPEC”表示特定的弱点或攻击类型，是抽象的安全概念，而 CVE 是特定的安全实例. 这些安全实体之间存在许多的关系，例如〈CAPEC-182，peerof，CAPEC-248〉、〈CWE-697，parentof，CWE-183〉. 但是也有一些隐藏关系没有被发现，比如〈CWE-128，childof，CWE-682〉这组关系在 CWE1.0 版本并不存在，直到 CWE2.0 版本才被添加. 为发现存在于安全实体间的隐藏关系，Xiao 等人[107]将不同数据库的异构软件安全实体整合到一个异构知识图谱中，并利用 TransH 模型将知识图谱中的符号关系和描述性信息嵌入到连续的向量空间中，通过实体在向量空间的距离判断是否存在关系. 与前者相同，Han 等人[108]也选择了基于翻译模型的知识表示学习方法，他们通过 TransE 模型将 CWE 及其在知识图谱中的关系嵌入到语义向量空间，最终实现了 CWE 链接预测，CWE 三重分类和常见后果预测三个推理任务.

上述两种方法均独立处理每个三元组，并没有考虑有关三元组周围相邻实体包含的信息. 为解决这一问题，Yuan 等人[109]提出了一个文本增强的图注意力网络模型，该模型将 2-hop 相邻节点的知识作为附加信息，并采用 TransE 模型作图嵌入，进而从相关安全漏洞库的知识图谱中获取更多的结构和文本信息. 然而，该工作只包含了不同安全实体之间的关联，并没有考虑安全实体描述中含有的信息. 因此 Wang 等人[110]以 SBERT 模型[111]为基础，提出了一种结合 SBERT 和 GAT 的知识图谱表示学习方法，并在漏洞知识图谱上进行了测试，验证了该模型可以很好地学习实体描述中存在的信息，以进行更好的预测.

5.4.2 字段语义消歧

在漏洞描述中存在同一词组具有多种不同的含义或者不同词组具有相同含义的情况，例如，“XSS”和“CWE-79”都是指跨站脚本，“Dirty Cow”则是指发生在写时复制的竞态条件漏洞. 如果没有明确特定术语的含义，可能会导致不同的理解和解决方法. 词性标注可以为句子中的每个词语赋予相应的词性标签，从而准确标记“Dirty Cow”这类术语；而词嵌入可以将词语映射到连续向量空间，以捕捉词语之间的语义关系，使具有相似语义的词在向量空间中距离较近，从而帮助识别“XSS”和“CWE-79”这类具有相同含义的不同词汇.

在词性标注方面，Ye 等人[112]为解决通用词性标注模型无法准确处理漏洞描述文本的问题，提出了一种针对软件领域的特定词性进行标注的方法：将词语按照其在软件上下文中的语义和功能进行分类，通过学习软件领域特定的词性和上下文关系，为软件文本中的单词赋予更准确的词性标签. 然而，由于漏洞描述的复杂性和多样性，通过人工的方式对漏洞进行标注和分类是一项繁琐且容易出错的任务. 因此，Yitagesu 等人[113]提出了一种自动化的词性标注方法，自动为漏洞描述中的单词赋予相应的词性标签，能够更快速地处理漏洞描述文本，并保持一致性和可重复性. 在词嵌入方面，Mumtaz 等人[114]提出了一种针对网络安全漏洞领域的词嵌入的方法：将每个单词映射到向量空间，以捕获单词之间的语义关系，使得相似含义的单词在向量空间中的距离更近. 通过使用学习到的词向量，安全人员可以更准确地分析和解释漏洞描述信息，从而提高漏洞处理效率.

5.4.3 知识检索与展示

漏洞知识检索与展示的目的是提供及时、准确的漏洞信息和漏洞关系并以直观的方式呈现给用

户.由于传统漏洞搜索通常基于关键字匹配或模式识别,因此存在无法准确捕捉复杂漏洞模式和新型攻击技术、需要消耗大量的时间和人力等的局限.为此,Cheng 等人[115]设计了基于知识驱动的漏洞搜索工具 KVS,该工具通过 BERT 模型从已有的漏洞库中提取漏洞知识并对这些知识进行语义分析和关联挖掘,以漏洞知识图谱的形式展示漏洞信息及其关系.在进行漏洞搜索时,KVS 会根据已有知识库进行智能匹配和推荐,提供与输入描述或关键词相关的漏洞信息,从而提高漏洞搜索的质量.虽然该工作简化了漏洞检索过程,但却无法满足安全人员快速查询漏洞特定信息的需求,比如当安全人员想了解漏洞的描述信息时,KVS 只会显示该漏洞的所有字段信息及其关系.Li 等人[116]的研究为解决这一问题提供了一个新的思路,他们设计了一个基于漏洞知识图谱的智能问答系统,该系统采用特征词匹配的方法解析问题语句,识别查询意图,将问题语句转换为相应的 Cypher 语句,在知识图谱中查询,并将结果返回给用户.

5.4.4 小 结

第 5.4 节围绕漏洞关系展开,分别从漏洞关系补全、字段语义消歧和漏洞的搜索与展示三个方面进行总结.在漏洞关系补全方面,从最初利用翻译模型到采用图注意力网络模型,研究人员不仅只处理实体自身的信息,还会考虑相邻实体包含的信息;然而,当前漏洞补全的工作主要针对 CVE、CWE 以及 CAPEC 这三个库,未来研究可以关注于 OVAL、ExploitDB 等更多数据库.在字段语义消歧方面,通过使用适用于特定领域的词性标注模型,已有研究可以很好地解决漏洞描述中存在的歧义问题,从而有助于理解漏洞信息的真实含义.但是现有工作无法很好地处理漏洞描述中所涉及的代码信息,这也是未来研究可以关注的方向.另外,目前围绕漏洞关系搜索与展示方面的研究还相对较少,科研人员通过自然语言处理分析用户的输入并将查询结果返回给用户,然而当用户输入存在歧义时还能否准确理解用户意图,仍是一个有待考虑的问题.

5.5 漏洞库质量评估

漏洞库作为对安全漏洞数据进行收集和发布的单位,可以全面收集并总结漏洞内容.高质量的漏洞库系统有利于为主流厂商和用户提供准确和及时的漏洞情报,还可以帮助政府部门从整体上把握网络安全的发展态势[28,117-118];而漏洞数据作为漏洞库的核心和基础,在漏洞库的建设中发挥着重要作用,因此,对于漏洞库和漏洞数据质量的评价是一项十分有意义的工作.随着数据分析与利用技术的快速发展,近年来,有越来越多的研究者也开始关注此方向,这些研究主要围绕漏洞库整体评价[119-120]以及漏洞数据的完整性、准确性[3,121-126]以及不同库间数据的一致性问题[127-130]展开.下面将分别进行介绍.

5.5.1 漏洞完整性和准确性评估

漏洞数据的完整性和准确性直接关系到基于漏洞的研究质量以及漏洞库的质量[121].Nguyen 等人[3]指出,NVD 中 25%的 Google Chrome CVE 具有不正确的版本信息.Christey 等人[126]同样探讨了 NVD 数据中的问题,并认为报告偏差是其根本原因.Malone 等人[124]发现 NVD 漏洞库中存在严重的补丁缺失问题,他们指出只有大约四分之一的 CVE 条目标记出了补丁,并且报告中存在大量数据输入错误.然而,许多 CVE 都能通过厂商维护的缺陷跟踪平台找到相应补丁.文献[122,125]围绕当前严重性字段质量进行探究,其中,Gallon[122]更加关注 CVSS 评分本身,而 Johnson 等人[125]结合贝叶斯方法对 CVSS 的可靠性及漏洞库应用 CVSS 的效果进行研究,结果发现尽管 CVSS 是可信度很高的漏洞评分机制,但在支持独立进行 CVSS 评分的漏洞库中,OSVDB 的评分效果仍然不理想.值得注意的是,Lyu 等人[123]经过研究发现,与基于 CVSS 评分的严重性指标相比,可利用性仍然缺乏公认的标准.

除此以外,披露的时间延迟也是影响漏洞数据质量的关键因素.与其他类型的数据不同的是,漏洞披露的效率往往会直接影响公众对于漏洞库的依赖程度.因此,Ruohonen 等人[131]调查了从公开邮件列表中分配 CVE 到后来在 NVD 中披露 CVE 之间的历史时间延迟,通过三个维度对该延迟进行建模,并对如何缓解这一问题提出了建议.

5.5.2 漏洞一致性评估

漏洞数据的缺失和不准确会影响后续研究的性能,而数据信息的不一致问题也同样值得关注.比如 Massacci 等人[128]发现,使用不同的数据源可能会导致完全不同的实验结果.目前对于不一致性的研究更多地关注于经过标准化的非自由格式字段,如日期、CWE 类型、CVSS 评级以及受影响软件及版本等.而对于漏洞描述、引用链接等自由格式字段,由于不具有标准化的结构,所以很难定义其不一致性问题.

基于此,Anwar 等人[129]分别对发布日期、厂商及产品名称、严重性评分、漏洞类型等的数据不一致

问题进行探究，证明了漏洞不一致问题的存在性，并表明商业收购及漏洞库管理人员的拼写错误往往会导致软件名称和供应商的不一致，CVSS 版本的不同导致了严重性评分的不一致结果，并且文献给出了纠正这些问题的建议. 文献[130]也同样证实了严重性评分的不一致. 与此同时，Dong 等人[127]也将 NVD 条目中的结构化信息与 CVE 摘要、外部报告等其他信息源的一致性进行了评估，同样发现易受攻击软件的版本信息不一致问题比较普遍，只有 59.82%的漏洞报告和 CVE 摘要与结构化的 NVD 条目完全匹配. 此外，作者也通过案例证实了 NVD 存在高估或低估易受攻击的软件版本问题.

5.5.3　漏洞库整体评价

最后，已有工作[119-120]对于如何建立漏洞库整体的评价体系也做出了一些探索. 虽然目前各厂商及政府都会考虑建设自己的漏洞库系统，但业界仍缺乏一个通用的质量衡量标准和评估方法. Tan 等人[119]指出在不同的漏洞库中，同一漏洞可能具有不同的发布时间和描述数据结构，这些异构数据结构阻碍了标准化建设和数据共享. 他们通过分析当前主流漏洞库数据，分别基于漏洞数量的规模、漏洞描述的独立性、标准化程度和完整性等四个度量指标，提出了一种利用 SCAP 协议及相应标准对漏洞数据库进行量化评估的方法，并依据正态分布规律对漏洞库进行了量化评估. 其结论表明 NVD 和 SCAP 中文社区是综合质量评价中得分最高的漏洞库，其中，NVD，CNNVD 和 SCAP 中文社区为数据标准化程度最高的漏洞库. Li 等人[120]通过系统映射研究探究了目前公开可用的漏洞数据集中的数据质量，并发现 NVD 和 CVE 是最常用的漏洞数据库. 同时他们还发现在文献[119]所指出的质量较高的漏洞库中，CNNVD、SCAP 中文社区等却都不经常被学术界采用.

5.5.4　小　结

本节围绕漏洞库质量评价，分别从评价体系建设和漏洞数据质量评估两个角度展开. 可以看到，目前针对漏洞库整体质量的量化评价仍缺乏全面客观的标准，并且该方向研究相对较少；而漏洞数据质量的评估，近些年来已经逐渐成为该领域研究的热点问题. 客观准确的评价结果将为漏洞库建设者如何改进工作提供重要参考. 本节的内容也可以为漏洞研究者提供参考. 一方面考虑如何更客观地评价数据质量以指导其选用合适的漏洞数据；另一方面，可以围绕现有漏洞数据的不足考虑合适的解决方案. 比如针对目前补丁漏洞库缺失的问题，如何通过将漏洞报告与提供修复的供应商存储库中的代码更改自动关联，并能够大规模应用于漏洞库建设中来缓解. 除此以外，如何对于漏洞描述、引用链接等自由格式字段进行客观的一致性评估也是未来研究工作需要考虑的.

5.6　漏洞库构建工作对比分析

结合上述内容，第 5.6 节对围绕漏洞库构建在各热点方向上研究的整体情况以及所做的改进进行了总结和对比，具体包括文献数量、研究问题和改进角度等三个方面，还对对应的工作进行了罗列，细节如表 2 所示.

表 2　漏洞库构建研究对比

热点方向	文献数	研究问题	改进角度	所对应的工作/模型
漏洞数据获取	25	多源数据获取	数据源拓展与数据检索	Armold 等人[36]、VulData7[46]、Pereira 等人[34]
			漏洞描述自动获取	CVErizer[53]
			多源数据的安全实体标记	Bridges 等人[52]、Li 等人[47]、Chaleshtori 等人[35]
			数据爬取算法改进	Li 等人[47]
		安全缺陷报告预测	多字段特征融合	SBRer[56]
			文本表示算法的改进	Behl 等人[57]、Itactivul[55]、SbrPBert[59]
			分类算法的改进	Zhou 等人[33]、Yang 等人[58]
		补丁数据获取	漏洞修复提交检测	Hermes[37]、SPI[41]、VulFixMiner[61]、VCMatch[64]、Sun 等人[39]、PatchScout[43]、Ponta 等人[45]、Zhou 等人[60]、Wang 等人[63]
			多补丁数据源关联分析	xVDB[38]、PatchDB[40]、CVEfixes[42]
漏洞管理	12	异构漏洞数据融合	多源重复漏洞识别	IFVD[65]、TRACER[50]
			多源漏洞信息融合	Sun 等人[48]、Guo 等人[49]
		漏洞数据维护	漏洞库维护效率	Rodriguez 等人[66]、Gong 等人[68]
			数据更新的完备性	Chan 等人[67]
		漏洞展示与共享	提升检索效率	Tsutsui 等人[69]
			漏洞数据可视化	CVExplorer[70]、Reynolds 等人[71]
			漏洞数据标准化	IVDA[73]、CVML[72]

（续 表）

热点方向	文献数	研究问题	改进角度	所对应的工作/模型
关键字段补全和预测	30	漏洞描述生成	摘要提取	CVErizer[53]、Sun 等人[75]、CaVAE[77]
			补全增强	CVML[72]、OVANA[76]、Guo 等人[49]、Le 等人[74]
		漏洞类型预测	CWE 自动化分类	ThreatZoom[78]、V2W-BERT[80]、Wang 等人[79]、Zhu 等人[82]
			无监督聚类分析	Panchal 等人[81]
		影响软件及版本预测	漏洞描述信息分析	Wåreus 等人[84]、Glanz 等人[85]、Elbaz 等人[104]
			程序补丁和日志分析	He 等人[83]
		漏洞严重性预测	基于 CVSS 和漏洞描述	Han 等人[29]、Gong 等人[68]、Spanos 等人[87]、Nikonov 等人[96]、Wang 等人[88]、Hakan 等人[90]、Li 等人[102]、Malhotra 等人[99]
			基于 CVSS 和多字段特征	AutoCVSS[93]、Allodi 等人[105]、Holm 等人[94]
			CVSS 评分准确性分析	VRSS[92]、Holm 等人[94]、Lee 等人[89]
漏洞关系补全与展示	9	漏洞关系补全	安全实体图嵌入	DeepWeak[108]、text-enhanced GAT[109]、Xiao 等人[107]、Wang 等人[110]
		字段语义消歧	词性标注算法改进	S-POS[112]、Yitagesu 等人[113]
			结合领域知识的词嵌入	Mumtaz 等人[114]
		知识检索与展示	知识推荐	KVS[115]
			智能应答	Li 等人[116]
漏洞库质量评估	12	漏洞完整和准确性	软件版本评价	Nguyen 等人[3]
			补丁信息评价	Malone 等人[124]
			可利用性评价	Lyu 等人[123]
			披露延迟评价	Ruohonen 等人[131]
			严重性评级评价	Johnson 等人[125]、VRSS[92]
		漏洞一致性	数据不一致性度量	VIEM[127]、Croft 等人[130]、Anwar 等人[129]
			对下游任务影响分析	Massacci 等人[128]、Anwar 等人[129]、Croft 等人[130]
		漏洞库整体评价	多指标对比评价	Tan 等人[119]、Li 等人[120]

