开源软件漏洞库综述

贾培养[1,2] 孙鸿宇[1,2] 曹婉莹[2] 伍高飞[1,2] 王文杰[2]
[1](西安电子科技大学网络与信息安全学院 西安 710126)
[2](中国科学院大学国家计算机网络入侵防范中心 北京 101408)
(jiapy@nipc.org.cn)

Open Source Software Vulnerability Data Base Overview

Jia Peiyang[1,2], Sun Hongyu[1,2], Cao Wanying[2], Wu Gaofei[1,2], and Wang Wenjie[2]
[1](*SchooL of Cyber Engineering, Xidian University, Xi'an* 710126)
[2](*National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing* 101408)

Abstract In recent years, with the continuous shortening of the software development cycle, a large number of open source code is used in modern software projects, and software developers tend to focus only on the security of the part of the project code they are responsible for, and rarely pay attention to the security of the open source code used in the project, and it is difficult for users to correspond the vulnerability entries in the traditional vulnerability repository to the current software version. There are some differences between the vulnerabilities existing version control schemes and those of open source code, so a vulnerability repository that can accurately collect open source code vulnerability intelligence and precisely match vulnerabilities is essential. This paper first introduces the potential security challenges brought by the widespread use of open source code, then analyzes in detail the existing open source vulnerability repository platforms and conducts a comparative study of existing open source vulnerability databases from several dimensions, then gives the problems and challenges faced by the construction of current open source vulnerability databases, and finally gives some suggestions for building open source vulnerability databases.

Key words open source software; vulnerability database; version control; open source vulnerability intelligence; open source vulnerability database

摘 要 近年来,随着软件开发周期的不断缩短,越来越多的软件开发者在其项目代码中使用大量的开源代码.软件开发者往往只关注自己负责部分的代码安全,几乎不关注项目采用的开源代码的安全性问题.开源软件的使用者很难将传统漏洞数据库中的漏洞条目对应到当前的软件版本中.现有的漏洞版本控制方案和开源代码的版本控制方案之间存在一定的差异,这使得开源代码使用者无法及时修复存在漏洞的代码,因此一个能够准确收集开源漏洞情报并对漏洞进行精确匹配的漏洞库是

收稿日期:2021-05-25
基金项目:国家自然科学基金重点项目(U1836210)

十分必要的.首先介绍了开源代码的广泛使用带来的潜在安全挑战;其次对现有开源漏洞库平台的情况进行了详细分析,同时对现有开源漏洞库从多个维度进行了对比研究;然后给出了当前开源漏洞库建设面临的问题和挑战;最后给出了建设开源漏洞数据库的一些建议.

关键词 开源软件;漏洞数据库;版本控制;开源漏洞情报;开源漏洞库

中图法分类号 TP393.08

随着开源代码在代码库中的比重不断增加,几乎所有企业通过使用免费的开源代码来开发自己的软件产品.开源代码能够帮助软件开发企业和组织极大地缩短软件开发的周期.由此可见开源代码已经成为软件供应链的重要组成部分,也成为了当今网络空间的基石.然而,开源软件的安全问题并没有得到足够的重视.根据开源安全管理平台White Source 在 2020 年发布的《开源漏洞安全现状》,开源代码和开源社区在过去的几年中迎来了大规模的增长,与此同时开源漏洞也随之爆发式的增长.

开源代码是指用户可以根据许可进行访问和改进的源代码.开源软件和专有软件的区别在于,用户第 1 次使用专有软件时必须签署同意最终用户许可协议(EULA),同时这些协议会限制用户共享或修改产品本身,用户在使用开源软件时则需遵守开源许可条款,开源许可证控制着除发起者以外的其他人如何使用、修改和共享软件.从开源软件使用者的角度出发开源软件具有以下几个优势:

1) 具有较高的灵活性.开源使用者可以根据自身需要对软件进行定制,以满足自身的需要,开源使用者被赋予了更多的控制权.

2) 更好的社区支持.开源软件通常具有论坛和开发人员组成的社区,这些社区会针对开源软件出现的问题进行修复和更新,从社区中往往能够获得对开源软件改进的支持.

3) 稳定性.开源软件的使用者可以放心地将开源软件应用于周期较长的项目,因为即使开源软件的作者停止使用它,开源软件本身不会从市场上消失.

4) 相对较低的费用.开源软件的购买成本往往低于同类型专有软件的购买成本,甚至大量的开源软件是免费的.

