北京邮电大学

网络空间安全学院

《网络安全分析实践》概要设计报告

项目：基于源代码检测和动态执行的二阶 SQL 注入漏洞检测

组员：王硕、彭致远、李懿飞、王晨旭

2020 年 9 月 30 日

目录

1 引言

1.1 背景

关系型数据库被广泛用于 Web 应用之中，然而它所带来的安全问题一直是威胁 Web 安全的主要因素之一。二阶结构化查询语言注入是一种新型 Web 漏洞，同一阶 SQL 注入技术一样，能够威胁客户端、服务器上的数据和系统的安全。

传统的一阶 SQL 注入检测方法不能有效对其进行检测。因此，二阶 SQL 注入漏洞具有极强的隐蔽性，广泛存在于 Web 应用中。近些年，二阶 SQL 注入逐渐替代传统 SQL 注入技术成为黑客行为的突破口。因此，对于二阶 SQL 注入漏洞的检测成为了研究的热点。

目前学术界对二阶 SQL 注入漏洞的检测方法上并不多，研究的深度和可行性也不高。

一种常见的技术为模糊测试，如 AWVS、X-Scan 等，对通过爬虫找到的可控参数发送大量测试用例，并分析应用的异常检测漏洞。动态分析虽然实施部署简单，误报率低，但也存在测试效率低、覆盖度不高等问题。并且这种针对单一注入点进行检测的方式无法有效处理 Web 应用多阶段之间的联系，不能检测出二阶 SQL 注入漏洞。

除了动态的注入测试，也有静态分析技术。早期的静态分析技术，如 ITS4 只是简单地在源代码中寻找危险函数的调用，误报率非常高，需要大量的人工分析其检测结果。以 Graudit 为代表的通过正则表达式匹配寻找漏洞的技术，虽然一定程度上增加了检测的灵活性，但是依然需要大量人工参与。基于数据流的污点分析技术，如 Pixy 是静态分析检测 Web 漏洞技术成熟的标志。Dashies 等实现了 Web 漏洞检测工具 RIPS，但是在检测二阶 SQL 注入漏洞时，仍然存在 2 个问题：无法准确定位污染数据的中间存储位置和无法判断污染数据到达危险函数前是否经过有效过滤。静态分析有覆盖面广、效率高的优点，但是误报率和漏报率高，尤其是不能准确检测多阶漏洞。

本项目提出了一种基于源代码检测和动态执行的二阶 SQL 注入漏洞检测技术，并给予了代码实现。想法来源在大二上学期《网络安全平台设计实践》课程中学习了如何使用动态注入的技术判断是否存在 SQL 注入漏洞，又在大二下学期学习了《编译原理与技术》课程，掌握了基本代码语言的语法分析与语义分析技术。所以有了检测二阶 SQL 注入漏洞的新方法——基于源代码检测和动态执行的二阶 SQL 注入漏洞检测方法。

1.2 编写目的

本概要设计说明书是基于需求说明书编写。目的在于在需求分析的基础上，结合文献调阅资料与实际情况，规定软件、硬件、技术选择，设计系统总体系统架构、总体功能模块以及各个子模块。为之后系统的详细设计以及程序编写、测试提供指导说明。

1.3 项目及范围

- 项目名称：基于源代码检测和动态执行的二阶 SQL 注入漏洞检测工具
- 项目成员：北京邮电大学网络空间安全学院“网络安全分析实践”课程开发小组
  - 王硕（组长）：2018213641
  - 彭致远：2018213646

- 李懿飞：2018213632
- 王晨旭：2018213636

- 系统范围：具有 PHP7.4 环境的 Windows 系统计算机
- 用户：无限制
- 实现项目的计算机网络：校园网

本设计文档适用于基于源代码检测和动态执行的二阶 SQL 注入漏洞检测工具项目进行系统分析、详细设计以及编码实现

1.4 读者对象和阅读建议

本说明书的预期读者为系统设计人员、软件开发人员、软件测试人员和项目评审人员。其中系统设计人员、软件开发人员、软件测试人员为小组内部成员，软件评审人员为课程老师或助教。

1.5 定义、术语和缩写

序号	术语或缩写	解释
1	SQL	结构化查询语言(Structured Query Language)简称 SQL，是一种特殊目的的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统。
2	AST	抽象语法树（Abstract Syntax Tree，AST），是源代码语法结构的一种抽象表示。它以树状的形式表现编程语言的语法结构，树上的每个节点都表示源代码中的一种结构。
3	CFG	控制流程图，是一个过程或程序的抽象表现，是用在编译器中的一个抽象数据结构，由编译器在内部维护，代表了一个程序执行过程中会遍历到的所有路径。它用图的形式表示一个过程内所有基本块执行的可能流向，也能反映一个过程的实时执行过程。本项目用于描述变量在流动过程的执行过程
4	PHP-Parser	开源工具，用于生成 PHP 文件的语法分析树
5	SQL-Parser	开源工具，用于生成 SQL 语句的语法分析树

