基于词法分析的代码漏洞模式识别

摘要

软件技术与社会生活的方方面面越来越紧密，软件开发又不可避免的产生各种漏洞，而漏洞识别与定位十分耗费人力物力。如何快速识别并定位漏洞以提升软件运行的稳定与安全成为越来越严峻的问题。随着深度学习的发展，出现了一些可快速自动化的识别软件漏洞的方法，譬如基于代码抽象语法树（以下简称AST）或程序数据流图(以下简称PDG)的漏洞识别方法，从AST或PDG中提取漏洞特征进行模式识别，但有些代码语义譬如通过AST进行抽象后会消失，不利于漏洞的识别。 本文提出一种基于词法分析的代码漏洞特征提取方案，以利用更广的语义依赖进行漏洞识别。

关键字：软件漏洞识别，词法分析，抽象语法树，程序数据流图，深度学习

引言

  软件漏洞与软件开发相伴相生。由于软件规模的不断扩大，导致软件复杂度也不断变高，兼zhi4开发人员对软件开发技术熟练程度、对开发逻辑理解等的差异，不可避免的在软件开发过程中不自觉的引入各种漏洞（业界也称为“Bug”），而且随着业务的迭代，老旧的程序代码可能不适应新业务的需求发展，也导致出现各种漏洞。

  漏洞是潜伏在软件系统中的，发现这些各种各样的漏洞通常是一种被动的行为，一般在软件运行期时系统出现问题了，漏洞才会被开发人员识别并修复，这使得维护软件运行的稳定性相对滞后。而且如果软件系统复杂性很高，开发人员识别漏洞的时间也可能相对变长，这对开发人员的技术水平提出越来越高的要求，从而使得漏洞识别是一件十分耗费人力物力的事。

  已经有一些静态的漏洞分析工具可以帮助开发人员较快的识别漏洞。如 Clang Static Analyzer(1)，这些工具一般是通过预设好