当前开源软件的使用者普遍存在以下几个误区:

1) 开源软件是免费的.开源并不意味着免费,这是很多开源使用者的误区,开源代码的免费指的是软件可以用于个人目的,其中开源许可证给出了开源软件的使用和修改的条款,必须在开源许可证的许可下合理地使用开源软件.

2) 开源软件相对于专有软件更加安全.许多攻击者利用开源软件在软件供应链中不可或缺的地位,通过依赖项管理器在软件生命周期中的自动解析,造成了针对软件供应链的攻击[1].

3) 开源软件的使用不用遵守任何规范.很多开源软件使用者认为开源软件可以任意使用,而不用遵守任何规范,这是一个很大的误区.错误地使用没有获得许可的开源软件可能会带来知识产权方面的诉讼,使用开源软件必须遵守相关的软件许可的条款[2].

奇安信代码安全实验室在 2019 年针对联网设备固件中引用的开源软件的检测和漏洞分析,发现这些固件中 86.4%存在至少 1 个以上的开源漏洞,漏洞最多的固件存在 74 个老旧开源漏洞,并且这些固件 100%使用了开源代码.88%的项目因使用开源软件引入了安全漏洞.根据 2020 年 Github 的 Octoverse 状态报告,目前的软件代码库中对开源代码、组件和库的依赖相较于以前更加普遍.同时漏洞管理公司 RiskSense 在其《开源代码中的黑暗现实》报告中明确指出开源漏洞发现后,通常需要很长时间才能添加到 NVD(美国国家漏洞数据库)中.从公开披露到将漏洞纳入 NVD 平均时间为 54 天,有的甚至超过 1 年,这给使用开源软件的相关组织带来了极大的安全隐患.因此,一个专门针对开源漏洞的开放型数据库是十分必要的.

从开源漏洞的特点出发,针对开源漏洞的漏洞数据库应该具备以下几个特点:

1）提供精确的开源漏洞元数据，提高使用者漏洞查询的准确性，包括准确的受影响版本等基本信息；

2）能够进行代码检测，并给出输入代码中开源组件的使用情况和相互依赖关系；

3）更新较为及时，防止漏洞收录延迟带来的不利影响；

4）较为简单的漏洞报告过程，减少发布漏洞所需的工作量；

5）能够提供漏洞引入和修复位置的精确数据.

针对开源漏洞的研究正在广泛展开，但是很少有文章对开源漏洞库进行针对性的研究，本文对开源漏洞库进行了系统性概述.

1 现有开源漏洞库介绍

通过开源漏洞库，开源软件的使用者可以查找到其应用程序中存在漏洞的开源组件的版本、漏洞的类型、漏洞的 CVSS 分数、漏洞的 POC 证明等信息.本节就当前主流的开源漏洞库进行介绍.

1.1 Open Source 漏洞数据库

Open Source 漏洞数据库（open source vulnerabilities，OSV）[3] 是开放源代码的漏洞数据库，该漏洞库是谷歌公司于 2021 年 5 月推出的一个开源项目，旨在帮助开源代码的使用者更好地维护和管理其项目中出现的开源漏洞.该漏洞库开放了一个 API，该 API 接受用户通过 git commit 指令获得的哈希号，然后返回给用户该版本存在的漏洞列表.通过该 API 用户可以查询其软件版本是否受到开源漏洞的影响.OSV 的特点在于向开源维护者提供了自动化的漏洞分类方案，开源维护者只需通过 OSV 提供的 API 即可对开源代码进行自动地分析并给出受影响的版本号和版本范围.OSV 的数据集目前主要来自于 OSS-Fuzz[4] 发现的漏洞，其中 OSS-Fuzz 是针对开源软件的持续模糊测试服务，当前已经集成了 350 多个开源项目，并发现了 25 000 多个 Bugs.图 1 是 OSS-Fuzz 对开源软件进行模糊测试的原理图：