可以看到，近年来聚焦漏洞数据的获取以及关键字段补全及预测这两个方向产生的研究成果较多. 在漏洞数据获取方面，由于补丁数据具有来源分散、数据量大且难以识别等特点，Hermes[37]、SPI[41]、VulFixMiner[61]等许多工作都围绕漏洞修复提交的识别和多补丁数据源分析展开探索，以开发安全修复补丁的智能化自动识别工具. 除此以外，多源数据的获取还面临着数据检索、安全实体标记等诸多问题，并且，由于漏洞早期通常会以缺陷报告的形式呈现. 因此，近年来已有研究也开始探索如何从大量缺陷报告中识别与安全漏洞相关的报告，以便于及时分析入库. 在关键字段补全及预测方面，越来越多研究注意到了漏洞库建设中各字段信息智能化处理的重要性. 因此围绕漏洞类型和严重性分类、漏洞描述和影响实体预测等方面也产生了较多的研究，尤其是对于漏洞严重性的预测. 由于 CVSS 特征需要进一步评估后逐步更新并且由专家手动评估，因此存在较严重的滞后性. 许多研究都对应地提出了改进方案，可以概括为基于 CVSS 评分结果的机器学习方法改进以及对 CVSS 评分本身的准确性分析. 在此基础上，Malhotra 等人[99]还探究了词嵌入方法对预测结果的影响. 另外，对现有漏洞库质量的评价以及漏洞管理和漏洞关系处理也是近些年来较为热门的研究方向，知识图谱技术的结合为解决漏洞数据间缺乏关联问题提供了很好的思路，并促进了漏洞数据的可视化. 然而，Dong 等人[127]发现并揭露了目前漏洞库所呈现出的质量问题，这也为漏洞管理工作提出了新的挑战. 已有研究围绕漏洞数据的实时更新和标准化等方面均做出了努力.

6 基于漏洞库的应用研究

由于漏洞库中包含了丰富的漏洞特征，并对海量漏洞数据进行了细致准确的分析、评估和分类工作，以报告的形式展示和共享漏洞信息. 因此，这些数量庞大的漏洞报告可以很好地支撑围绕漏洞生命周期的各项研究，如早期的漏洞预测与扫描工作，以发现系统中存在的安全问题，并及时有效地开展针对性的修补工作. 然而，在漏洞被正式修复前，其很可能会被攻击者利用并发起攻击. 因此作为一种主动安全方法，网络攻击建模、安全态势分析等技术能够帮助在潜在攻击发生前后及时预测和处理它们. 除此以外，值得注意的是，随着漏洞数量的急剧增加以及组件式开发的常态化，应用程序组件的漏洞往往难以避免并且可能会导致大量软件面临安全风险. 因此，软件安全性及成分分析等安全保障工作也受到了越来越多人的关注. 在以上研究工作中，漏洞数据均提供了有效的数据支持，推动了机器学习尤

其是深度学习技术在这些方向中的应用，然而对于漏洞报告本身，有些研究也开始关注于围绕其体现出的更多特征所开展的探索性分析，以更好地把握漏洞的发展规律与关系，推进安全漏洞治理，本章将对以上内容分别进行介绍.

6.1　漏洞预测与扫描

漏洞库在漏洞检测和发现领域内的应用主要集中在漏洞扫描和漏洞预测两个方面. 其中，漏洞扫描是指基于漏洞数据库，通过扫描等手段对指定的远程或者本地计算机系统的安全脆弱性进行检测，以发现可利用的已知漏洞的一种安全检测行为. 而漏洞预测则更倾向于基于已披露漏洞的机理和特征，通过分析软件、系统或网络来发现潜在的未知漏洞[132]. 漏洞数据质量往往会直接影响漏洞扫描的效果，同时，随着人工智能产业的兴起，大量机器学习方法也被尝试用于解决软件漏洞预测的问题，下面将分别进行介绍.

6.1.1　漏洞预测

通过梳理近年来该方向的研究工作，我们发现基于漏洞库的预测技术可以分为漏洞挖掘[133-145]和漏洞特征预测[146-149]两个部分. 其中，漏洞挖掘一直是最近的热点和重点部分，而漏洞库在该方向的应用主要体现在基于机器学习的静态漏洞挖掘上. 虽然漏洞库中收集了丰富的漏洞数据，但大多是以漏洞报告的形式呈现，能直接用于表征漏洞代码特征的信息有限. 而源代码相比自然语言更具逻辑性和结构化且具有更高的细粒度，加之漏洞报告中概念漂移等问题的存在，仅基于 NVD 数据很难实现对漏洞的准确预测[133,136]，需要构建专用的漏洞数据集[140-142]. 在算法实现上，Cao 等人[143]所提出的使用二分图神经网络(BGNN)的漏洞检测方法，将源代码转换为可以反映语法和语义特征的不同图，再将图中的这些节点矢量化并输入到 BGNN 进行训练，该模型最终在基于 NVD 和 GitHub 构建的数据集中取得了较高的准确率. 在程序表示方面，文献[140-142,144]也做出了一些改进，使得深度学习在漏洞挖掘工作中发挥出了较好的性能，Nguyen 等人[144]提出了 ReGVD，模型将程序源代码视为平面令牌序列，并使用图神经网络生成源代码的图嵌入，在测试数据集上取得了比之前研究更好的效果. 为了从应用角度对深度学习模型在漏洞检测场景下的能力与价值进行验证，最近，Steenhoek 等人[145]通过实验复现了近年来出现的多个最新的深度学习模型，并详细评估了这些深度学习模型在漏洞检测的性能、鲁棒性和可解释性等方面的表现，推动了本领域最新提出模型的复现工作和进一步研究. 从已有工作可以看出，深度学习模型能够获得超越传统静态分析工具的检测效果，而漏洞类型、程序代码特征、数据集的大小都会对模型效果有着较大的影响. 因此，需要针对不同漏洞类型构建相应的高质量数据集，并且在代码特征表示方面，可以考虑 VulDeePecker[140]、ReGVD、VulCNN[141]等基于不同网络结构的改进. 另外，Steenhoek 等人[145]在对大量模型进行对比时也发现，所有模型都在增加数据量时相应地得到了性能的提升，并且 Liu 等人[137]、Lin 等人[136]及 Croft 等人[135]的研究也均指出，NVD、SARD[134]等漏洞数据集的质量也会直接影响基于深度学习的漏洞检测器的有效性，而现有数据集仍面临缺乏准确标记、规模较小、缺乏可扩展性等诸多问题. 可以看出，在漏洞预测这一具体场景中，仅使用 NVD 等漏洞库数据已无法满足模型和任务的要求，而由于漏洞库具有较高的数据质量和可信度，其在漏洞预测数据集的构建方面仍发挥着不可或缺的重要作用.

不同于漏洞挖掘，漏洞特征预测旨在预测漏洞出现时的特征和规律(比如时间)，从而有助于把握漏洞发展的趋势等. 例如，厂商及安全人员总是期望能在漏洞被攻击者利用前识别并完成修补. 为此，Zhang 等人[146]试图根据 NVD 中的漏洞发现趋势[147]预测特定软件包的下一个漏洞可能出现的时间. 与此不同的是，Last[148]则更关注于历史规律应该如何指导研究者选择最佳的回归模型，以对软件包进行漏洞预测，他们首先通过拟合 NVD 的历史漏洞数据以分析漏洞发现的历史规律，并结合 KNN 和时间序列距离测量为预测选择合适的回归模型，最终得到最佳的时间序列距离度量值集合. 除此以外，Johnson 等人[149]还指出漏洞披露间隔时间也可以作为在给定时间范围内发现零日漏洞可能性的有意义的度量.

6.1.2　漏洞扫描

与漏洞预测相比，扫描技术往往会更加依赖漏洞库的数据支持. 近些年来针对漏洞扫描工具的研究主要集中在对扫描算法的改进[150-152]. 文献[150]首先探究了漏洞库数据对漏洞扫描的影响，Houmz 等人[150]将该影响定义为扫描统计属性随时间的变化，接着通过比较每个漏洞披露前后扫描的时间序列值的分布来证明该影响是真实存在的，并通过训练机器学习模型用于预测新发布的漏洞的影响.

作者还发现 CVSS 值、供应商和产品这几个字段在漏洞扫描中起到了重要作用. O'Hare[153] 提出了一种基于 NVD 等漏洞库数据的漏洞扫描工具 Scout,将 CVE 与 CPE 相关联用于漏洞识别,比已有工具体现出了更优越的性能. Sultan 等人[152] 则在此基础上,考虑如何围绕减少扫描时间和网络流量进行改进,所提出的 CVS 能够支持在大规模网络中的高效部署.

6.1.3 小 结

漏洞库在漏洞预测与扫描研究中的应用比较如表 3 所示. 可以看到,基于漏洞库的漏洞预测与扫描工作最直接的应用就是漏洞扫描与漏洞特征预测,借助漏洞报告中的格式化、标准化信息,可以很好地预测漏洞发现趋势,并在待测系统中对已知漏洞进行扫描. 然而,对于特征预测而言,由于 NVD 等漏洞库中的漏洞报告日期并不代表发现漏洞的日期,这将影响已有研究中对漏洞发现趋势分析的准确性. NVD 中的软件名称和版本等信息也被证明存在不准确[3] 和不一致性[129] 问题. 为此,建议未来研究考虑字段的准确性问题. 除此以外,概念漂移也是基于历史漏洞报告进行研究工作需要解决的问题. 虽然漏洞库中包含了丰富的漏洞数据,但大多是以漏洞报告的形式呈现,能直接用于表征漏洞代码特征的信息有限[154]. 因此近些年来基于机器学习的静态漏洞挖掘研究都普遍使用专用漏洞数据集. 然而,这些数据集的构建都十分依赖 NVD 漏洞库,已有研究也表明,NVD 漏洞库的数据并非完全可靠和全面,因此,建议未来工作可以考虑更多的漏洞数据来源.

表 3 漏洞库在漏洞预测与扫描中的应用比较

特性/维度	漏洞静态检测	漏洞特征预测	漏洞扫描
目标	预测未知漏洞	根据历史数据预测可能出现的新漏洞及特征	确认系统中是否存在已知漏洞
相关工作	文献 [133-145]	文献 [146-149]	文献 [150-153]
应用价值	提供大量已知漏洞信息和相关特征	支撑统计分析漏洞出现时的字段特征	核心的数据支撑来源
局限性	直接描述代码特征的信息有限	漏洞字段可能存在不准确或不一致问题	规模较小,需要确保漏洞的全面性

6.2 漏洞修补技术

漏洞库在漏洞修复工作中同样扮演着极其重要的角色. 首先,作为漏洞披露的主要渠道,漏洞库可以为厂商提供最新的安全情报,跟进漏洞修复状态,并进行及时的披露;其次,漏洞库的字段信息也可以帮助软件开发者了解漏洞机理,更好地开展修复工作;除此以外,漏洞库中记录的包括历史漏洞、补丁等的结构化信息,也可以为自动化程序修复工作提供很好的数据支持,提高修复效率. 另外,对于开源软件用户而言,也需要依赖漏洞库的安全公告来了解最新的漏洞修复并及时应用修复. 目前围绕漏洞报告的漏洞修复技术研究主要集中在修复优先级预测和辅助修复技术两个方面,下面将分别进行介绍.

6.2.1 优先级预测

随着被发现漏洞的增加,安全团队经常被淹没在大量的漏洞告警中,但是安全人员不可能立即处置所有的问题. 如何确定漏洞修复优先级显得尤为重要. 例如,由于很容易会被攻击者利用以获得未经授权的访问并破坏敏感数据,Web 应用程序中的跨站脚本(XSS)或 SQL 注入漏洞可能对应较高的修复优先级,而需要管理员访问权限或仅出现在本地网络中的漏洞则对应较低的优先级[2]. 目前大部分对优先级进行排序的技术都通过漏洞评分来进行,且基于 CVSS 标准实现,而 Sharma 等人[155] 发现,现有的 CVSS 基础分数中存在可利用性和影响分数权重太过固定且影响分数包括的三个指标权重始终相同等问题,使其无法适用于优先级评分场景,因此提出了一个可变影响及利用性权重的评分系统 VIEWSS,并在漏洞库数据中进行了大规模测试验证,结果表明 VIEWSS 的分数分布更能反映正态性. 与此不同的是,Costa 等人[156] 认为关注基于 CVSS 的漏洞严重性评分并不是指导漏洞修复优先级的最佳策略,因为 CVSS 并不考虑漏洞的利用概率[157],例如漏洞具有高严重性但极其复杂且不太可能被利用的情况[158]. 事实上,已有研究也指出在所有已发布的漏洞中只有不到 3% 的漏洞被利用[159]. 接着,Costa 等人基于遗传算法优化的神经网络实现了对利用可能性的测量,以补充其他指标从而确定漏洞的修复优先级,并通过实验证明了方法较 CVSS 策略的有效性. 为了进一步提高修复效率,Jacobs 等人[160] 基于已收集的漏洞利用数据集构建了一个预测漏洞在一个月内被利用概率的机器学习模型 EPSS,并将其应用于漏洞修复优先级的场景,结果指出该研究可以减少修补高危漏洞所需的工作量到基于 CVSS 策略的八分之一. 综合来看,VIEWSS 为 CVSS 标准的扩展提供了思路,而 ETP[157]、V-Rex[156]、EPSS 等则相较而言更加关注漏洞的可利用性. 基于以上提出的 CVSS、VIEWSS

等指标来分析优先级虽然可能是有效的，但忽略了时间成本. 有研究[161]已指出 CVSS 指标的填充延迟正在不断增加，这也就意味着一个公开的漏洞完全可能在其 CVSS 值确定之前就被利用，那么供应商将无法依靠 CVSS 信息确定补丁开发的优先级，系统管理员将无法使用这些信息及时确定补丁安装的优先级. Sharma 等人[162]通过实现了一种仅基于漏洞描述字段信息进行优先级预测的轻量级方法，为考虑时间成本的境况下此类问题的解决提供了一个很好的思路.

