 DOI: https://doi.org/10.26549/bdai.v4i3.13864

Discussion on Open Source API Code Security Detection

Bo Lu Shicong Song Yan Yang Ruipeng Zhou

Xinjiang Photoelectric Cloud Information Technology Co., Ltd., Urumqi, Xinjiang, 830000, China

Abstract

With the wide application of open source API, it is particularly important to ensure its code security. The purpose of this paper is to explore the problems and methods related to the security detection of open source API code. Through this study, we hope to improve the developers' understanding of the security of open source API code, and promote the further development of relevant research and tools, so as to ensure the security and stability of the software system.

Keywords

open source API; code security; static code analysis; dynamic code analysis

开源 API 代码安全性检测相关阐述

鲁博 宋仕聪 杨艳 周瑞鹏

新疆光电云信息技术有限公司，中国·新疆 乌鲁木齐 830000

摘 要

随着开源API的广泛应用，保障其代码安全性变得尤为重要。论文旨在探讨开源API代码安全性检测相关的问题和方法。通过本研究，希望提高开发者对开源API代码安全性的认识，并促进相关研究和工具的进一步发展，以保障软件系统的安全性和稳定性。

关键词

开源API；代码安全性；静态代码分析；动态代码分析

1 引言

随着信息技术的快速发展和互联网的普及，开源 API（Application Programming Interface）作为软件开发中的重要组成部分，扮演着连接不同软件和服务的桥梁。开源 API 的使用不仅能够提高软件开发的效率和灵活性，还可以促进开发者之间的合作和知识共享。然而，随之而来的是对开源 API 代码安全性的关注和需求。通过论文的研究，我们希望提高开发者对开源 API 代码安全性的认识，加强对潜在安全风险的警惕，并为开源 API 代码的安全性检测提供有益的指导和启示。

2 开源 API 的定义和特点

2.1 开源 API 概述

开源 API（Application Programming Interface）是指为软件开发者提供的一组定义和规范，用于在应用程序之间进行交互和通信。它可以看作是不同软件组件之间的桥梁，提供了一种标准的方式来访问和利用特定功能或服务，见图 1。开源 API 的设计目的是促进软件的可重用性、互操作性和扩展性，使开发者能够更加灵活地构建和集成软件系统 [1]。

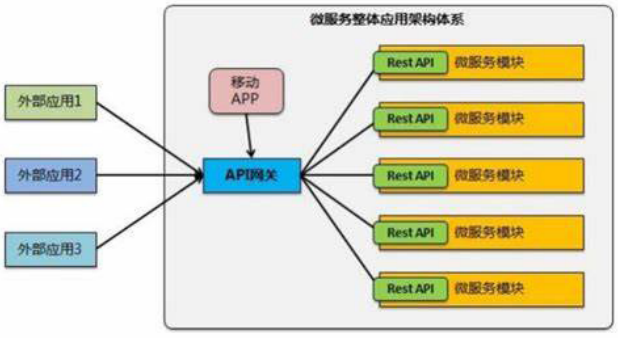

图 1 开源 API

2.2 开源 API 的重要性和应用范围

开源 API 在现代软件开发中具有重要的地位和广泛的应用范围。首先，开源 API 促进了软件开发的协作和共享。通过开放和公开的 API，开发者可以共享自己的功能和服务，并与其他开发者进行合作。这种协作和共享模式推动了创新和技术进步，加速了软件开发的速度和效率。其次，开源 API 提供了丰富的功能和服务，使开发者能够更加轻松地构

【作者简介】鲁博（1987–），男，中国新疆乌鲁木齐人，本科，助理工程师，从事电力系统相关系统开发及运维研究。

建复杂的应用程序。通过使用开源 API，开发者可以直接利用已有的功能和服务，而不必从头开始开发。这大大简化了开发过程，并缩短了软件的上市时间。最后，开源 API 还推动了软件生态系统的发展。开发者可以基于开源 API 构建插件、扩展和应用程序，进一步拓展和丰富现有的软件产品。这种生态系统的发展带来了更多的选择和灵活性，满足了不同用户的需求。

2.3 开源 API 的特点

开源 API 具有以下几个特点：开放性，开源 API 是公开的，任何人都可以访问和使用。这种开放性带来了创新和协作的机会，但也增加了潜在的安全风险。多样性，存在着各种不同的开源 API，涵盖了不同的领域和功能。每个 API 都有其特定的特性和用途，因此需要针对不同的 API 进行安全性考虑和检测。更新性，开源 API 的更新频率较高，不断推出新的版本和功能。这为开发者带来了更多的选择和优化，但也带来了安全性的挑战。每个新版本都可能引入新的安全漏洞或弱点，需要及时进行代码安全性检测和更新。