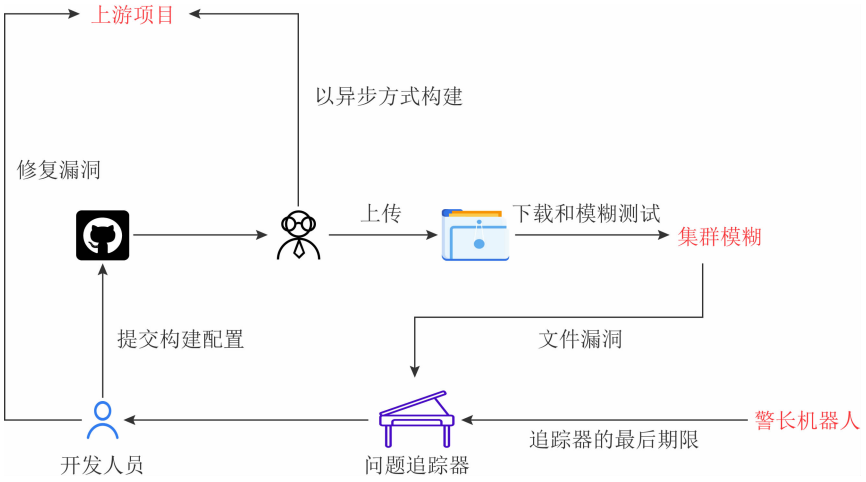

图 1 OSS-Fuzz 原理图

OSV 漏洞库通过二分法的查找方式来查找造成漏洞的准确提交版本和修复该错误的准确版本，从而确定该错误提交影响的版本范围.OSV 相较于现有的 CVE 数据库，起到了很好的补充作用.OSV 的目标是更好地对开源漏洞进行分类、存储和查询.OSV 为了方便自动化工具的调用，对漏洞条目使用了较为简单的描述.

OSV 对开源漏洞的字段描述（以 OSV-2021-534 为例）如表 1 所示.

1.2 WhiteSource 漏洞数据库

WhiteSource 漏洞数据库（WhiteSource vulnerability database，WSVD）[5] 是由美国权威开源安全管理平台 WhiteSource 推出的开源漏洞库，该公司致力于为用户提供开源安全服务和许可证合规管理.该公司通过自动连续扫描数十个开源代码库来帮助用户寻找解决方案，可以自动警告用户正在使用的开源组件中的已知安全漏洞、错误、新版本、补丁等信息，同时该公司提供的开源安全

服务具有较高的兼容性，可以与多种编程语言、构建工具和开发环境兼容，可以使得开源软件使用者无需将精力放在开源组件的安全性上，而是将精力集中在对软件的定制开发上.

表 1　OSV 漏洞字段描述(OSV-2021-534)

字段信息	数据库中的位置	案例
包文件名	Package	OSS-Fuzz/tesseract-ocr
仓库地址	Repo URL	https://github.com/tesseract-ocr/tesseract
漏洞概述	Summary	Container-overflow intesseract::ExtractResults16
漏洞细节	Details	OSS-Fuzz report: https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=32142
漏洞严重程度	Severity	HIGH
引入版本	Introduced in	2a3682a35e643cefb86eefa4c9a3deddc75295bd
修复版本	Fixed in	91b2b4f4a08d4693b02838636c53a2af93397138
受影响的版本	Affected versions	5.0.0-alpha-20210401
参考链接	References	https://bugs.chromium.org/p/oss-fuzz/issues/detail?id=32142

WSVD 漏洞库允许使用者通过 CVE 编号或项目名称检索开源漏洞，并在网站首页给出了近 90 天内危险系数最高的开源漏洞排名.该漏洞库宣称覆盖了 200 多种编程语言和 300 万个开源组件，同时数据库来源包括 NVD、安全公告、开源项目问题追踪器等多种渠道.同时在漏洞库的资源中心，该公司免费提供了针对开源漏洞的白皮书、网络研讨会和研究报告，为研究开源漏洞提供了便利.

WSVD 允许用户通过邮件的形式提交开源漏洞，同时在收录未经 CVE/NVD 等数据库收录的漏洞时，会将该漏洞进行重新编号，进行单独管理.WhiteSource 使用软件组成分析(SCA)技术来帮助用户检测软件项目中的开源组件的使用情况.SCA 可以对应用程序的代码进行扫描，这些扫描包括注册表在内的多种资源，以准确识别所有开应用程序中的所有开源组件、许可证合规和安全漏洞.SCA 工具会生成目标应用程序中所有开源组件(包括直接和间接依赖项)的清单报告，然后根据开源组件的清单查询到关于这些开源组件的开源许可证信息，以便于开源维护者判断许可证是否与自己的应用程序相兼容.先进的 SCA 解决方案可以让开源组件的选择、使用和跟踪变得更加自动化.如果只是通过开源安全维护人员手动管理日益增多的开源漏洞，其工作量是难以想象的，WhiteSource 提供的 SCA 解决方案可以极大地降低开源安全维护者的工作量，是一个较为先进的开源安全解决方案.SCA 的工作原理如图 2 所示.