6.2.2　辅助修复技术

漏洞修复是一项亟需自动化的困难任务. 最近几年围绕该方向已经产生了两种十分有前景的技术：基于大型代码语言模型(Large Language Models，LLMs)的自动代码完成技术和自动程序修复(APR)[163]. 然而无论是对于哪种技术，都非常依赖准确和充足的数据集，漏洞库在该类数据集构建任务中发挥着重要且无法替代的作用. Fan 等人[164]基于 CVE 库以及相应的源代码存储库收集了大量漏洞信息，包括从 CVE 中获取到的 CVE-ID、严重性评分、漏洞描述、外部引用链接等，以及从 GitHub 存储库中获取到的与漏洞相关的代码更改，以构建涵盖 91 种漏洞类型的 C/C++漏洞修复数据集. 与该方案类似，Wu 等人[163]基于 NVD 和代码存储库构建了 Java 漏洞修复数据集，并在该数据集上分别比较了 LLMs 和基于深度学习的 APR 模型对于 Java 漏洞修复能力. 这样的修复数据集还包括基于 NVD 和其他来源的 Ponta[45]等，漏洞修复任务还十分依赖 CWE 来区分漏洞的类型信息[165]. 除了用于训练模型以进行自动修复，漏洞库中数据还被用来作为补丁正确性验证[166]、识别安全和非安全补丁[167]和安全补丁类型分类[168]等具体任务的数据来源.

除了数据集的构建，漏洞库还被用来支持与漏洞修复相关的探索性研究. 如 Nappa 等人[169]基于 NVD 所确定的受影响的软件及版本探究了共享代码对漏洞修补的影响研究，他们发现共享库或同一程序的多次安装所导致的共享代码会直接影响漏洞修复的效果，并演示了两种该场景下的实际攻击. 又比如 Zhou 等人[170]基于漏洞报告对漏洞修复后每个软件指标的变化进行了跟踪，以指导开发人员能够在软件开发过程中通过评估指标值来了解并编写安全代码. 除此以外，Forootani 等人[171]首先提出了自我修复的概念，即将漏洞交给最初引入它的同一开发人员进行修复，接着基于 NVD 中涉及的 C 和 PHP 项目漏洞分别研究了自修复漏洞在软件项目中的扩散情况、更容易自我修复的漏洞类型，以及与非自我修复漏洞相比，解决自我修复漏洞所需的时间等，以帮助研究人员和从业者确定修复特定漏洞的最佳候选者. 另外，Zhang 等人[172]发现目前学术界尚缺少现有大模型在程序漏洞修复性能上的系统化比较. 因此，通过实验对比从不同方面比较和探讨了基于预训练模型的程序漏洞修复技术的有效性和局限性，例如漏洞 CWE 类型以及大语言模型使用的代码表示，并提出了一种基于迁移学习技术的程序漏洞修复方法. 最近，基于 CodeT5 预训练语言模型，Fu 等人[173]也构建了一个 T5 架构的 VulRepair 模型，该模型可以生成有效的向量表示，并取得较好的程序修复效果.

6.2.3　小　结

漏洞库在漏洞修补研究中的应用比较如表 4 所示. 应用主要体现在优先级预测和辅助修复技术两个方面. 从已有工作中可以看到，与缺陷优先级预测时考虑更多的因素比如缺陷的修复时间、开发人员特征等相比，对于漏洞的优先级研究主要还是基于漏洞自身的固有属性进行，尤其是大多数研究都基于 CVSS 进行修改或完善. 而由于 CVSS 并未考虑漏洞的利用概率等，因此，未来对于优先级的预测研究应该考虑更多与利用性相关的因素以及非漏洞固有属性的因素，因为优先级与严重性相比更加面向解决漏洞的开发人员. 另外，漏洞库中数据更新存在延迟，这也是优先级预测研究当前仍然面临的问题. 而围绕漏洞修复技术，尤其是基于大型代码语言模型的自动代码完成技术，由于已有很多研究表明 NVD 所涉及的漏洞修复信息并非全面和准确的，因此未来研究可以考虑在训练语料中结合更多漏洞库的信息.

表 4　漏洞库在漏洞修补中的应用比较

特性/维度	优先级预测	辅助修复技术
目标	确定最优修复顺序	提高修复速度和效率
相关工作	文献[155-162]	文献[163-173]
应用价值	核心的数据支撑来源	提供源码补丁等数据及索引
局限性	数据更新存在延迟	描述修复特征的信息有限

6.3　软件安全性及成分分析

漏洞库在软件安全性及成分分析领域的应用主要围绕软件组成分析与软件安全评估两个方面展开. 在软件组成分析中，主要关注从整体上理解并管理软件供应链的安全风险，而软件安全评估则更侧

重于针对已部署或正在开发的具体软件产品进行深度安全审查和分析,两者虽有交集,但侧重点不同,分别服务于不同的安全实践环节.漏洞库可以作为软件组成分析过程中软件库提取的初始数据源并被用于实现软件安全评估方法的构建及验证,下面将对两方面研究分别进行介绍.

6.3.1 软件组成分析

大型软件的开发过程中不可避免地会依赖各种软件库,而当其中某个软件库出现漏洞时,如果不及时发现并采取相应措施,可能导致用户数据泄露以及给软件厂商带来经济损失[174]等.比如在 Log4j 漏洞事件中[1],Apache 的日志库 Log4j 被发现漏洞,这严重影响了使用该库的所有软件系统,因此,软件组成分析技术受到了学术界的广泛关注,它主要解决的问题是如何结合漏洞报告等信息识别提取软件库,现有研究也围绕该问题展开了一系列探索.Chen 等人[175]将 Fast 极端多标签学习模型应用于软件组成分析,首次将 NVD 数据中的库名称识别提取保存为 XML 格式并将该方法部署到生产环境中取得了较好的评估效果.在此基础上,Haryono 等人[176]评估了已有方案中使用的多种极端多标签学习模型,发现 Bonsai 与 Parabel 模型的性能优于 FastXML 模型.但 Lyu 等人[177]发现已有研究均忽略了漏洞数据中存在的漏洞发现时间顺序问题,因此,他们提出了考虑漏洞报告时间顺序的极端多标签学习算法进行组件库的扫描提取,提升了自动提取组件库的效果.最近,Zhao 等人[178]发现,随着软件功能的复杂性不断增加,软件组成分析技术在依赖关系解析过程中可能会遇到不同的依赖项导入和不同的依赖关系规范等各种复杂场景,然而目前仍缺乏对考虑复杂场景时针对 Java 的软件组成分析工具的全面评估.为此,作者提出了一种由扫描模式、扫描方法和 Maven 组合分析范围组成的评估模型,用于全面评估组合分析工具的依赖关系解析能力和有效性.

6.3.2 软件安全评估

除了软件组成分析,漏洞库在软件安全评估方面也取得了较好的应用.首先,漏洞库可以支持软件安全评估方法的验证.Zhang 等人[179]通过实时提取待评估软件中现有用户的使用经验信息来构建评估模型以预测软件安全性,并基于 CVE 与 NVD 中数据进行了验证,证明了该方法的有效性.漏洞数据还可以用于评估过程的对比分析,文献[180]通过提取软件的漏洞特征并将其与 NVD 漏洞库数据进行对照研究,计算了软件产品和供应商的漏洞分布得分以进行安全评估,分析出了包含漏洞较多的软件产品.Gao 等人[181]还将漏洞库应用于软件安全评估方法的构建上,他们基于 NVD 数据构建了一个本体攻击模型,用其对运行软件的安全性进行评估并验证了方法的可行性.此外,文献[104]注意到漏洞信息不完整可能会影响评估效果,进而导致现有方法不能很好地应用于新漏洞.为此,他们通过对新发布漏洞的特征进行提取以及时评估受影响的软件,提高了应对风险的能力.

6.3.3 小 结

漏洞库在软件组成分析和软件安全评估这两个方面的应用比较如表 5 所示.可以看到漏洞库在其中均起到了重要作用,但也存在着数据质量和完整性、概念漂移、依赖关系复杂性等方面的局限性.对于软件组成分析,由于漏洞库通常仅提供单个组件级别的漏洞信息,而软件组成的复杂性意味着组件间相互依赖关系可能并未充分反映在漏洞库中.因此在实际评估时需要考虑更复杂的系统层面风险.而对于软件安全评估,由于名称和版本信息等存在不准确和不一致的情况,这种数据缺陷也会影响基于历史记录进行趋势分析的准确性,也限制了漏洞预测模型的效果,以上局限性在漏洞库建设中也值得考虑和完善.

表 5 漏洞库在软件安全分析中的应用比较

特性/维度	软件组成分析	软件安全评估
目标	减小软件供应链安全风险	减轻目标产品漏洞威胁
相关工作	文献[175-178]	文献[104,179-181]
应用价值	与识别的软件成分对比	提供漏洞及 CVSS 等
局限性	组件依赖未充分反映	字段不准确、概念漂移

另外,现有研究讨论了从漏洞报告中提取软件库所面临的问题并提出了有效的解决办法.然而相比于关注基于历史漏洞报告的软件库提取,由于新漏洞中包含的信息不全但也涉及软件库的相关信息,未来研究可以更多关注对于新发布的漏洞如何进行有效的软件组成分析.

6.4 网络攻击建模技术

网络攻击建模技术将网络系统中的不同组成部分(例如主机、网络设备、应用程序等)和与其相关的漏洞、攻击模式、攻击行为进行建模和表示,进而描述网络攻击的过程,帮助安全团队分析系统中存在的漏洞,并提供有效的决策支持.而漏洞库则为该工作提供了数据支撑.已有研究通过网络攻击建模技术实现了预测攻击路径[182-184]以及预测攻击影

响[185-186]等.目前围绕漏洞库在网络攻击建模技术应用的研究主要集中在攻击模式识别[187-190]和攻击路径分析[191-197]两个方面,下面将分别进行介绍.

6.4.1　攻击模式识别

漏洞攻击模式识别是指通过分析已知漏洞信息和攻击样本,将漏洞与攻击模式或攻击方法关联的过程.CVE 记录了大量已披露的软件和硬件漏洞,CAPEC 记录了攻击模式相关的内容,然而它们之间的信息并没有直接联系,因此 Dang 等人[187]首先聚焦于网络功能虚拟化(NFV)和软件定义网络(SDN)相关的漏洞,将 CVE、CWE 和 CAPEC 的关系转换为图的形式后通过链接预测的方法实现了 CVE 与 CAPEC 的关联,从而更好地理解攻击行为.与该工作不同,Kanakogi 等人[188-189]通过相似性度量的方法追踪与 CVE 相关的 CAPEC,他们利用 TF-IDF 分析了 CVE 漏洞描述和 CAPEC 攻击模式描述之间的相似性并成功追踪了 48 个 CVE 的攻击模式关系.虽然通过 CAPEC 可以更清晰地了解攻击所需的权限、资源等信息,但是却无法了解攻击者攻击时可能会采用的方法,而 ATT&CK 则是一个记录攻击技术和战术的威胁情报库.Grigorescu 等人[190]使用 BERT 模型对 CVE 描述和 ATT&CK 技术描述进行了编码,并计算它们之间的相似度以实现 CVE 与 ATT&CK 的映射,从而获得更准确、全面的威胁情报,以帮助安全人员进行全面的风险评估并制定防御策略.

6.4.2　攻击路径分析

尽管通过漏洞攻击模式识别将漏洞与攻击类型关联可以获取有关漏洞的攻击信息,但仍无法直观展示攻击者可能的行动路径.已有研究通过引入多种模型来解决这一问题.例如,杀伤链模型可以概述攻击者从侦察到最终实现目标的整个过程,为安全专家提供了攻击逐步发展的视角;STRIDE 模型辅助于识别与安全目标相关的六大威胁类别,从而揭示系统的潜在薄弱环节;而攻击图模型是一种基于图论的建模技术,可以通过图的形式呈现攻击者在系统中可能的攻击路径,从而展示攻击者如何利用系统中的漏洞和弱点逐步渗透攻击系统的过程.

相比于其他模型,攻击图的可视化特性有助于更清楚地分析复杂网络中潜在的安全风险,然而,由于近年来漏洞数量增长迅速,通过手工标记生成攻击图十分耗时且容易出错,因此如何自动生成攻击图是最近的热点方向.Aksu 等人[191]从 NVD 中提取漏洞数据,通过分析漏洞数据中的关联信息构建攻击图的节点和边,并利用路径搜索算法生成潜在的攻击路径.然而该工作无法将新漏洞添加到已生成的攻击图中,为解决攻击图的不可拓展问题,Bezawada 等人[193]设计了一种名为 AGBuilder 的攻击图自动生成工具,该工具不仅可以增量更新攻击图,还可以将多个较小的攻击图聚合成一个较大攻击图.然而在扩展或者聚合攻击图时,可能会产生冗余节点和冗余攻击路径,从而降低攻击图的可读性和理解性.为了解决这一问题,Yu 等人[194]利用 NVD 和 Bugtraq 的漏洞及其相关信息构建了原子攻击数据库,并通过匹配攻击数据库中的原子攻击生成潜在的攻击路径.该方法可以识别攻击图中的关键节点且有效减少冗余节点和路径,从而有助于找到攻击的关键路径.掌握攻击的关键路径不仅可以帮助安全人员更全面地分析已存在的漏洞,还有助于预测未知风险.例如,Liu 等人[184]通过分析攻击图中的关键节点构建攻击路径预测模型,而 Keramati[195]则通过分析攻击图中的攻击路径评估零日攻击的整体风险水平.

然而,Sadlek 等人[197]发现,攻击图通常没有标准化的攻击步骤表示,且无法直接映射到杀伤链的各阶段.因此在此基础上,结合杀伤链提出了一种新的攻击图类型—杀伤链攻击图(KCAG),并结合 STRIDE 安全属性对资产进行分类,以便更好地了解攻击者在实现其目标过程中可能采取的行动序列及其对受保护基础设施的影响.Florian 等人[198]还注意到,有关过去攻击的网络威胁情报可以通过帮助深入了解攻击者使用的工具和攻击模式来更好地重建攻击并预测正在进行的攻击过程.基于此,结合多个威胁情报源数据,作者首先提出了多级威胁知识库 AttackDB,接着提出了一种基于知识图遍历算法和各种链接预测方法的攻击假设生成器,该生成器能够自动推断 ATT&CK 技术,帮助分析师提高攻击假设的准确性并自动化攻击假设生成过程.

6.4.3　小　结

本节围绕漏洞库在网络攻击建模技术领域的应用展开讨论,主要分为攻击模式识别和攻击路径分析.对于这两方面的研究比较如表 6 所示,由于新出现的漏洞可能会因披露延迟等而未被及时收录,这在一定程度上限制了其在预测未来攻击趋势上的作用.不仅如此,漏洞报告中的信息可能存在不完整、不准确的问题,导致攻击模式识别及路径分析的准确性受限.另外,不同的漏洞报告格式、分类标准和

攻击模型之间可能存在差异，使得从漏洞库中提取有效信息用于攻击模式识别和路径分析时往往依赖额外的转换和解析工作，这也增加了研究和实践的复杂度.