3 开源 API 代码安全性隐患分析

3.1 常见的开源 API 安全漏洞和弱点

开源 API 在其代码中可能存在多种安全漏洞和弱点，以下是一些常见的示例：不安全的输入验证，开源 API 未能正确验证和过滤用户输入数据，导致恶意用户可以注入恶意代码或执行未授权的操作。不当的权限控制，开源 API 中缺乏适当的权限控制机制，导致未经授权的用户可以访问和操作敏感数据或功能。缓冲区溢出，开源 API 中存在缓冲区溢出漏洞，攻击者可以通过输入超出缓冲区边界的数据来执行恶意代码或破坏系统。跨站脚本攻击（XSS），开源 API 未对用户提供的数据进行适当的转义和过滤，使得恶意用户可以在网页中注入恶意脚本，从而导致用户受到攻击。跨站请求伪造（CSRF），开源 API 未采取有效的防护措施来防止 CSRF 攻击，使得攻击者可以通过伪造用户请求来执行未经授权的操作。不安全的加密和哈希算法，开源 API 中使用的加密和哈希算法可能存在弱点，导致敏感数据容易被破解或篡改 [2]。

3.2 安全隐患对软件系统的影响和风险

开源 API 代码的安全隐患对软件系统和用户都带来重大的影响和风险。以下是一些主要的影响和风险示例：数据泄露，安全漏洞可能导致用户的敏感信息泄露，如个人身份信息、密码和支付信息。这会对用户的隐私和安全造成严重威胁，可能导致身份盗窃和经济损失。未授权访问和篡改，攻击者通过安全漏洞可能获得对系统的未授权访问权限，从而能够修改、删除或篡改数据。这会破坏系统的完整性和可靠性，对业务运营和用户信任造成严重影响。服务拒绝（DoS）攻击，安全漏洞可能导致系统遭受服务拒绝攻击，使系统无法正常提供服务。这会导致业务中断和用户体验下降，对企业形象和声誉带来负面影响。恶意代码注入，通过安全漏洞，攻击者可以在系统中注入恶意代码，如恶意脚本或恶意程序。这可能导致系统崩溃、数据破坏或远程控制，对系统和用户造成严重危害 [3]。

3.3 开源 API 代码安全性的需求和重要性

开源 API 代码的安全性具有重要的需求和意义，主要体现在以下几个方面：用户信任和满意度，用户对软件系统的安全性有很高的期望。通过确保开源 API 代码的安全性，可以增强用户对系统的信任，提高用户的满意度和忠诚度。法规和合规要求，许多行业和领域有关数据安全和隐私保护的法规和合规要求。开源 API 代码的安全性是满足这些要求的基础，确保组织合法经营和数据保护的合规性。业务连续性和可靠性，安全漏洞可能导致系统中断、数据丢失或服务不可用。通过对开源 API 代码进行安全性检测和修复，可以提高系统的稳定性和可靠性，确保业务的持续运营。风险管理和预防，安全漏洞的存在意味着系统面临潜在的威胁和风险。通过及时发现和修复开源 API 代码中的安全漏洞，可以减少潜在攻击的风险，降低系统遭受攻击的可能性。

4 开源 API 代码安全性检测方法

4.1 静态代码分析

4.1.1 原理和技术

静态代码分析是一种在不运行代码的情况下对源代码进行检查的方法。它通过分析代码的结构、逻辑和语法，检测潜在的安全漏洞和弱点，见图 2。静态代码分析可以发现诸如不安全的函数调用、缓冲区溢出、代码注入等常见的安全问题。

4.1.2 静态代码分析的优缺点和适用场景

静态代码分析具有以下优点：①自动化，静态代码分析可以自动进行，无需运行代码。这节省了时间和人力资源，并且可以在早期发现问题，减少后期修复的成本。②全面性，静态代码分析可以对整个代码库进行分析，覆盖广泛。它可以检测到代码中的潜在问题，即使是隐藏在复杂逻辑中的问题也能被发现。

静态代码分析也存在一些限制：①假阳性和假阴性，静态代码分析可能会产生误报（假阳性）或无法检测到真正的问题（假阴性）。因此，人工审核和验证仍然是必要的。②代码覆盖范围，静态代码分析只能分析可用的源代码，对于依赖于动态输入或运行时生成的代码，其分析能力有限。静态代码分析适用于整体代码审查、早期发现和修复潜在问题的场景 [4]。