WSVD 对漏洞的描述主要包括该漏洞的程序语言类型、CWE 类型、CVSS 分数(CVSS V2 和 CVSS V3)、发布日期、参考链接、修复信息等.以

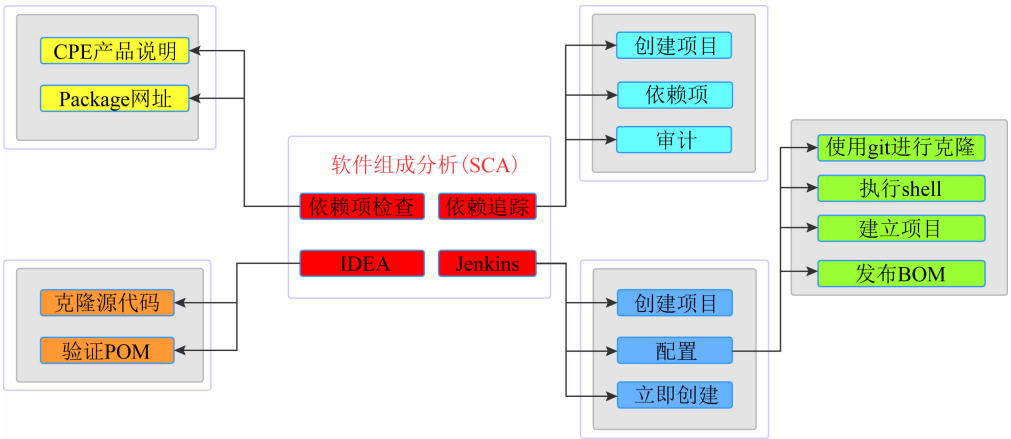

图 2　SCA 工作原理图

CVE-2021-29475 为例，漏洞字段描述如表 2 所示：

表 2　WSVD 漏洞字段描述(CVE-2021-29475)

字段信息	数据库中的位置	案例
漏洞编号	Title	CVE-2021-29475
发布日期	Date	April 26, 2021
威胁分数	Severity Score	10.0
漏洞类型	Weakness Type (CWE)	CWE-94, CWE-918
最新修复信息	Top Fix	Upgrade to version 1.5.0
程序语言类型	Language	TYPE_SCRIPT
参考链接	Related Resources	https://nvd.nist.gov/vuln/detail/CVE-2021-29475
CVSS v3.1	CVSS v3.1	10.0
CVSS v2	CVSS v2	5.8

1.3　Snyk 漏洞数据库

Snyk 漏洞数据库(Snyk vulnerability database, SVD)[6]是由专注于开源安全的 Snyk 公司推出的开源漏洞数据库，Snyk 通过扫描程序的方式来发现应用程序中的开源组件漏洞和许可证违规.SVD 开源漏洞库在检索漏洞方面允许用户通过开源组件的名字或 CVE 编号来检索漏洞，方便用户查看其使用的开源组件中是否含有漏洞.SVD 提供了测试代码的入口，通过该入口用户可以检测其开源项目中是否包含开源漏洞.

SVD 的开源漏洞数据来源较为广泛，包括现有的 NVD，CVE 数据库，监视 Github 上可能出现的漏洞、PR 等信息，同时 SVD 还通过手动审核的方式来检测被广泛使用的软件包中是否含有安全漏洞，通过聆听有关安全公告、JIRA 公告板、Github 提交等可能出现的尚未报告的漏洞.其次与社区合作，通过漏洞赏金的方式来获得最新的漏洞披露，其团队还与学术实验室合作，以交换工具、方法和数据.SVD 开源漏洞库中漏洞字段的描述如表 3 所示：

表 3　SVD 漏洞字段描述(以软件包 bundler 为例)

字段信息	数据库中的位置	案例
漏洞类型	Title	Arbitrary Code Execution
受影响的版本	versions	≥1.14.0, <2.1.0
漏洞概述	Overview	Affected versions of this package are vulnerable to Arbitrary Code Execution.
漏洞修复	Remediation	Upgrade bundler to version 2.1.0 or higher.
参考链接	References	https://github.com/rubygems/bundler/commit/65cfebb041c454c246aaf32a177b0243915a9998
CVSS 分数	CVSS SCORE	7.0
CWE 类型	CWE	CWE-94
参考 CVE	CVE	CVE-2019-3881

1.4　Veracode 漏洞数据库

Veracode 漏洞数据库(Veracode vulnerability database, VVD)[7]是一家专注于应用程序安全测试平台推出的开源漏洞库，该漏洞库可以根据开源软件包名或者 CVE 编号查询有关的开源漏洞信息，同时能够根据漏洞所属的编程语言类型或者操作系统来筛选可能的开源漏洞.