表 6 漏洞库在网络攻击建模中的应用比较

特性/维度	攻击模式识别	攻击路径分析
目标	关联漏洞与攻击模式/方法	构建和解析攻击路径
相关工作	文献[187-190]	文献[191-197]
应用价值	为关联映射提供数据支持	支持攻击路径构建分析
局限性	更新延迟及库间格式差异	字段不准确或不一致

尽管如此，由于漏洞库中存储了大量的漏洞数据以及与攻击相关的信息，建立漏洞与攻击模式的关联关系可以帮助安全人员更全面地分析攻击产生的原因，从而更有针对性地采取防御措施. 虽然当前有很多研究关注漏洞与攻击模式关系映射的问题，但是只考虑了漏洞与攻击的描述，并没有分析其他字段信息，导致不能准确映射漏洞与攻击模式的关系，因此如何提升模型准确率是未来急需解决的问题. 同时，这些漏洞数据和攻击信息也为攻击图相关研究提供了高质量的数据支撑，现有围绕生成攻击图的研究虽然解决了攻击图不可拓展的问题，但是无法根据系统的状态生成特定攻击图，未来研究可以考虑如何感知系统状态并随着系统的变化实时更新攻击图.

6.5 安全态势分析

随着网络攻击技术的不断革新，如何更好地预测和把握漏洞的整体发展态势和提高现有安全技术对安全状态的感知、判断能力，对保障网络安全起着重要作用. 本节将漏洞库在安全态势分析领域的研究分为漏洞演变趋势分析和网络安全态势感知这两个方向.

6.5.1 漏洞演变趋势分析

了解漏洞整体演变趋势是安全风险管理过程的重要环节[199]，漏洞库在该类型工作中发挥着重要作用. Murtaza 等人[200]基于 NVD 中的漏洞数据对各软件产品在一段时间内漏洞数量的变化趋势进行了分析，发现 SQL 注入漏洞在过去几年有所减少，而加密漏洞出现了显著增加的趋势. 除此以外，他们还发现许多漏洞的产生主要源于其他更早期漏洞的出现，比如缓冲区错误漏洞可能会导致未授权访问漏洞，因此还探究了软件漏洞间的常见关系模式. 另外，他们指出漏洞的关系模式可以用来评估软件产品未来可能会出现的漏洞，比如历史上经常同时出现的两个漏洞在未来也可能会一起出现，这种顺序模式可以从一定程度上帮助避免再次受到相同的威胁. 在对 NVD 中漏洞演变趋势进行分析时，Tang 等人[201]首先发现了波动率聚类效应(ARCH)的存在，并基于复合模型提出了相应的解决方案. Williams 等人[202]通过监督主题演化模型(STEM)探究了 NVD 收录的漏洞是如何随时间演变的并检测了特定网络安全威胁的演变. Chang 等人[203]则对 15 种漏洞类型的出现频率、严重性等指标的变化趋势进行了分析，发现所有类型漏洞的频率都在逐渐下降，呈现出积极的趋势，且高严重性漏洞的频率也在下降. 相较于其他类型漏洞，并发漏洞受到线程调度不确定性的影响而更具隐藏性和延迟性，因此，Bo 等人[204]对在 CVE 中提取的并发漏洞的趋势、影响等进行了分析，发现有近一半的并发漏洞可以被远程利用，且并发漏洞披露的数量总体呈逐年上升趋势.

6.5.2 网络安全态势感知

态势感知是以安全大数据为基础，从全局视角提升对安全威胁的发现识别、理解分析、响应处置能力的一种方式，可以更好地加强纵深防御. 通过建设主动防御、持续监测、应急响应、溯源取证、风险预警等安全能力，最终实现安全运营等闭环管理[205]. 随着人工智能技术的发展，NVD、CVE 等漏洞库和 ATT&CK 等攻击行为库在智能态势感知领域发挥着越来越重要的作用[206]. 近年来，研究人员围绕网络安全态势感知进行了大量研究，Endsley[207]提出了最经典的态势感知模型，该模型将态势感知分为态势感知、态势理解、态势预测三个阶段. Chen 等人[208]利用回归预测的思想来预测潜在的攻击，并通过提高预测精度来提高态势感知的性能. 为了能够准确感知 APT(Advanced Persistent Threat)攻击下的网络安全态势，Chen 等人[206]首先基于漏洞库等安全知识库构建了 APT 攻击的知识图谱，接着构建了态势感知模型，以允许从潜在威胁中评估 APT 的网络态势. Kou 等人[209]则提出了一种基于攻击意图识别的态势评估方法，首先对攻击事件进行因果分析，简化并识别出各个攻击阶段，进而实现态势评估，该研究需要结合漏洞库来识别各个攻击阶段以及攻击者的攻击意图.

6.5.3 小 结

漏洞库在安全态势分析研究中的应用比较如表7所示，漏洞库作为关键的数据源，对于掌握漏洞演变趋势，理解当前网络环境的安全状态和潜在威胁都具有重要意义；然而其应用也同时存在一些局

限性，比如库中数据主要关注通用漏洞特征，但在特定网络环境下，某些漏洞的影响程度可能会因为具体设备配置、网络架构等因素而有所不同，因此单纯依赖漏洞库信息难以充分把握特定组织或系统的实际安全状况等等.

表 7　漏洞库在安全态势分析中的应用比较

特性/维度	漏洞演变趋势分析	网络安全态势感知
目标	预测漏洞整体演变趋势	整合多源情报并评估风险
相关工作	文献[199-204]	文献[205-209]
应用价值	核心的数据支撑来源	与其他情报资源整合分析
局限性	数据不完整且更新延迟	漏洞影响与实际环境相关

在研究现状方面，尽管漏洞库中记录了主要软件系统丰富的漏洞信息，但由于缺乏支持漏洞广泛分析的工具和算法，如此庞大的结构化数据集在很大程度上仍然被忽视. 因此，对漏洞相互作用、趋势和演变等方面的分析仍然较少. 了解漏洞趋势将为研究人员和厂商开发更安全的产品铺平道路，减轻现有漏洞的影响，并指导网络安全领域的新兴研究. 因此，未来围绕漏洞库的研究可以更多关注该方向. 除此以外，目前对于大数据技术在网络安全态势感知场景下的应用研究尚处于起步阶段，如何更好地结合用户异常行为等更多网络数据并分析数据间相关性，以支持更准确的态势预测也是值得进一步研究的方向.

6.6　漏洞特征的规律及关联性挖掘

已有研究基于漏洞库数据，除了关注漏洞预测、修复、利用与攻击以及软件和供应链安全以外，也试图围绕漏洞报告所体现出的更多特征进行探索性分析[210]，从而帮助安全分析人员更好地分析和修补漏洞. 其中一些研究还能有助于完善漏洞库本身的建设，比如关注社交媒体讨论对漏洞披露的作用[211]、漏洞披露和报告者的关系[212]以及漏洞严重性和漏洞赏金间的关系[213]等等. 本节将按照特征分析的对象不同，分为单个特征规律发现和多特征间关联分析这两个部分展开介绍.

6.6.1　特征规律发现

针对所关注的漏洞特征，单独分析某一个或某几个漏洞报告可能无法发现漏洞数据所呈现出的规律性特点. 比如对于漏洞的报告者，漏洞库使用人员往往并不太关注，而 Alexopoulos 等人[212]首次进行了围绕漏洞报告者所呈现出规律的研究，通过对 FLOSS 项目中报告漏洞的人员和组织进行了大规模的实证研究后他们发现，有大约 80%漏洞报告仅来自 20%的报告者，即少数报告者报告了大部分漏洞，且其余大部分报告者仅报告了少部分漏洞. 并且随着时间的推移，漏洞报告的数量与报告者的数量相关. 另外，作者还发现有大约一半的报告者都表明了其隶属关系，且每年也会产生很多的首次报告者，该研究可以帮助指导关于软件项目的安全决策，以吸引更多的漏洞研究人员参与. 除此以外，Votipka 等人[214]的研究也表明，对于软件测试人员和白帽黑客这两大漏洞发现主体，漏洞赏金不一定是鼓励他们发现漏洞的全部动机，因此建议为了鼓励黑客的参与也可以引入一些非奖金激励.

又比如在舆论对漏洞的讨论热度方面，Alperin 等人[211]首先指出 NVD 中的许多漏洞在发布的几个月前就已经在 Twitter 和 Reddit 等社交媒体平台上被公开讨论了. 因此，提出了一个无监督框架以自动过滤出 Twitter 数据集中与安全相关的推文，并发现来自推文的信息可以有助于漏洞库专家对新入库的漏洞进行风险评估等安全分析. 在对漏洞严重性的统计分析过程中，Li 等人[215]探究了环境指标的分布以及其可能对漏洞严重性产生的影响，发现对于任何漏洞其环境指标都存在一个模式值，可以帮助 NVD 进行更全面的 CVSS 评级. 另外，Kudjo 等人[216]基于加权移动窗口[217]构建了漏洞预测模型，结合了漏洞数据中的时间顺序，以探究漏洞发现的年度季节性趋势.

在漏洞特征规律发现方面，Shahzad 等人[218]分别从漏洞风险级别、软件供应商、漏洞利用的访问要求以及漏洞多年来的演变等八个维度对漏洞库中的大量数据进行了探索性分析研究，具体发现包括漏洞的披露率近些年来已不再呈指数级增长、已发布漏洞的主要漏洞类型会随着时间变化、开源软件漏洞较闭源软件往往更快被利用而修补速度更慢等等. 然而有趣的是，我们发现在漏洞修补速度方面，该研究结果与 Arora 等人[219]的结论截然相反，后者认为开源供应商往往会比闭源供应商更快地提供补丁. 我们对造成该现象的原因进行了分析，发现研究时间的较大差异可能是导致结论不同的一个重要原因，Arora 等人[219]的研究发表于 2004 年，较文献[218]早了十五年，而随着漏洞规模不断扩大、软件复杂度迅速增加以及漏洞披露进程的加快，闭源供应商更加重视漏洞修补和补丁发布的迅速反应[219]，同时，也拥有更集中的资源来尽快修复新披露的修复[218]. 与此不同的是，漏洞披露进程的变化对开源供应商的影响相对较小[219]，且开源产品漏洞的修复工作更依赖自由开发人员的贡献[218]. 因

此,近年来相比于闭源产品存在滞后.总之,这些探索性分析能够有助于未来制定更有效的安全策略和评估特定供应商的产品风险[177,220].

6.6.2 特征间关联性分析

漏洞报告中通常包含有关漏洞丰富的描述信息,并通过半非结构化的字段信息向用户呈现.然而这些字段间并非完全独立的,比如一个高危且影响范围较大的漏洞,其可能包含较多的外部引用索引.因此,可以通过对不同字段间的特征进行关联性探究,来挖掘漏洞库中各字段间的相关性,以及识别漏洞生命周期中所体现出的隐藏特征,比如漏洞发现策略等.

为了识别漏洞的发现策略,Bhuiyan 等人[221]对 NVD 漏洞库中的漏洞及其索引到的软件缺陷报告进行实证研究,即分别通过基于文本特征和正则表达式的方法自动识别包含漏洞发现策略的报告.发现其选择的三个测试项目的缺陷报告均涉及漏洞的发现策略信息,该发现为漏洞库维护人员在分析漏洞时应使用开源软件缺陷报告提供了实证基础.同时,影响漏洞利用的因素也是个值得探究的主题,由于某些网络攻击会同时利用多个漏洞,且这些漏洞很有可能都来自同一供应商的产品,为了探究产品和漏洞的关系,Tsutsui 等人[69]分析了 NVD 中近五年的漏洞报告,发现有超过一半的产品存在多个漏洞且其中 47%的产品都至少存在一个高危漏洞,并且由于多个产品可能来自同一个供应商,可能会共有相同的高危漏洞.在漏洞修复方面,Woo 等人[222]探究了补丁的产生效率和软件及版本的关系,首先给出了“零漏洞”的概念,即漏洞最先产生的软件,接着他们对大量漏洞进行跟踪发现错误的零漏洞可能导致补丁更新时间延长.为解决此类问题,他们提出了一个能精确发现零漏洞(包括软件名称和版本)的机制 VOFinder,可以通过识别漏洞软件之间的复用关系来明确漏洞的传播方向.Xiong 等人[223]探讨了漏洞信息披露与软件厂商补丁研发之间的关系,结果表明,第三方共享平台的漏洞披露可以提高软件厂商的补丁研发概率.不仅如此,信息处理需求(例如漏洞信息的关注度、漏洞评分、是否提前披露漏洞等)均可以加速漏洞补丁的研发,然而,受漏洞影响的产品数量和软件厂商的软件版权数量对补丁研发没有显著影响.另外,在影响漏洞发布时延因素方面,Ruohonen 等人[224]以 Windows、openSUSE 和 Ubuntu Linux 为例探究并证明了软件产品的年龄不会影响安全公告发布和 CVE 发布的具体时间延迟,在漏洞类型方面,Lin 等人[225]通过使用关联规则挖掘算法对漏洞属性之间的关联规则进行分析,以查找漏洞描述库 NVD,弱点枚举库 CWE 之间的相关性.

6.6.3 小 结

围绕漏洞生命周期,漏洞库数据除了在预测、利用、修复等方向为研究者提供了准确和全面的数据支持.由于其还记录了漏洞在整个阶段的关键信息,如提交记录、发布日期、影响、修复信息等,因此,也支持通过探索性分析发现漏洞特征的统计规律,从而帮助安全分析人员更好地分析和修补漏洞,规避安全风险.漏洞库在特征与关系挖掘中的应用比较如表 8 所示,可以看到,漏洞库在其中都可以作为核心数据来源,扮演着重要角色.

表 8 漏洞库在特征与关系挖掘中的应用比较

特性/维度	特征规律发现	特征间关联性分析
目标	围绕漏洞报告所体现出的更多特征进行探索性分析	研究漏洞数据之间的内在联系和相互作用
相关工作	文献[211,212,214-220]	文献[69,213,221-225]
应用价值	可作为核心的数据支撑来源	可作为核心的数据支撑来源
局限性	规模较小,需要确保漏洞的全面性,概念漂移	存在数据不完整、更新滞后及字段信息不准确问题

虽然库中包含大量数据,但如何利用这些信息来发现未知并且有价值的模式和关系,也是我们要解决的问题.从对已有研究的调研来看,该方向研究起步相对较晚,且披露日期、严重性评级、漏洞报告者和受影响产品及版本等为当前研究重点关注的特征,而针对其他特征如报告修订记录、漏洞类型等还缺乏统计分析.另外,现有研究大多过于关注 NVD 漏洞库而未考虑其他漏洞库数据,已有研究已经指出,NVD 并非完全准确和全面的数据来源,因此,未来工作可以考虑结合更多漏洞库如 ExploitDB、IBM X-Force Exchange 等展开.