4.2 动态代码分析

4.2.1 原理和技术

动态代码分析是通过运行代码并监视其行为来检测潜在的安全问题。它可以捕获运行时的异常、漏洞和攻击，并提供详细的运行时信息和报告。在开源 API 代码安全性检

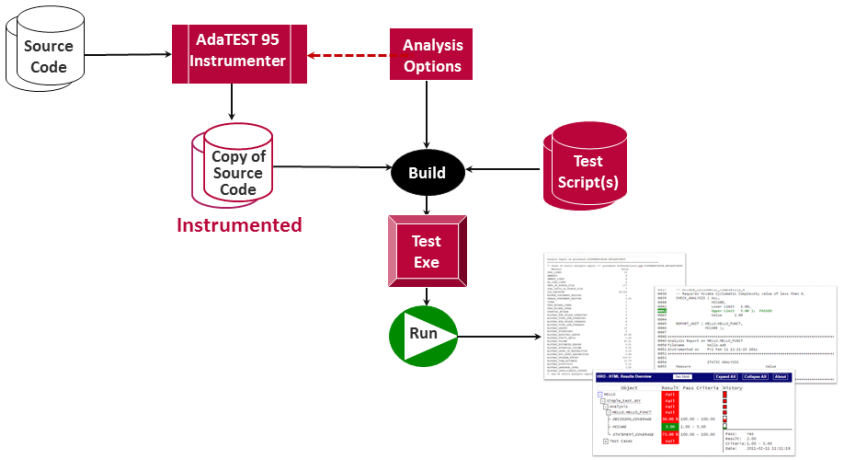

图 2 静态代码分析

测中，常用的动态代码分析工具包括：调试器和代码插桩工具，如 GDB、Valgrind 等，它们可以在代码执行期间捕获和分析运行时信息。安全测试框架，如 OWASP ZAP、Burp Suite 等，它们可以模拟攻击行为，并检测应用程序的安全问题。

4.2.2 动态代码分析的优缺点和适用场景

动态代码分析具有以下优点：①实时检测，动态代码分析可以在代码执行期间实时监视和检测安全问题。它可以发现与特定输入和运行环境相关的问题。②准确性，动态代码分析提供了更准确的运行时信息，可以帮助确定真正的安全问题，减少误报的可能性。

动态代码分析也存在一些限制：①运行时开销，动态代码分析可能会增加系统的运行时开销，并且可能会影响系统的性能和响应时间。②部分覆盖，动态代码分析可能无法覆盖所有代码路径和执行场景，特别是在复杂的应用程序中。动态代码分析适用于模拟攻击行为、检测运行时异常和动态环境的场景。

4.3 漏洞扫描工具

4.3.1 常见的漏洞扫描技术和工具

漏洞扫描工具可以扫描开源 API 代码以查找已知的安全漏洞和弱点。常见的漏洞扫描技术和工具包括：漏洞数据库，如 CVE（通用漏洞和暴露）数据库，它提供了公开的已知漏洞和安全问题的列表。自动化扫描工具，如 Nessus、OpenVAS 等，它们可以自动扫描代码和系统，查找已知的漏洞和弱点 [5]。

4.3.2 漏洞扫描的优缺点和适用场景

漏洞扫描工具具有以下优点：①自动化，漏洞扫描工具可以自动进行扫描和检测，大大减少了人工工作的工作量。②及时更新，漏洞扫描工具通常会更新漏洞数据库，以便及时检测新发现的漏洞和弱点。

漏洞扫描也存在一些限制：①假阳性和假阴性，漏洞扫描工具可能会产生误报或无法检测到真正的问题，因此需要人工审核和验证。②限于已知漏洞，漏洞扫描工具只能检测已知的漏洞和弱点，对于未知的漏洞和定制代码可能无法检测到。漏洞扫描适用于查找已知漏洞和弱点、自动化扫描和持续监测的场景。

5 结语

综上所述，开源 API 代码安全性检测是软件开发中不可或缺的一环。通过深入了解开源 API 代码的安全隐患、合理选择和综合利用多种检测方法，我们可以确保开源 API 代码的安全性，并为用户提供更可靠和安全的软件产品和服务。因此，我们鼓励开发者和组织在开源 API 代码的使用过程中，始终将安全性放在首位，并持续关注和改善开源 API 代码的安全性。

参考文献

[1] 刘国乐,吴越,薛寒星.开源API代码安全性检测研究[J].保密科学技术,2022(4):15-21.

[2] 聂黎明,江贺,高国军,等.代码搜索与API推荐文献分析[J].计算机科学,2017(S1):8.

[3] 蓝海翔.基于Python的RESTful API接口测试框架设计与实现[D].厦门:厦门大学,2016.

[4] 张德浩,徐云.函数级别的复用开源代码检测方法[J].信息技术与网络安全,2021,40(6):7.

[5] 靳鑫,杜林.开源软件安全检测关键技术[C]//数字中国 能源互联——2018电力行业信息化年会论文集,2018.