VVD 可以识别不在国家漏洞数据库(NVD)或尚未纳入 NVD 的漏洞，为了发现这些漏洞，VVD 会搜索所有的开源代码存储库，通过扫描代码、元数据、提交日志、错误修复、补丁说明和开发人员注释，使用机器学习算法来发现尚未公开的安全问题.这使得该数据库领先于网络攻击者.当漏洞在 NVD 公布时，对于组织和攻击者而言，有足够的时间在漏洞修复之前发起攻击.

VVD 漏洞库的系统结构如图 3 所示.

从图 3 我们可以看出，该漏洞库的来源包括以下几个方面：

1) 漏洞提交；

2) JIRA tickets；

3) Bugzilla 报告；

4) 数千个开源库的 GitHub 问题和请求.

对于漏洞数据的不同来源，该漏洞库都相对应地建立了训练有素的机器学习模型，以准确识别来源中的每个项目是否与漏洞相关，识别出的漏洞将发送到其漏洞管理平台进行审查，然后将

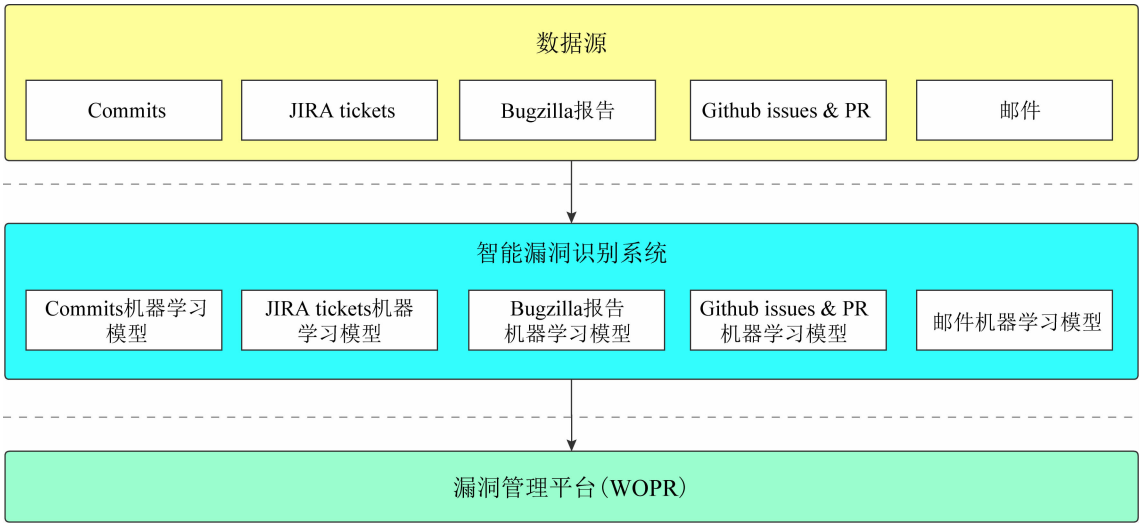

图 3　VVD 漏洞库的系统结构图

其添加到威胁情报中心.VVD 漏洞库中漏洞字段的描述如表 4 所示(以 firefox 为例):

表 4　VVD 漏洞字段描述(以软件包 firefox 为例)

字段信息	数据库中的位置	案例
软件包名	Title	firefox Vulnerability Data
危险等级	CVSS v2	HIGH
软件包系统	OS	RPM
版本号	Versions	479
最新版本	Latest version	78.10.0—1.el8_3
漏洞总数	Vulnerabilities	1028
受影响的版本	Versions Affected	461
安全版本	Safe Versions	18
高危漏洞数	High RiskVulns	536

2　开源漏洞库对比研究

在第 1 节对 4 个开源漏洞库单独介绍的基础上,本文详细对比了不同开源漏洞库在漏洞来源、漏洞管理、漏洞分析处理等方面的不同.