6.7 漏洞库应用工作对比分析

结合以上内容,本节对于围绕漏洞库应用的相关热点研究进行了整理和对比,具体包括文献数量、研究问题和各方向研究所依赖的漏洞数据基础,细节如表 9 所示.

表 9　漏洞库应用研究对比

热点方向	文献数	研究问题	所需漏洞信息/字段	所对应的工作/模型
漏洞预测与扫描	21	漏洞静态检测	标识号、类型、代码、产品/版本等	ReGVD[144]、VulDeePecker[140]、BGNN4VD[143]、VulDeBERT[142]等
		漏洞特征预测	标识号、严重性、描述、类型、产品/版本、发布/更新时间等	Zhang 等人[146]、Last[148]、Johnson 等人[149]等
		漏洞扫描	标识号、严重性、描述、类型、产品/版本、发布/更新时间等	Houmz 等人[150]、O'Hare[153]、Sultan 等人[152]等
漏洞修补	19	优先级预测	漏洞描述、严重性、类型、产品/版本等	VIEWSS[155]、ETP[157]、V-Rex[156]、EPSS[160]等
		辅助修复技术	标识号、严重性、描述、类型、代码、产品/版本、参考链接等	Fu 等人[173]、Nappa 等人[169]、Fan 等人[164]、Wu 等人[163]等
软件安全分析	8	软件组成分析	标识号、描述、产品/版本、发布/更新时间、代码、参考链接等	Chen 等人[175]、Haryono 等人[176]、Lyu 等人[177]、Zhao 等人[178]
		软件安全评估	标识号、严重性、描述、类型、代码、产品/版本、参考链接等	Zhang 等人[179]、Gao 等人[181]、Rasheed[180]、Elbaz 等人[104]
网络攻击建模技术	11	攻击模式识别	标识号、严重性、描述、利用信息、类型、产品/版本等	Dang 等人[187]、Kanakogi 等人[188-189]、Grigorescu 等人[190]等
		攻击路径分析	标识号、描述、利用信息、类型、产品/版本等	Aksu 等人[191]、Bezawada 等人[193]、Yu 等人[194]、Florian 等人[198]等
安全态势分析	11	漏洞演变趋势分析	标识号、严重性、描述、类型、产品/版本、发布/更新时间等	Murtaza 等人[200]、Tang 等人[201]、Williams 等人[202]、Chang 等人[203]等
		网络安全态势感知	标识号、严重性、描述、利用信息、类型、产品/版本、发布/更新时间、参考链接等	Chen 等人[208]、Endsley[207]、Chen 等人[206]、Kou 等人[209]等
特征与关系挖掘	16	特征规律发现	严重性、描述、利用信息、类型、产品/版本、发布/更新时间、参考链接、提交者等	Alexopoulos 等人[212]、Alperin 等人[211]、Votipka 等人[214]、Li 等人[215]等
		特征间关联性分析	严重性、描述、利用信息、类型、产品/版本、发布/更新时间、参考链接等	Bhuiyan 等人[221]、Tsutsui 等人[69]、Xiong 等人[223]、Ruohonen 等人[224]等

从文献数量上看，由于漏洞发现和修补始终作为安全漏洞生命周期中十分重要的节点，并且逐步建设完善的漏洞库能够为其研究提供大量数据支持. 因此近年来围绕这两个方向的研究居多，尤其是对于漏洞静态挖掘和自动修复这两类问题. 而不论是对于何种方向的研究，所依赖的漏洞数据都往往离不开漏洞所影响的产品及版本信息. 此外，漏洞描述、类型、严重性及漏洞标识也已经成为大多数研究所依赖的字段. 同样，在漏洞库构建的相关研究工作中，前面四种字段信息的补全及预测也是对应的热门研究方向，这也凸显了以上漏洞字段在漏洞报告中的重要性. 然而不同的研究方向所依赖的数据信息仍然会有所差异，比如网络攻击建模技术研究就需要使用大量漏洞利用信息，对安全态势的分析工作还需要了解漏洞的时间特征等等. 总的来说，目前绝大多数大数据驱动的漏洞研究都离不开漏洞库的支持. 尽管在一些漏洞检测、修复等的具体任务场景中仅使用 NVD 等漏洞库数据无法完全满足模型和任务的要求，需要构建专用数据集，如 SARD[134]、Big-Vul[164]等，而由于漏洞库具有较高的数据质量和可信度，其在漏洞预测数据集的构建方面仍发挥着不可或缺的重要作用.

7　研究挑战及未来方向

作为信息安全基础设施中的重要一环，漏洞库保存了漏洞的属性、特征及解决方案等，并且能够作为一个信息披露与共享的渠道，为用户及产品厂商提供及时且标准化、可定制化的安全服务，其海量漏洞数据也可以作为围绕漏洞生命周期开展科研工作的重要支撑. 因此漏洞库的建设和完善是十分有意义的，本文旨在通过对已有研究工作的介绍，帮助围绕漏洞及漏洞库的研究人员在掌握研究现状的基础上，了解如何更好地完善漏洞库建设，并且如何基于漏洞数据开展一系列前沿和富有探索性的研究. 近年来随着人工智能和大数据技术的快速发展，通过智能化地处理漏洞信息来辅助安全漏洞研究成为了热点问题. 然而目前该领域仍然存在诸多不足和挑战，值得大家仔细思考，表 10 列出了现有研究存在的一些问题以及可能的解决方法.

(1) 如何获取更加全面的数据并解决多源异构数据融合问题

受厂商对产品维护方式和漏洞发现者主观因素等的影响，新产生的漏洞往往可能散布在软件缺陷

表 10 研究面临的问题与机遇

挑战	机遇
早期漏洞数据获取困难	跨项目安全性缺陷报告预测
多源异构数据融合问题	结合除漏洞标识以外的更多特征
漏洞数据间关系的缺失	考虑关联更多库以进行关系补全
漏洞描述等存在歧义问题	考虑消歧处理时结合代码逻辑等更多特征信息
实现漏洞的智能化检索	漏洞知识理解与推理
现有工作所评估的漏洞字段有限	对更多字段信息的质量进行实验探究
对漏洞库开展准确的质量评估	提出通用质量衡量标准与评估方法
漏洞修复优先级的确定	考虑漏洞的利用性及非固有属性因素
对漏洞统计特征的探索性研究	结合更多数据来源以及漏洞报告特征进行统计分析

管理系统、论坛等不易被公众获取的来源中. 因此，如何收集这些未被披露的安全漏洞就成了学术界十分关注的问题[30-33]. 然而，目前的研究未能充分考虑模型在跨项目任务中的适用性与通用性，所选取的研究数据来源较为单一，并且数据集中的数据标注也受到提交报告用户专业知识和标注专家主观因素的影响，并非完全准确，这也影响了模型的训练效果. 除此以外，目前围绕漏洞研究十分依赖的 NVD、CVE 等漏洞库均为所收录的漏洞分配了唯一标识符 CVE-ID. 虽然 MITRE 已试图通过为每个漏洞分配唯一的 ID 来缓解漏洞标识的问题，但 Sun 等人[48]的研究仍然发现大多数漏洞库（例如 ExploitDB、Openwall、IBM X-Force）并不会在所有报告中引用 CVE-ID，比如对于 ExploitDB，有大约 52% 的条目信息都缺失 CVE-ID. 因此，异构漏洞数据的融合无法通过简单的编号索引匹配来实现. 对于没有 CVE-ID 的漏洞，仍然缺乏关联性分析，这也是在漏洞收集的相关工作中有待解决的问题.

（2）漏洞库的漏洞关系缺失和展示问题

在漏洞库构建过程中会单独存储漏洞数据，仅会存储少量有关漏洞关系的信息，这也导致漏洞库缺失了大量漏洞关系信息，使得漏洞库无法有效利用、展示漏洞间存在的关系. 为补充漏洞库缺失的漏洞关系，文献[107-108]给出了基于翻译模型的解决方案，文献[109-110]则通过使用图注意力网络模型解决了这一问题，然而他们的工作主要是针对 CVE、CWE 以及 CAPEC 三个数据库展开. 因此，未来工作可以考虑结合更多的漏洞库如 ExploitDB、OVAL 等. 在补充漏洞库缺失的漏洞关系后，为避免多义词和同义词对漏洞描述产生的歧义，文献[112-114]对研究词性标注和词嵌入技术展开了研究并解决了描述的歧义问题，然而他们没有处理漏洞描述中所存在的代码信息. 因此未来的工作也可以考虑加入代码逻辑消歧处理，当然这也存在很多挑战，比如是否可以处理不同的代码编写风格，是否能识别不同的代码语言等. 如何利用漏洞间的关系及漏洞描述更好地帮助漏洞库展示漏洞数据也是近期的研究热点. 文献[115-116]分别对智能问答系统和可视化工具进行了研究，他们的解决方案极大地便利了安全人员，然而对于不熟悉安全知识的人群来说，他们可能没办法明确表达自己的意思从而输入带有歧义的搜索条件. 因此未来的工作可以考虑当用户输入存在歧义时如何准确理解其含义，并将期待的答案返还给用户.

（3）如何开展准确的质量评估并指导漏洞库的完善

目前的工作分别围绕漏洞库整体评价以及漏洞数据的完整性、准确性和一致性进行探究；然而研究对象均更多地关注于经过标准化的非自由格式字段，如日期、CWE 类型、CVSS 评级以及受影响软件及版本等. 而对于漏洞描述、引用链接等自由格式字段，由于不具有标准化的结构，所以很难定义其质量问题并且也为多库之间的一致性评估带来了挑战. 除此以外，已有工作[119-120]也对如何建立漏洞库整体评价体系做出了探索，虽然目前各厂商及政府都会考虑建设自己的漏洞库系统，但领域内仍缺乏通用的质量衡量标准和通用的评估方法，并且该方向研究相对较少. 而对于漏洞数据质量的评估，已经逐渐成为近些年来该领域研究的热点问题，客观准确的评价结果将为漏洞库建设者如何改进工作提供重要参考. 另外，基于对漏洞数据准确性和一致性等的评估结果，如何考虑自动化的方案以更好地指导质量问题字段的补全和预测，也是一个有待解决的问题. 比如 Malone 等人[124]发现的 NVD 漏洞库中存在严重补丁缺失问题，只有大约四分之一的 CVE 条目标记出了补丁，那么如何做到及时和准确的补丁监测与收集，还需要一个通用成熟的解决方案.

（4）如何考虑优先级以支持更高效的漏洞修复

已有研究结合深度学习技术在自动化程序修复领域开展了一系列研究. 然而围绕漏洞修复优先级的研究则相对较少. 随着被发现漏洞的增加，安全团队经常被淹没在大量的漏洞告警中，但是安全人员不可能立即处置所有的问题，所以优先级的识别显得尤为重要. 尽管文献[2,155-157]做出了一些初步探索，但是可以看到，与缺陷优先级预测时考虑更多的因素比如缺陷的修复时间、开发人员特征等相比，

对于漏洞的优先级主要还是基于漏洞自身的固有属性进行研究，尤其是大多数研究还都基于 CVSS 进行修改或完善，而 CVSS 并未考虑漏洞的利用概率等，因此，未来对于优先级的预测研究应该考虑更多与利用性相关的因素以及非漏洞固有属性的因素，因为优先级与严重性相比更加面向解决漏洞的开发人员.

(5) 如何围绕漏洞报告所体现出的更多特征进行探索性分析

漏洞数据除了支持围绕漏洞预测、利用和修复等热点方向研究外，其还记录了漏洞在整个阶段的关键信息，如提交记录、发布日期、影响、修复信息等，因此，也支持通过探索性分析发现漏洞特征的统计规律，从而帮助安全分析人员更好地分析和修补漏洞，规避安全风险. 针对某个特定特征进行分析. 如 Alexopoulos 等人[212]关注于漏洞报告者的规律、Li 等人[215]关注漏洞的 CVSS 评级变化、Kudjo 等人[216]关注漏洞发现的规律等，以及针对多个漏洞特征间的关联性进行分析，如产品和漏洞的关系[69]、补丁的产生效率和软件及版本的关系[222]等等，都体现出漏洞特征的规律性、复杂性以及关联性的特点. 从对已有研究的调研来看，该方向研究起步较晚，且披露日期、严重性评级、漏洞报告者和受影响产品及版本等为当前研究重点关注的特征，而针对其他特征如报告修订记录、漏洞类型等还缺乏统计分析；另外，未来工作也可以考虑结合更多漏洞库如 ExploitDB、IBM X-Force Exchange 等展开.

除了以上不足与挑战，未来围绕漏洞库的研究还可以考虑在软件安全性评估、网络安全态势感知等方面的改进. 总之，对漏洞库中大量漏洞数据的挖掘与利用，还有较大的探索和提升空间，尤其随着知识图谱和大型语言模型技术的快速发展，相信更加完备、准确的漏洞库数据将在助力漏洞全生命周期的研究中大有可为.

8　总　结

随着人工智能技术的快速发展，越来越多的研究开始将机器学习、深度学习等技术应用于漏洞库研究中，不仅在漏洞数据处理和管理方面，在漏洞检测、修复等围绕漏洞生命周期的研究中也取得了较好的应用效果，极大推进了安全漏洞领域的研究进展.

本文从基础知识、背景、理论方法和创新等方面对近些年来的研究进行了调查和分析，首次全面、系统地介绍了近些年来围绕漏洞库构建与应用的代表性成果，发现该领域已有工作中存在的不足并总结了我们认为未来可能的研究趋势. 漏洞库研究作为计算机领域的一个研究热点，目前仍有许多问题需要做深入的研究.

参　考　文　献

[1] Feng S, Lubis M. Defense-in-depth security strategy in Log4j vulnerability analysis//Proceedings of the International Conference Advancement Data Science, E-learning and Information Systems. Bandung, Indonesia, 2022: 01-04

[2] Le T H, Chen H, Babar M A. A survey on data-driven software vulnerability assessment and prioritization. ACM Computing Surveys, 2022, 55(5): 1-39

[3] Nguyen V H, Massacci F. The (un) reliability of NVD vulnerable versions data: An empirical experiment on Google chrome vulnerabilities//Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. New York, USA: Association for Computing Machinery, 2013: 493-498

[4] Hao Yong-Le, Zheng Liang, Jia Yi-Zhen, et al. GB/T 30279-2020. Information security technology—Guidelines for categorization and classification of cybersecurity vulnerability. State Administration for Market Regulation; Standardization Administration of the People's Republic, 2020(in Chinese)
(郝永乐，郑亮，贾依真等. GB/T 30279-2020. 信息安全技术 网络安全漏洞分类分级指南. 国家市场监督管理总局；国家标准化管理委员会，2020)

[5] Zhang Yu-Qing, Wu Shu-Ping, Liu Qi-Xu, et al. Design and implementation of national security vulnerability database. Journal on Communications, 2011, 32(6): 93-100(in Chinese)
(张玉清，吴舒平，刘奇旭等. 国家安全漏洞库的设计与实现. 通信学报，2011，32(6)：93-100)

[6] CNNVD. China National Vulnerability Database of Information Security[EB/OL]. https://www.cnnvd.org.cn/

[7] CNVD. China National Vulnerability Database. https://www.cnvd.org.cn/

[8] Menefee T, Gaitan I, Abadura A. National Vulnerability Database[EB/OL]. https://nvd.nist.gov/vuln

[9] Rasch M, Poulsen K. SecurityFocus [EB/OL]. http://www.securityfocus.com/

[10] Flexera. Secunia[EB/OL]. http://secunia.com/

[11] IBM. IBM X-Force Exchange[EB/OL]. http://xforce.iss.net

[12] Amxku. Seebug[EB/OL]. https://www.seebug.org/

[13] OffSec. Exploit Database[EB/OL]. https://www.exploit-db.com/

[14] Armstrong K, Beardsley T, Coffin C. CVE. https://cve.mitre.org/

[15] Yun Xiao-Chun, Shu Min, Cui Mu-Fan, et al. GB/T 30276-2020. Information security technology—Specification for cybersecurity vulnerability management. State Administration for Market Regulation; Standardization Administration of the People's Republic, 2020(in Chinese)
(云晓春，舒敏，崔牧凡等. 信息安全技术网络安全漏洞管理规范. GB/T 30276-2020. 国家市场监督管理总局；国家标准化管理委员会，2020)

[16] Mell P, Scarfone K. Common Vulnerability Scoring System v3.0: User Guide[EB/OL]. https://www.first.org/cvss/v3.0/user-guide

[17] Liu Qi-Xu, Zhang Yu-Qing, Gong Ya-Feng, et al. The development of the vulnerability identification and description specification. Netinfo Security, 2011, (7): 4-6(in Chinese)
(刘奇旭，张玉清，宫亚峰等. 安全漏洞标识与描述规范的研究. 信息网络安全，2011，(7)：4-6)

[18] Anderson P, Curtis B. Common Weakness Enumeration. https://cwe.mitre.org/

[19] Menefee T, Gaitan I, Abadura A. Official Common Platform Enumeration(CPE) Dictionary. https://nvd.nist.gov/products/cpe

[20] CXSEC. CXSecurity. https://cxsecurity.com/

[21] PacketStormSecurity. Exploit the possibilities. https://packetstormsecurity.com/

[22] Jia Pei-Yang, Sun Hong-Yu, Cao Wan-Ying, et al. Open source software vulnerability data base overview. Journal of Information Security Research, 2021, 7(6): 566-574(in Chinese)
(贾培养，孙鸿宇，曹婉莹等. 开源软件漏洞库综述. 信息安全研究，2021，7(6)：566-574)

[23] NIPC. National Computer Network Instrusion Protection Center. http://nipc.org.cn/

[24] CICSVD. China National Industrial Cyber Security Vulnerability Database. https://www.cics-vd.org.cn/

[25] Zhuge Jianwei. Information Security Vulnerability Portal. https://www.scap.org.cn/

[26] NSFocus. Vulnerability. http://www.nsfocus.net/

[27] Wu Shu-Ping, Zhang Yu-Qing. Research and Enlightenment of the Development Status of Vulnerability Database. Computer Security, 2010, (11): 82-84(in Chinese)
(吴舒平，张玉清. 漏洞库发展现状的研究及启示. 计算机安全，2010，(11)：82-84)

[28] Yang Gang. Current situation analysis and quality evaluation of vulnerability database. Telecommunications Network Technology, 2018, (2): 6-15(in Chinese)
(杨刚. 漏洞库现状分析及质量评价. 电信网技术，2018，(2)：6-15)

[29] Han Z, Li X, Xing Z, et al. Learning to predict severity of software vulnerability using only vulnerability description//Proceedings of the 2017 IEEE International Conference on Software Maintenance and Evolution (ICSME). Shanghai, China, 2017: 125-136

[30] Wu X, Zheng W, Chen X, et al. CVE-assisted large-scale security bug report dataset construction method. The Journal of Systems and Software, 2020, 160: 110456

[31] Wu X, Zheng W, Xia X, et al. Data quality matters: A case study on data label correctness for security bug report prediction. IEEE Transactions on Software Engineering, 2022, 48: 2541-2556

[32] Zheng W, Xun Y, Wu X, et al. A comparative study of class rebalancing methods for security bug report classification. IEEE Transactions on Reliability, 2021, 70: 1658-1670

[33] Zhou Y, Sharma A. Automated identification of security issues from commit messages and bug reports//Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE). New York, USA, 2017: 914-919

[34] Pereira J D, Antunes J H, Vieira M. A software vulnerability dataset of large open source C/C++ projects//Proceedings of the 2022 IEEE 27th Pacific Rim International Symposium on Dependable Computing (PRDC). Beijing, China, 2022: 152-163

[35] Chaleshtori F H, Ray I. Automation of vulnerability information extraction using transformer-based language models//Proceedings of the European Symposium on Research in Computer Security. Cham: Springer International Publishing. Copenhagen, Denmark, 2022: 645-665

[36] Armold A D, Hyla B M, Rowe N. Automatically building an information-security vulnerability database//Proceedings of the 2006 IEEE Information Assurance Workshop. West Point, USA, 2006: 376-377

[37] Nguyen-Truong G, Kang H J, Lo D, et al. Hermes: Using commit-issue linking to detect vulnerability-fixing commits//Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). Honolulu, USA, 2022: 51-62

[38] Hong H, Woo S, Choi E, et al. xVDB: A high-coverage approach for constructing a vulnerability database. IEEE Access, 2022, 10: 85050-85063

[39] Sun M, Wang W, Feng H, et al. Identify vulnerability fix commits automatically using hierarchical attention network. EAI Endorsed Transactions on Security and Safety, 2020, 7(23): e2

[40] Wang X, Wang S, Feng P, et al. PatchDB: A large-scale security patch dataset//Proceedings of the 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks(DSN). Taipei, China, 2021: 149-160

[41] Zhou Y, Siow J, Wang C, et al. SPI: Automated identification of security patches via commits. ACM Transactions on Software Engineering and Methodology (TOSEM), 2021, 31(1): 1-27

[42] Bhandari G, Naseer A, Moonen L. CVEfixes: Automated collection of vulnerabilities and their fixes from open-source software//Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering. New York, USA, 2021: 30-39

[43] Tan X, Zhang Y, Mi C, et al. Locating the security patches for disclosed OSS vulnerabilities with vulnerability-commit correlation ranking//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York, USA, 2021: 3282-3299

[44] Wang Z, Wang Z, Wang Z, et al. Design and implementation of security vulnerability sharing platform based on web crawler//Proceedings of the 11th International Conference on Computer Engineering and Networks. Hechi, China, 2022: 678-687

[45] Ponta S E, Plate H, Sabetta A, et al. A manually-curated dataset of fixes to vulnerabilities of open-source software//Proceedings of the 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). Montreal, Canada, 2019: 383-387

[46] Jimenez M, Traon Y L, Papadakis M. Enabling the continuous analysis of security vulnerabilities with VulData7//Proceedings of the 2018 IEEE 18th International Working Conference on Source Code Analysis and Manipulation (SCAM). Madrid, Spain, 2018: 56-61

[47] Li X, Hu C, Feng Z Y, et al. An approach to obtain software security vulnerabilities based on vertical search. Advanced Materials Research, 2012, 403: 3203-3206

[48] Sun J, Xing Z, Xu X, et al. Heterogeneous vulnerability report traceability recovery by vulnerability aspect matching//Proceedings of the 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME). Limassol, Cyprus, 2022: 175-186

[49] Guo H, Xing Z, Chen S, et al. Key aspects augmentation of vulnerability description based on multiple security databases//Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). Madrid, Spain, 2021: 1020-1025

[50] Kang W, Son B, Heo K. Tracer: Signature-based static analysis for detecting recurring vulnerabilities//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS). New York, USA, 2022: 1695-1708

[51] Jiang J. Information extraction from text. Mining Text Data, 2012: 11-41

[52] Bridges R, Jones C L, Iannacone M D, et al. Automatic labeling for entity extraction in cyber security. arXiv: abs/1308.4941, 2013

[53] Russo E, Sorbo A D, Visaggio C A, et al. Summarizing vulnerabilities' descriptions to support experts during vulnerability assessment activities. The Journal of Systems and Software, 2019, 156: 84-99

[54] Marconato G, Nicomette V, Kaâniche M. Security-related vulnerability life cycle analysis//Proceedings of the 2012 7th International Conference on Risks and Security of Internet and Systems (CRiSIS). Cork, Ireland, 2012: 1-8

[55] Zheng W, Zhang M, Tang H, et al. Automatically identifying bug reports with tactical vulnerabilities by deep feature learning//Proceedings of the 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE). Wuhan, China, 2021: 333-344

[56] Zou D, Deng Z, Li Z, et al. Automatically identifying security bug reports via multitype features analysis//Proceedings of the Australasian Conference on Information Security and Privacy. Wollongong, Australia, 2018: 619-633

[57] Behl D, Handa S, Arora A. A bug mining tool to identify and analyze security bugs using naive bayes and TF-IDF//Proceedings of the International Conference on Reliability Optimization and Information Technology. Faridabad, India, 2014: 294-299

[58] Yang X, Lo D, Xia X, et al. High-impact bug report identification with imbalanced learning strategies. Journal of Computer Science and Technology, 2017, 32: 181-198

[59] Cao X, Liu T, Zhang J, et al. SbrPBert: A BERT-based model for accurate security bug report prediction//Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). Baltimore, USA, 2022: 129-134

[60] Zhou Peng, Wu Yan-Jun, Zhao Chen. Identify Linux security vulnerability fix patches automatically. Journal of Computer Research and Development, 2022, 59(1): 197-208(in Chinese)

(周鹏，武延军，赵琛. 一种 Linux 安全漏洞修复补丁自动识别方法. 计算机研究与发展，2022，59(1)：197-208)

[61] Zhou J, Pacheco M, Wan Z, et al. Finding a needle in a haystack: Automated mining of silent vulnerability fixes//Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). Melbourne, Australia, 2021: 705-716

[62] Alon U, Zilberstein M, Levy O, et al. code2vec: Learning distributed representations of code. Proceedings of the ACM on Programming Languages, 2019, 3: 1-29

[63] Wang X, Sun K, Batcheller A, et al. Detecting "0-Day" vulnerability: An empirical study of secret security patch in OSS//Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Portland, USA, 2019: 485-492

[64] Wang S, Zhang Y, Bao L, et al. VCMatch: A ranking-based approach for automatic security patches localization for oss vulnerabilities//Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). Honolulu, USA, 2022: 589-600

[65] Li R, Tan S, Wu C, et al. IFVD: Design of intelligent fusion framework for vulnerability data based on text measures//Proceedings of the 2020 29th International Conference on Computer Communications and Networks (ICCCN). Honolulu, USA, 2020: 1-6

[66] Rodriguez L G A, Trazzi J S, Fossaluza V, et al. Analysis of vulnerability disclosure delays from the national vulnerability database//Proceedings of the Anais do I Workshop de Segurança Cibernética em Dispositivos Conectados. Porto Alegre, Brasil, 2018

[67] Chan N, Chandy J. Extracting vulnerabilities from github commits//Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). Honolulu, USA, 2022: 235-239

[68] Gong X, Xing Z, Li X, et al. Joint prediction of multiple vulnerability characteristics through multi-task learning//Proceedings of the 2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS). Guangzhou, China, 2019: 31-40

[69] Tsutsui T, Shiraishi Y, Morii M. Systemization of vulnerability information by ontology for impact analysis//Proceedings of the 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C). Sanya, China, 2021: 1126-1134

[70] Pham V V, Dang T. CVExplorer: Multidimensional visualization for common vulnerabilities and exposures//Proceedings of the 2018 IEEE International Conference on Big Data (Big Data). Seattle, USA, 2018: 1296-1301

[71] Reynolds S L, Mertz T, Arzt S, et al. User-centered design of visualizations for software vulnerability reports//Proceedings of the 2021 IEEE Symposium on Visualization for Cyber Security (VizSec). New Orleans, USA, 2021: 68-78

[72] Tian H, Huang L, Zhou Z, et al. Common vulnerability markup language//Proceedings of the International Conference on Applied Cryptography and Network Security. Kunming, China, 2003: 228-240

[73] Zheng C, Zhang Y, Sun Y, et al. IVDA: International vulnerability database alliance//Proceedings of the 2011 Second Worldwide Cybersecurity Summit (WCS). London, UK, 2011: 1-6

[74] Le T H, Sabir B, Babar M. Automated software vulnerability assessment with concept drift//Proceedings of the 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR). Montreal, Canada, 2019: 371-382

[75] Sun J, Xing Z, Guo H, et al. Generating informative CVE description from ExploitDB posts by extractive summarization. arXiv: abs/2101.01431, 2021

[76] Kuehn P D, Bayer M, Wendelborn M, et al. OVANA: An approach to analyze and improve the information quality of vulnerability databases//Proceedings of the 16th International Conference on Availability, Reliability and Security. New York, USA, 2021: 1-11

[77] Yitagesu S, Xing Z, Zhang X, et al. Unsupervised labeling and extraction of phrase-based concepts in vulnerability descriptions//Proceedings of the 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). Melbourne, Australia, 2021: 943-954

[78] Aghaei E, Shadid W, Al-Shaer E. ThreatZoom: CVE2CWE using hierarchical neural network. arXiv: abs/2009.11501, 2020

[79] Wang T, Qin S, pui Chow K. Towards vulnerability types classification using pure self-attention: A common weakness enumeration based approach//Proceedings of the 2021 IEEE 24th International Conference on Computational Science and Engineering (CSE). Shenyang, China, 2021: 146-153

[80] Das S S, Serra E, Halappanavar M, et al. V2W-BERT: A framework for effective hierarchical multiclass classification of software vulnerabilities//Proceedings of the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA). Porto, Portugal, 2021: 1-12

[81] Panchal K, Das S S, Torre L D L, et al. Efficient clustering of software vulnerabilities using self organizing map (SOM)//Proceedings of the 2022 IEEE International Symposium on Technologies for Homeland Security (HST). Boston, USA, 2022: 1-7

[82] Zhu C, Du G, Wu T, et al. BERT-based vulnerability type identification with effective program representation//Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications. Dalian, China, 2022: 271-282

[83] He Y, Wang Y, Zhu S, et al. Automatically identifying CVE affected versions with patches and developer logs. IEEE Transactions on Dependable and Secure Computing, 2023, 1-15

[84] Wåreus E, Hell M. Automated CPE labeling of CVE summaries with machine learning. Lecture Notes in Computer Science, 2020, 12223: 3-22

[85] Glanz L, Schmidt S, Wollny S, et al. A vulnerability's lifetime: Enhancing version information in CVE databases//Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business. New York, USA, 2015: 1-4

[86] Bao L, Xia X, Hassan A, et al. V-SZZ: Automatic identification of version ranges affected by CVE vulnerabilities//Proceedings of the 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE). New York, USA, 2022: 2352-2364