2.1　漏洞来源

Open Source 漏洞数据库(OSV)中收录的开源漏洞主要来自于 OSS-Fuzz 项目发现的漏洞.该项目旨在通过连续模糊测试来发现开源项目中可能存在的漏洞,通过该项目发现的开源漏洞统一进行单独的 OSV 编号.这些漏洞并没有和现有的 CVE/NVD 等权威漏洞数据库进行关系映射,所以 OSV 漏洞库目前收录的开源漏洞数量有限,且来源较为集中.

WhiteSource 漏洞数据库(WSVD)的漏洞主要来源包括 NVD,OSVDB 等漏洞库,同时 WSVD 通过跟踪开源项目的问题追踪器和安全公告来获取可能的开源漏洞,NVD 是其漏洞的主要来源.WSVD 漏洞库中收录的漏洞多为开源漏洞,漏洞库的针对性很强.

Snyk 漏洞数据库(SVD)的漏洞来源较为广泛,包括:手动审核现有开源软件包中的漏洞;监视开源代码存储库中可能带来漏洞的 issue 和 PR;监听开源项目安全公告、社区合作、学术界合作等方式来获取最新的开源漏洞.

Veracode 漏洞数据库(VVD)获取开源漏洞的方式包括用户漏洞的提交、漏洞报告、扫描 Github 等开源代码库的 issue 和 Pull Request、缺陷跟踪管理系统 JIRA 的用户 bug,VVD 在获取开源漏洞方面相较于其他开源漏洞库并未从现有的漏洞库中获取漏洞,其获取的开源漏洞具有较好的独立性.

当前开源漏洞库获取开源漏洞的主要方式如表 5 所示.

从表 5 我们可以看出当前开源漏洞库的主要来源有哪些,其中以 SVD 为代表的开源漏洞库的漏洞来源较为广泛,这些获取方式为获取开源漏洞提供了新思路.

表 5　当前开源漏洞库获取开源漏洞的主要方式

开源漏洞来源	OSV(Google)	WSVD(WhiteSource)	SVD(Snyk)	VVD(Veracode)
模糊测试	有	无	无	无
其他权威漏洞库	无	有	无	无
问题追踪器[8]	无	有	有	有
安全公告	无	有	有	有
手动审核	无	无	有	无
监视开源代码存储库[9]	无	无	有	有
社区合作	无	无	有	无
漏洞邮件提交	有	有	有	有

2.2　应用方向

2.1 节介绍的 4 个开源漏洞库目标都是及时有效地跟踪开源漏洞，但是每一个漏洞库的侧重方向各有不同，下文将详细介绍每个开源漏洞库的主要应用方向.

Open Source 漏洞数据库(OSV)旨在帮助开源项目的使用者通过 API 来查询其正在使用的开源项目版本是否存在漏洞，其 API 对外开放了 POST 请求和 GET 请求. POST 请求需要通过 json 的格式发起请求，该请求主要接收 3 个参数，分别是 commit，version，package，这里的 commit 指的是开源代码提交到代码库中的哈希值，version 则表示该开源代码的版本号，package 可以是一个对象，这个对象接收开源软件的包名和 ecosystem，该 API 的结构树如图 4 所示. 同时 OSV 的另一个主要应用方向是帮助开源维护者自动化地对开源漏洞进行分类，如果没有自动化的漏洞分类，对呈现海量增长趋势的开源漏洞来说将是一个难以想象的工作量[10]. 通过 OSV 的自动分类

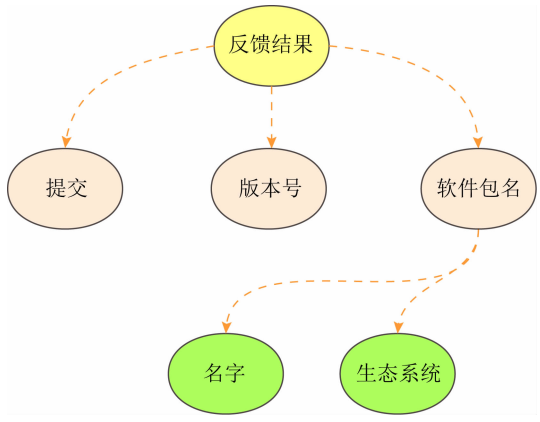

图 4　OSV 开源漏洞查询 API 结构树

可以将开源漏洞分类的工作量降低很多，极大提高了分类效率.

WhiteSource 漏洞数据库(WSVD)主要优势在于汇总了数百个开源社区资源，并兼容了多种编程语言和数以万计的开源组件. 该漏洞库的特点是数据来源广泛，且多为开源漏洞，是收录开源漏洞数量较多的数据库之一.

Snyk 漏洞数据库(SVD)相较于其他的开源漏洞库在提供基本的查询功能之外，还向用户提供检测开源代码的 API. 通过该接口，用户登录后即可在线检测其开源代码中是否存在开源漏洞，同时在开源漏洞的字段描述上，该漏洞库提供了详细的受影响版本等信息. 提供在线代码检测是该漏洞库的特点之一.

Veracode 漏洞数据库(VVD)相较于其他开源漏洞库而言，其收集到可能出现开源漏洞的数据之后，会通过其内部的机器学习模型去识别这些可能的漏洞. 一旦漏洞被识别，将添加到其漏洞管理中心，将机器学习算法应用到漏洞检测是近年来开源漏洞检测的新趋势[11]. 该漏洞库根据不同的数据来源建立了不同的机器训练模型，可以极大地提高检测效率，有些漏洞的发现甚至早于 NVD 等数据库. 因此，通过机器学习算法来检测开源漏洞是该漏洞库的特点之一.

2.3　漏洞分析处理

Open Source 漏洞数据库(OSV)通过二分法(bisects)的方式来找到开源漏洞错误提交的版本和错误修复的版本，bisects 在查找错误的提交时首先启动 bisect 进程，然后输入包含错误的提交版本，其次输入已知的修复版本，运行代码以查看错误是否仍然存在，返回 bisects 结果，然后一直

重复前 2 项直到找到错误的提交版本.bisects 查找错误提交的基本流程如图 5 所示：

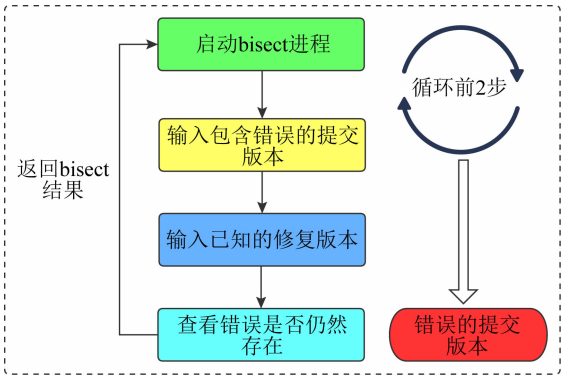

图 5　bisects 查找错误提交的基本流程

WhiteSource 漏洞数据库(WSVD)通过 SCA 技术来扫描开源组件中可能出现的漏洞，可以让开源代码使用者快速跟踪和分析引入到项目中的任何开源组件，SCA 工具在查找项目中使用的组件以及它们的依赖库时，还会检测相关软件许可证，并给出不赞成使用的依赖项和可能的漏洞利用.

Snyk 漏洞数据库(SVD)通过依赖项扫描程序可以主动地发现开源依赖项中的漏洞和许可证违规.在漏洞扫描方面，SVD 使用了 SCA 技术，该技术可以发现开源软件包中复制的代码片段，极大地提高了漏洞发现的准确性.

Veracode 漏洞数据库(VVD)在漏洞分析处理方面通过静态应用程序安全测试工具(SAST)来分析源代码和代码的编译版本，以此来发现开源项目中的漏洞.它可以向开发人员提供有关在代码编写过程中可能引入的代码错误的即时反馈，这种反馈相比于软件开发周期后期才发现漏洞是十分有效的.

3　问题和挑战

上文对不同开源漏洞库之间的差异和侧重点进行了对比分析，主要介绍了每个开源漏洞库的数据来源、特点和应用方向，下面介绍开源漏洞库旨在解决的共同问题和挑战.

自动化的开源漏洞检测.面对日益增长的开源漏洞数量[12]，当前的开源漏洞库都在尝试使用自动化的开源漏洞检测技术[13]来跟踪开源漏洞.因为手动审核开源项目中的漏洞需要考虑到开源项目中不断更新的版本和当前项目中的版本是否一致[14]，同时还要考虑开源项目中复杂的依赖关系[15]，仅仅依靠人工几乎是不可能的.

持续跟踪开源漏洞情报[16].只有做到快速的更新开源漏洞的信息，才能够有效地避免针对开源漏洞的软件供应链攻击[17]，当前的开源漏洞库对不同数据来源的开源代码进行监测，以期获得对开源漏洞情报的持续跟踪.