[87] Spanos G, Angelis L, Toloudis D. Assessment of vulnerability severity using text mining//Proceedings of the 21st Pan-Hellenic Conference on Informatics. New York, USA, 2017: 1-6

[88] Wang P, Zhou Y, Sun B, et al. Intelligent prediction of vulnerability severity level based on text mining and XGBboost//Proceedings of the 2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI). Guilin, China, 2019: 72-77

[89] Lee Y, Shin S. Toward semantic assessment of vulnerability severity: A text mining approach//Proceedings of the Conference on Information and Knowledge Management (CIKM Workshops). Torino, Italy, 2018

[90] Kekül H, Ergen B, Arslan H. A multiclass hybrid approach to estimating software vulnerability vectors and severity score. Journal of Information Security and Applications, 2021, 63: 103028

[91] Ohuabunwa B C. Predicting Cybersecurity Vulnerability Severity Via Boosted Machine Learning Ensembles and Feature Ranking [Ph. D. dissertation]. The George Washington University, Washington, USA, 2022

[92] Liu Q, Zhang Y. VRSS: A new system for rating and scoring vulnerabilities. Computer Communications, 2011, 34(3): 264-273

[93] Zou D, Yang J, Li Z, et al. AutoCVSS: An approach for automatic assessment of vulnerability severity based on attack process//Proceedings of the Green, Pervasive, and Cloud Computing. Uberlândia, Brazil, 2019: 238-253

[94] Holm H, Afridi K K. An expert-based investigation of the common vulnerability scoring system. Computers & Security, 2015, 53: 18-30

[95] Dass S, Namin A. Evolutionary algorithms for vulnerability coverage//Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC). Madrid, Spain, 2020: 1795-1801

[96] Nikonov A, Vulfin A, Vasilyev V, et al. System for estimation cvss severity metrics of vulnerability based on text mining technology//Proceedings of the 2021 International Conference on Information Technology and Nanotechnology (ITNT). Samara, Russian Federation, 2021: 1-5

[97] Gencer K, Başçiftçi F. The fuzzy common vulnerability scoring system (F-CVSS) based on a least squares approach with fuzzy logistic regression. Egyptian Informatics Journal, 2021, 22(2): 145-153

[98] Spanos G, Angelis L. A multi-target approach to estimate software vulnerability characteristics and severity scores. The Journal of Systems and Software, 2018, 146: 152-166

[99] Malhotra R, Vidushi V. Impact of word embedding methods on software vulnerability severity prediction models//Proceedings of the 2023 13th International Conference on Cloud Computing, Data Science & Engineering. Noida, India, 2023: 293-297

[100] Li Z, Tang C, bin Hu J, et al. Vulnerabilities scoring approach for cloud saas//Proceedings of the 2015 IEEE 12th International Conference on Ubiquitous Intelligence and Computing and 2015 IEEE 12th International Conference on Autonomic and Trusted Computing and 2015 IEEE 15th International Conference on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom). Beijing, China, 2015: 1339-1347

[101] Rivera A C A, Shaghaghi A, Nguyen D D, et al. Is this IoT device likely to be secure? Risk score prediction for IoT devices using gradient boosting machines//Proceedings of the International Conference on Mobile and Ubiquitous Systems: Networking and Services. Virtual Event, 2021: 115-127

[102] Li X, Ren X, Xue Y, et al. Prediction of vulnerability characteristics based on vulnerability description and prompt learning//Proceedings of the 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). Taipa, Macao, 2023: 604-615

[103] Lim S, Muis A O, Lu W, et al. Malwaretextdb: A database for annotated malware articles//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL). Vancouver, Canada, 2017: 1557-1567

[104] Elbaz C, Rilling L, Morin C. Automated keyword extraction from "One-day" Vulnerabilities at disclosure//Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium. Budapest, Hungary, 2020: 1-9

[105] Allodi L, Banescu S, Femmer H, et al. Identifying relevant information cues for vulnerability assessment using CVSS//Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy (CODASPY). New York, USA, 2018: 119-126

[106] Jia Yan, Qi Yu-Lu, Shang Huai-Jun, et al. A practical approach to constructing a knowledge graph for cybersecurity. Engineering, 2018, 4(1): 53-60

[107] Xiao H, Xing Z, Li X, et al. Embedding and predicting software security entity relationships: A knowledge graph based approach//Proceedings of the International Conference on Neural Information Processing. Sydney, Australia, 2019: 50-63

[108] Han Z, Li X, Liu H, et al. DeepWeak: Reasoning common software weaknesses via knowledge graph embedding//Proceedings of the 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER). Campobasso, Italy, 2018: 456-466

[109] Yuan L, Bai Y, Xing Z, et al. Predicting entity relations across different security databases by using graph attention network//Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC). Madrid, Spain, 2021: 834-843

[110] Wang Y, Hou X, Ma X, et al. A software security entity relationships prediction framework based on knowledge graph embedding using sentence-BERT//Proceedings of the Wireless Algorithms, Systems, and Applications. Dalian, China, 2022: 501-513

[111] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Hong Kong, China, 2019: 3980-3990

[112] Ye D, Xing Z, Li J, et al. Software-specific part-of-speech tagging: An experimental study on stack overflow//Proceedings of the 31st Annual ACM Symposium on Applied Computing. New York, USA, 2016: 1378-1385

[113] Yitagesu S, Zhang X, Feng Z, et al. Automatic part-of-speech tagging for security vulnerability descriptions//Proceedings of the 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR). Madrid, Spain, 2021: 29-40

[114] Mumtaz S, Rodríguez C, Benatallah B, et al. Learning word representation for the cyber security vulnerability domain//Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN). Glasgow, UK, 2020: 1-8

[115] Cheng X, Sun X, Bo L, et al. KVS: A tool for knowledge-driven vulnerability searching//Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). New York, USA, 2022: 1731-1735

[116] Li Y, Guo Y, Hao Y, et al. Intelligent answer system based on vulnerability knowledge graph//Proceedings of the 2021 7th International Conference on Computer and Communications. Chengdu, China, 2021: 1646-1651

[117] Forain I, Albuquerque R O, Júnior R T S. Towards system security: What a comparison of national vulnerability databases reveals//Proceedings of the 2022 17th Iberian Conference on Information Systems and Technologies (CISTI). Madrid, Spain, 2022: 1-6

[118] Jiang Y, Jeusfeld M, Ding J. Evaluating the data inconsistency of open-source vulnerability repositories//Proceedings of the 16th International Conference on Availability, Reliability and Security. New York, USA, 2021: 1-10

[119] Tan Tiantian, Wang Baosheng, Tang Yong, et al. A method for vulnerability database quantitative evaluation. Computers, Materials & Continua, 2019, 61(3): 1129-1144

[120] Li X, Moreschini S, Zhang Z, et al. The anatomy of a vulnerability database: A systematic mapping study. Journal of Systems and Software, 2023, 201: 111679

[121] Croft R, Babar M A, Kholoosi M. Data quality for software vulnerability datasets//Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE). Melbourne, Australia, 2023: 121-133

[122] Gallon L. On the impact of environmental metrics on CVSS scores//Proceedings of the 2010 IEEE Second International Conference on Social Computing. Minneapolis, USA, 2010: 987-992

[123] Lyu J, Bai Y, Xing Z, et al. A character-level convolutional neural network for predicting exploitability of vulnerability//Proceedings of the 2021 International Symposium on Theoretical Aspects of Software Engineering (TASE). Shanghai, China, 2021: 119-126

[124] Malone M, Wang Y, Snow K, et al. Applicable micropatches and where to find them: Finding and applying new security hot fixes to old software//Proceedings of the 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST). Porto de Galinhas, Brazil, 2021: 394-405

[125] Johnson P, Lagerström R, Ekstedt M, et al. Can the common vulnerability scoring system be trusted? A Bayesian analysis. IEEE Transactions on Dependable and Secure Computing, 2016, 15(6): 1002-1015

[126] Christey S, Martin B. Buying into the bias: Why vulnerability statistics suck. Las Vegas, USA: MITRE, BlackHat: Technical Report 1, 2013

[127] Dong Y, Guo W, Chen Y, et al. Towards the detection of inconsistencies in public security vulnerability reports//Proceedings of the 28th USENIX Security Symposium (USENIX Security 19). Santa Clara, USA, 2019: 869-885

[128] Massacci F, Nguyen V H. Which is the right source for vulnerability studies? An empirical analysis on Mozilla Firefox//Proceedings of the 6th International Workshop on Security Measurements and Metrics (MetriSec'10). New York, USA, 2010: 1-8

[129] Anwar A, Abusnaina A, Chen S, et al. Cleaning the NVD: Comprehensive quality assessment, improvements, and analyses. IEEE Transactions on Dependable and Secure Computing, 2021, 19(6): 4255-4269

[130] Croft R, Babar M, Li L. An investigation into inconsistency of software vulnerability severity across data sources//Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). Honolulu, USA, 2022: 338-348

[131] Ruohonen J, Rauti S, Hyrynsalmi S, et al. A case study on software vulnerability coordination. Information and Software Technology, 2018, 103: 239-257

[132] Zheng W, Gao J, Wu X, et al. The impact factors on the performance of machine learning-based vulnerability detection: A comparative study. The Journal of Systems and Software, 2020, 168: 110659

[133] Bullough B L, Yanchenko A K, Smith C L, et al. Predicting exploitation of disclosed software vulnerabilities using open-source data//Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics. New York, USA, 2017: 45-53

[134] Black P. A software assurance reference dataset: Thousands of programs with known bugs. Journal of Research of the National Institute of Standards and Technology, 2018, 123: 1-3

[135] Croft R, Xie Y, Babar M A. Data preparation for software vulnerability prediction: A systematic literature review. IEEE Transactions on Software Engineering, 2021, 49: 1044-1063

[136] Lin Y, Li Y, Gu M, et al. Vulnerability dataset construction methods applied to vulnerability detection: A survey//Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). Baltimore, USA, 2022: 141-146

[137] Liu L, Li Z, Wen Y, et al. Investigating the impact of vulnerability datasets on deep learning-based vulnerability detectors. Peer J Computer Science, 2022, 8: e975

[138] Garg A, Degiovanni R, Jimenez M, et al. Learning from what we know: How to perform vulnerability prediction using noisy historical data. Empirical Software Engineering, 2022, 27(7): 169

[139] Lomio F, Iannone E, De Lucia A, et al. Just-in-time software vulnerability detection: Are we there yet? Journal of Systems and Software, 2022, 188: 111283

[140] Li Z, Zou D, Xu S, et al. VulDeePecker: A deep learning-based system for vulnerability detection//Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS 2018). San Diego, USA, 2018: 1-15

[141] Wu Y, Zou D, Dou S, et al. VulCNN: An image-inspired scalable vulnerability detection system//Proceedings of the 44th International Conference on Software Engineering. New York, USA, 2022: 2365-2376

[142] Kim S, Choi J, Ahmed M E, et al. VulDeBERT: A vulnerability detection system using BERT//Proceedings of the 2022 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). Charlotte, USA, 2022: 69-74

[143] Cao S, Sun X, Bo L, et al. BGNN4VD: Constructing bidirectional graph neural-network for vulnerability detection. Information and Software Technology, 2021, 136: 106576

[144] Nguyen V A, Nguyen D Q, Nguyen V, et al. ReGVD: Revisiting graph neural networks for vulnerability detection//Proceedings of the 44th International Conference on Software Engineering Companion (ICSE-Companion). New York, USA, 2022: 178-182

[145] Steenhoek B, Rahman M M, Jiles R, et al. An empirical study of deep learning models for vulnerability detection//Proceedings of the 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE). Melbourne, Australia, 2023: 2237-2248

[146] Zhang S, Caragea D, Ou X. An empirical study on using the national vulnerability database to predict software vulnerabilities//Proceedings of the International Conference on Database and Expert Systems Applications. Toulouse, France, 2011: 217-231

[147] Williams M A, Barranco R C, Naim S M, et al. A vulnerability analysis and prediction framework. Computers & Security, 2020, 92: 101751

[148] Last D C. Using historical software vulnerability data to forecast future vulnerabilities//Proceedings of the 2015 Resilience Week (RWS). Philadelphia, USA, 2015: 1-7

[149] Johnson P, Gorton D, Lagerström R, et al. Time between vulnerability disclosures: A measure of software product vulnerability. Computers & Security, 2016, 62: 278-295

[150] Houmz A, Mezzour G, Zkik K, et al. Detecting the impact of software vulnerability on attacks: A case study of network telescope scans. Journal of Network and Computer Applications, 2021, 195: 103230

[151] Satheesh N, Mydukuri R V, Rajeshkumar G, et al. Flow-based anomaly intrusion detection using machine learning model with software defined networking for openflow network. Microprocessors & Microsystems, 2020, 79: 103285

[152] Sultan S, Salman A. Calcium vulnerability scanner (CVS): A deeper look. arXiv: abs/1911.00950. 2019

[153] O'Hare J. Scout: A Contactless 'Active' Reconnaissance Known Vulnerability Assessment Tool [Undergraduate degree]. Edinburgh Napier University, Edinburgh, UK, 2018

[154] Jeon S, Kim H. Autovas: An automated vulnerability analysis system with a deep learning approach. Computers & Security, 2021, 106: 102308

[155] Sharma A, Sabharwal S, Nagpal S. A hybrid scoring system for prioritization of software vulnerabilities. Computers & Security, 2023, 129: 103256

[156] Costa T F, Tymburibá M. Challenges on prioritizing software patching//Proceedings of the 2022 15th International Conference on Security of Information and Networks (SIN). Sousse, Tunisia, 2022: 1-8

[157] Sato R, Kawaguchi H, Nakatani Y. A stochastic model for calculating well-founded probabilities of vulnerability exploitation//Proceedings of the 2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS-C). Guangzhou, China, 2022: 34-43

[158] da Ponte F R P, Rodrigues E, Mattos C. A vulnerability risk assessment methodology using active learning//Proceedings of the International Conference on Advanced Information Networking and Applications. Juiz de Fora, Brazil, 2023: 171-182

[159] Fang Y, Liu Y, Huang C, et al. FastEmbed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm. PLoS ONE, 2020, 15(2): e0228439

[160] Jacobs J, Romanosky S, Suciuo O, et al. Enhancing vulnerability prioritization: Data-driven exploit predictions with community-driven insights//Proceedings of the 2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). Delft, Netherlands, 2023: 194-206

[161] Feutrill A, Ranathunga D, Yarom Y, et al. The effect of common vulnerability scoring system metrics on vulnerability exploit delay//Proceedings of the 2018 Sixth International Symposium on Computing and Networking (CANDAR). Takayama, Japan, 2018: 1-10

[162] Sharma R, Sibal R, Sabharwal S. Software vulnerability prioritization using vulnerability description. International Journal of System Assurance Engineering and Management, 2020, 12: 58-64

[163] Wu Y, Jiang N, Pham H, et al. How effective are neural networks for fixing security vulnerabilities//Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA). Seattle, USA, 2023: 1282-1294