理清开源项目之间的依赖关系.开源漏洞之所以较难以处理，和其项目中较为复杂的依赖有关.只有建立了开源项目中的依赖关系树，对开发者而言才能够清晰地知道自己的项目中使用了开源项目的哪些版本，哪些版本是安全的，只有这样才是解决开源安全问题行之有效的方案.

4　结束语

近年来，随着开源漏洞数量的不断增长，针对软件供应链的攻击事件层出不穷.通过开源漏洞库获取正在使用的开源项目中的漏洞信息是我们防止此类攻击事件的有效方案.本文对当前开源安全的现状、开源漏洞库的基本情况进行了系统性的介绍，并对当前的开源漏洞库从漏洞来源、应用方向和漏洞分析处理等方面进行了详细的对比分析，最后对开源漏洞库待解决的问题和面临的挑战进行了归纳.

参　考　文　献

[1] Ohm M, Plate H, Sykosch A, et al. Backstabber's knife collection: A review of open source software supply chain attacks[C] //Proc of Int Conf on Detection of Intrusions and Malware, and Vulnerability Assessment. Cham: Springer, 2020: 23-43

[2] 李震宁，刘莉，孟杰. 开源软件商业化中面临的知识产权风险[J]. 网络空间安全，2020，9(8)：42-44

[3] OSV 漏洞数据库[EB/OL]. [2021-05-19]. https://osv.dev/

[4] OSS-Fuzz [EB/OL]. [2021-05-17]. https://github.com/google/oss-fuzz

[5] WhiteSource 漏洞数据库[EB/OL]. [2021-05-20]. https://www.whitesourcesoftware.com/vulnerability-database/

[6] Snyk 漏洞数据库[EB/OL]. [2021-05-20]. https://snyk.io/vuln

[7] Veracode 漏洞数据库[EB/OL]. [2021-05-20]. https://www.sourceclear.com/vulnerability-database

[8] Rath M, Mäder P. Request for comments: Conversation patterns in issue tracking systems of open-source projects [C] //Proc of the 35th Annual ACM Symp on Applied Computing. New York: ACM, 2020: 1414-1417

[9] Campbell D, Cabrera-Diego L A, Korkontzelos Y. What is the message about? Automatic multi-label classification of open source repository messages into content types [C] //Proc of Joint European-US Workshop on Applications of Invariance in Computer Vision. Berlin: Springer, 2020: 520-531

[10] 郭雪, 孔松, 王皓月. 企业级开源风险及治理模式研究[J]. 信息通信技术与政策, 2020, 46(5): 45-48

[11] 孙鸿宇, 何远, 王基策, 等. 人工智能技术在安全漏洞领域的应用[J]. 通信学报, 2018, 39(8): 1-17

[12] 冯兆文, 刘振慧. 开源软件漏洞安全风险分析[J]. 保密科学技术, 2020 (2): 27-32

[13] 赵尚儒, 李学俊, 方越, 等. 安全漏洞自动利用综述[J]. 计算机研究与发展, 2019, 56(10): 73-87

[14] Dong Y, Guo W, Chen Y, et al. Towards the detection of inconsistencies in public security vulnerability reports [C] //Proc of the 28th USENIX Security Symp. Berkeley, CA: USENIX Association, 2019: 869-885

[15] Decan A, Mens T, Grosjean P. An empirical comparison of dependency network evolution in seven software packaging ecosystems [J]. Empirical Software Engineering, 2019, 24 (1): 381-416

[16] Ponta S E, Plate H, Sabetta A. Detection, assessment and mitigation of vulnerabilities in open source dependencies [J]. Empirical Software Engineering, 2020, 25(5): 3175-3215

[17] Vu D L, Pashchenko I, Massacci F, et al. Towards using source code databases to identify software supply chain attacks [C] //Proc of the 2020 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 2093-2095

贾培养

硕士研究生.主要研究方向为网络空间安全.

jiapy@nipc.org.cn

孙鸿宇

博士研究生.主要研究方向为信息安全与机器学习.

sunhy@nipc.org.cn

曹婉莹

博士研究生.主要研究方向为网络空间安全.

caowy@nipc.org.cn

伍高飞

博士,硕士生导师.主要研究方向为密码学.

wugf@nipc.org.cn

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.

wangwj@gucas.ac.cn