[164] Fan J, Li Y, Wang S, et al. A C/C++ code vulnerability dataset with code changes and CVE summaries//Proceedings of the 17th International Conference on Mining Software Repositories (MSR). New York, USA, 2020: 508-512

[165] Chi J, Qu Y, Liu T, et al. Seqtrans: Automatic vulnerability fix via sequence to sequence learning. IEEE Transactions on Software Engineering, 2020, 49: 564-585

[166] Anjum M, Singhal S, Kapur P K, et al. Analysis of vulnerability fixing process in the presence of incorrect patches. The Journal of Systems and Software, 2022, 195: 111525

[167] Li F, Paxson V. A large-scale empirical study of security patches//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS). New York, USA, 2017: 2201-2215

[168] Wang X, Wang S, Sun K, et al. A machine learning approach to classify security patches into vulnerability types //Proceedings of the 2020 IEEE Conference on Communications and Network Security(CNS). Avignon, France, 2020: 1-9

[169] Nappa A, Johnson R B, Bilge L, et al. The attack of the clones: A study of the impact of shared code on vulnerability patching//Proceedings of the 2015 IEEE Symposium on Security and Privacy(SP). San Jose, USA, 2015: 692-708

[170] Zhou A, Sultana K Z, Samanthula B. Investigating the changes in software metrics after vulnerability is fixed// Proceedings of the 2021 IEEE International Conference on Big Data (Big Data). Orlando, USA, 2021: 5658-5663

[171] Forootani S, Di Sorbo A, Visaggio C A. An exploratory study on self-fixed software vulnerabilities in oss projects// Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering(SANER). Honolulu, USA, 2022: 90-100

[172] Zhang Q, Fang C, Yu B, et al. Pre-trained model-based automated software vulnerability repair: How far are we? IEEE Transactions on Dependable and Secure Computing, 2023: 1-18

[173] Fu M, Tantithamthavorn C, Le T, et al. VulRepair: A T5-based automated software vulnerability repair//Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). New York, USA, 2022: 935-947

[174] Anwar A, Khormali A, Mohaisen A. Poster: Understanding the hidden cost of software vulnerabilities: Measurements and predictions//Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS). New York, USA, 2018: 793-795

[175] Chen Y, Santosa A, Sharma A, et al. Automated identification of libraries from vulnerability data//Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice. New York, USA, 2020: 90-99

[176] Haryono S A, Kang H J, Sharma A, et al. Automated identification of libraries from vulnerability data: Can we do better?//Proceedings of the 2022 IEEE/ACM 30th International Conference on Program Comprehension (ICPC). New York, USA, 2022: 178-189

[177] Lyu Yun-Bo, Thanh Le-Cong, Kang Hong-Jin, et al. Chronos: Time-aware zero-shot identification of libraries from vulnerability reports//Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE). Melbourne, Australia, 2023: 1033-1045

[178] Zhao L, Chen S, Xu Z, et al. Software composition analysis for vulnerability detection: An empirical study on java projects//Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York, USA, 2023: 960-972

[179] Zhang Yi-Fan, Tang En-Yi, Su Yan-Zi, et al. Natural language data driven approach for software intelligent safety evaluation. Journal of Software, 2018, 29(8): 2336-2349 (in Chinese)
(张一帆, 汤恩义, 苏琰梓等. 自然语言数据驱动的智能化软件安全评估方法. 软件学报, 2018, 29(8): 2336-2349)

[180] Rasheed H. Vulnerability distribution scoring for software product security assessment. International Journal of Information and Computer Security, 2014, 6: 270-285

[181] Gao J B, Zhang B W, Chen X H, et al. Ontology-based model of network and computer attacks for security assessment. Journal of Shanghai Jiaotong University (Science), 2013, 18: 554-562

[182] Wang Y, Sun Z, Han Y. Network attack path prediction based on vulnerability data and knowledge graph. International Journal of Innovative Computing, Information and Control, 2021, 17(5): 1717

[183] Sun P, Zhang H, Li C. Attack path prediction based on Bayesian game model. Journal of Physics: Conference Series. IOP Publishing, 2021, 1955, (1): 012098

[184] Liu X. A network attack path prediction method using attack graph. Journal of Ambient Intelligence and Humanized Computing, 2020: 1-8

[185] Datta P, Namin A, Jones K S. Can we predict consequences of cyber attacks?//Proceedings of the 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). Nassau, Bahamas, 2022: 1047-1054

[186] Liu C, Singhal A, Wijesekera D. A layered graphical model for mission attack impact analysis//Proceedings of the 2017 IEEE Conference on Communications and Network Security (CNS). Las Vegas, USA, 2017: 602-609

[187] Dang Q V, François J. Utilizing attack enumerations to study SDN/NFV vulnerabilities//Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). Montreal, Canada, 2018: 356-361

[188] Kanakogi K, Washizaki H, Fukazawa Y, et al. Tracing CVE vulnerability information to CAPEC attack patterns using natural language processing techniques. Information, 2021, 12(8): 298

[189] Kanakogi K, Washizaki H, Fukazawa Y, et al. Tracing CAPEC attack patterns from CVE vulnerability information using natural language processing technique//Proceedings of the Hawaii International Conference on System Sciences. Hawaii, USA, 2021: 1-9

[190] Grigorescu O, Nica A, Dascalu M, et al. CVE2ATT&CK: BERT-based mapping of CVEs to mitre ATT&CK techniques. Algorithms, 2022, 15(9): 314

[191] Aksu M U, Bicakci K, Dilek M, et al. Automated generation of attack graphs using NVD//Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. New York, USA, 2018: 135-142

[192] Cheng F, Roschke S, Schuppenies R, et al. Remodeling vulnerability information//Proceedings of the Information Security and Cryptology. Beijing, China, 2010: 324-336

[193] Bezawada B, Ray I, Tiwary K. AGBuilder: An ai tool for automated attack graph building, analysis, and refinement //Proceedings of the Data and Applications Security and Privacy XXXIII. Charleston, USA, 2019: 23-42

[194] Yu X, Jiang J, Shuai C. Approach to attack path generation based on vulnerability correlation//Proceedings of the IEEE Conference Anthology. China, 2013: 1-6

[195] Keramati M. An attack graph based procedure for risk estimation of zero-day attacks//Proceedings of the 2016 8th International Symposium on Telecommunications (IST). Tehran, Iran, 2016: 723-728

[196] Liu H, Li B. Automated classification of attacker privileges based on deep neural network//Proceedings of the International Conference on Smart Computing and Communication. Birmingham, UK, 2019: 180-189

[197] Sadlek L, Čeleda P, Tovarňák D. Identification of attack paths using kill chain and attack graphs//Proceedings of the NOMS 2022—2022 IEEE/IFIP Network Operations and Management Symposium. Budapest, Hungary, 2022: 1-6

[198] Kaiser F K, Dardik U, Elitzur A, et al. Attack hypotheses generation based on threat intelligence knowledge graph. IEEE Transactions on Dependable and Secure Computing, 2023, 20(6): 4793-4809

[199] Kuhn D R, Raunak M S, Kacker R. An analysis of vulnerability trends, 2008－2016//Proceedings of the 2017 IEEE International Conference on Software Quality, Reliability and Security Companion(QRS-C). Prague, Czech Republic, 2017: 587-588

[200] Murtaza S S, Khreich W, Hamou-Lhadj A, et al. Mining trends and patterns of software vulnerabilities. The Journal of Systems and Software, 2016, 117: 218-228

[201] Tang M, Alazab M, Luo Y. Big data for cybersecurity: Vulnerability disclosure trends and dependencies. IEEE Transactions on Big Data, 2019, 5: 317-329

[202] Williams M A, Dey S, Barranco R C, et al. Analyzing evolving trends of vulnerabilities in national vulnerability database//Proceedings of the 2018 IEEE International Conference on Big Data (Big Data). Seattle, USA, 2018: 3011-3020

[203] Chang Y Y, Zavarsky P, Ruhl R, et al. Trend analysis of the CVE for software vulnerability management//Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing. Boston, USA, 2011: 1290-1293

[204] Bo L, Meng X, Sun X, et al. A comprehensive analysis of NVD concurrency vulnerabilities//Proceedings of the 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS). Guangzhou, China, 2022: 9-18

[205] Bode M A, Oluwadare S, Alese B K, et al. Risk analysis in cyber situation awareness using Bayesian approach// Proceedings of the 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA). London, UK, 2015: 1-12

[206] Chen K, Zhu J, Han L, et al. A novel network security situation awareness model for advanced persistent threat// Proceedings of the 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC). Guilin, China, 2022: 9-16

[207] Endsley M. Design and evaluation for situation awareness enhancement. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 1988, 32(2), 97-101

[208] Chen Gang, Zhao Yu-Qian. RF-SVM based awareness algorithm in intelligent network security situation awareness system//Proceedings of the 3rd Workshop on Advanced Research and Technology in Industry. Guilin, China, 2017: 224-228

[209] Kou G, Wang S, Tang G. Research on key technologies of network security situational awareness for attack tracking prediction. Chinese Journal of Electronics, 2019, 28(1): 162-171

[210] Li X, Chen J, Lin Z, et al. A mining approach to obtain the software vulnerability characteristics//Proceedings of the 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD). Shanghai, China, 2017: 296-301

[211] Alperin K, Joback E, Shing L, et al. A framework for unsupervised classificiation and data mining of tweets about cyber vulnerabilities. arXiv: abs/2104.11695. 2021

[212] Alexopoulos N, Meneely A, Arnouts D, et al. Who are vulnerability reporters? A large-scale empirical study on floss//Proceedings of the 15th ACM/IEEE International

Symposium on Empirical Software Engineering and Measurement (ESEM). New York, USA, 2021: 1-12

[213] Munaiah N, Meneely A. Vulnerability severity scoring and bounties: Why the disconnect?//Proceedings of the 2nd International Workshop on Software Analytics. New York, USA, 2016: 8-14

[214] Votipka D, Stevens R, Redmiles E M, et al. Hackers vs. testers: A comparison of software vulnerability discovery processes//Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP). San Francisco, USA, 2018: 374-391

[215] Li H, Zhao L, Xi R. Study on the distribution of CVSS environmental score//Proceedings of the IEEE International Conference on Electronics Information and Emergency Communication. Beijing, China, 2015: 122-125

[216] Kudjo P, Chen J, Brown S A, et al. The effect of weighted moving windows on security vulnerability prediction//Proceedings of the 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW). San Diego, USA, 2019: 65-68

[217] Kitchenham B, Pfleeger S, McColl B, et al. An empirical study of maintenance and development estimation accuracy. The Journal of Systems and Software, 2002, 64: 57-77

[218] Shahzad M, Shafiq M, Liu A. Large scale characterization of software vulnerability life cycles. IEEE Transactions on Dependable and Secure Computing, 2020, 17: 730-744

[219] Arora A, Krishnan R, Nandkumar A, et al. Impact of vulnerability disclosure and patch availability—An empirical analysis//Proceedings of the 3rd Workshop on the Economics of Information Security. Minneapolis, USA, 2004, 24: 1268-1287

[220] Abedin M, Nessa S, Al-Shaer E, et al. Vulnerability analysis for evaluating quality of protection of security policies//Proceedings of the 2nd ACM Workshop on Quality of Protection (QoP'06). New York, USA, 2006: 49-52

[221] Bhuiyan F A, Shakya R, Rahman A. Can we use software bug reports to identify vulnerability discovery strategies?//Proceedings of the 7th Symposium on Hot Topics in the Science of Security. New York, USA, 2020: 1-10

[222] Woo S, Lee D, Park S H, et al. V0Finder: Discovering the correct origin of publicly reported software vulnerabilities//Proceedings of the 30th USENIX Security Symposium (USENIX Security 21). Vancouver, Canada, 2021: 3041-3058

[223] Xiong Q, Lian S, Zeng Z, et al. An empirical analysis of vulnerability information disclosure impact on patch R&D of software vendors. Journal of Intelligent & Fuzzy Systems, 2022, 44: 839-853

[224] Ruohonen J, Hyrynsalmi S, Leppänen V. Software vulnerability life cycles and the age of software products: An empirical assertion with operating system products//Proceedings of the International Conference on Advanced Information Systems Engineering. Ljubljana, Slovenia, 2016: 207-218

[225] Lin Z, Li X, Kuang X. Machine learning in vulnerability databases//Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID). Hangzhou, China, 2017, 1: 108-113

CAO Xu-Dong, Ph.D. candidate. His research interests include network and system security.

HUANG Zai-Qi, M.S. candidate. His research interests include artificial intelligence and information security.

CHEN Yu-Jie, M.S. candidate. His research interests include artificial intelligence and information security.

WANG Wen-Jie, Ph.D., associate professor. His research interests include information security and intelligent information processing.

SHI Hui-Yang, Ph.D., senior engineer. Her research interests include network and system security.

LI Shu-Hao, Ph.D., senior engineer. His research interests include threat detection and information confrontation, network attack and defense technology.

ZHANG Yu-Qing, Ph.D., professor, Ph.D. supervisor. His research interests include network and system security.

Background

As an important part of the information security infrastructure, the vulnerability database can not only save the basic information of various vulnerabilities, but also quickly respond to the latest vulnerabilities and spread them in time, improving the public's ability to deal with information security threats. With the sharp increase in the number of vulnerabilities and the speed of discovery in recent years, the construction of vulnerability databases has received more and more attention.

Moreover, in recent years, machine learning, natural language processing and other technologies have been continuously developed, and the application of artificial intelligence to various stages of vulnerability database construction and application has become a research hotspot and focus in the computer field.

This paper is the first to investigate and summarize the application of artificial intelligence technology in the construction and application of vulnerability databases. First of all, focusing on the challenges faced in vulnerability collection, processing, management and other stages of vulnerability database construction, the author introduces the achievements of existing research, besides, the author also introduces the research related to the quality assessment of the vulnerability database, and summarizes the significant role of artificial intelligence technologies in the construction of vulnerability databases. Secondly, the author categorizes the application research of the vulnerability database in the field of computer science into six types, including vulnerability detection and discovery, vulnerability repair, software security assessment, network attack modeling, security situation analysis and exploratory research based on vulnerability reports, on this basis, the author respectively introduces the current applications and research innovations of the vulnerability database. Finally, the author analyzes and summarizes the existing deficiencies and challenges in the construction and application of the vulnerability database, including how to obtain more comprehensive vulnerability data and solve multi-source heterogeneous data fusion, how to carry out accurate quality assessment, how to consider vulnerability priority to support repair work, how to conduct exploratory discoveries around the patterns exhibited by vulnerabilities and the relationships between vulnerabilities, etc. The author also looks forward to possible research directions in the future, so as to provide reference guidance for subsequent scholars to understand or study vulnerability databases and vulnerability data, and provide some ideas for applying artificial intelligence technology to the security field.

This work is supported by the National Key Research and Development Program of China (Nos. 2023YFB3106400, 2023QY1202), the National Natural Science Foundation of China (Nos. U2336203, U1836210), the Key Research and Development Science and Technology of Hainan Province (No. GHYF2022010), and the Beijing Natural Science Foundation (No. 4242031).