2 系统设计概述

2.1 限制和约束

受小组成员知识水平和调研结果，现列出在项目开发过程中需要准守的一些标准和规则如下：

1） 开发期限：2020 年 11 月 6 日前完成
2） 硬件限制：小组内成员设对本项目的功能和要求没有问题
3） 编程语言：PHP
4） 待扫描的网站源码：以 php 语言为主
5） 界面语言：尽量使用英文、避免出现编码问题
6） PHP 文件解析工具：PHP-Parser 工具
7） SQL 语句解析工具：SQL-Parser 工具
8） 开发工具：PhpStorm 2020.2.3

2.2 设计原则和设计要求

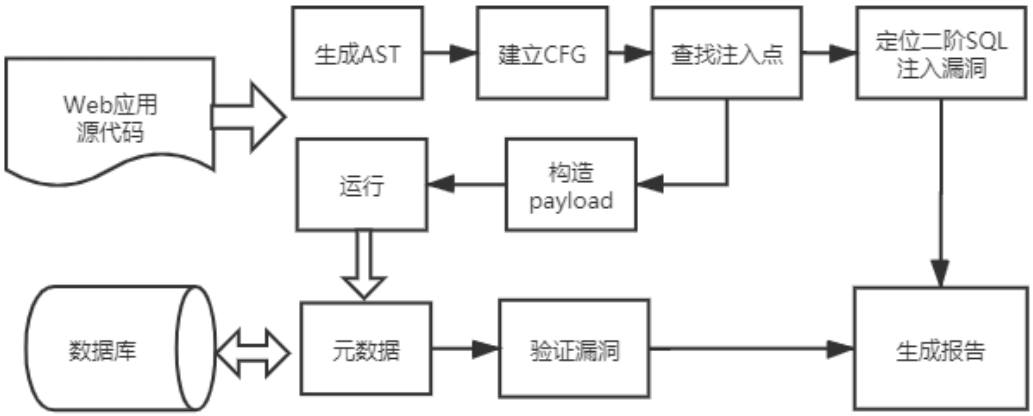
为提高小组合作效率，保证分工独立、软件的易用性、稳定性较高，本项目考虑如下设计原则和要求：

1） 命名规则：下划线分割小写字母的方式命名，如 static_scan()，或者以“骆驼命名法”，为以大写字母开头的英文单词的组合，如 variable_list
2） 模块独立性原则：每个模块有单独的输出，也可以被其他模块调用执行。
3） 系统容错率高：对用户输入、外界因素等做检查，防止程序出错。
4） 软件高效：优化程序的算法，降低时间复杂度，提高程序执行效率。
5） 用户友好性：提高交互性。

2.3 系统模块划分

2.3.1 系统模型

本工具的系统模型如下所示：



实现的过程可以总结如下：

- 以 PHP 项目源码作为输入，对每个 PHP 文件生成语法分析树，进而对每个 PHP 文件内的变量生成变量的控制流图，通过“超全局变量”“引用”“函数调用”等建立各个 PHP 文件之间的关系，从而生成所有变量的控制流图。
- 分析变量在流动过程中的来源、去向、变化。发现敏感数据即发现二阶 SQL 注入的注入点和触发点，保存为静态检测的结果
- 通过动态的执行，注入点输入 payload，观察数据库内数据的变化，如果符合二阶 SQL 注入漏洞的特点，即生成报告

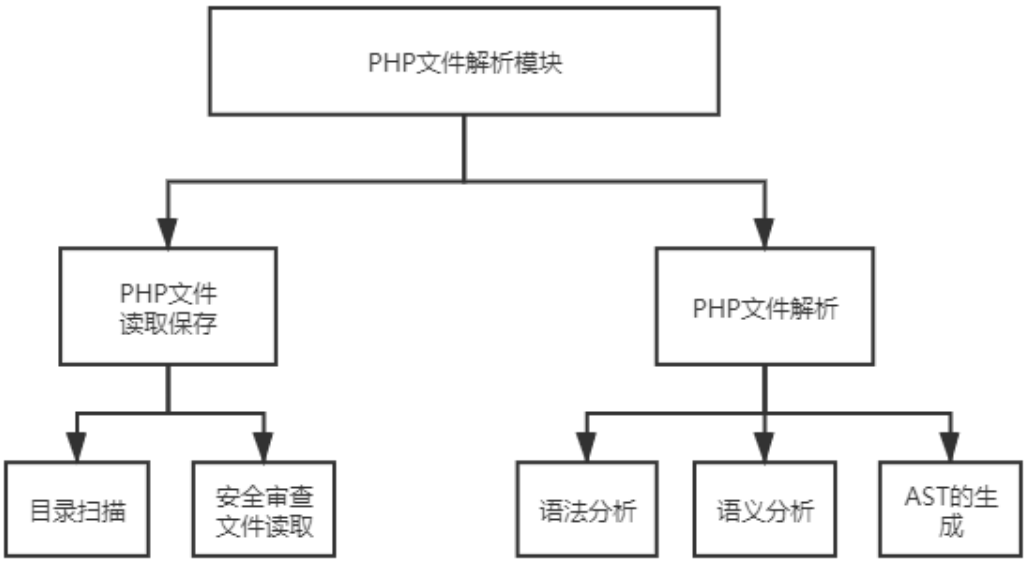
2.3.2 系统模块划分

根据系统功能图，可以将本项目划分为六个模块，分别为：

1） PHP 文件解析模块
2） 变量控制流图生成模块
3） SQL 语言解析模块
4） 回溯扫描查找模块
5） 动态注入测试模块

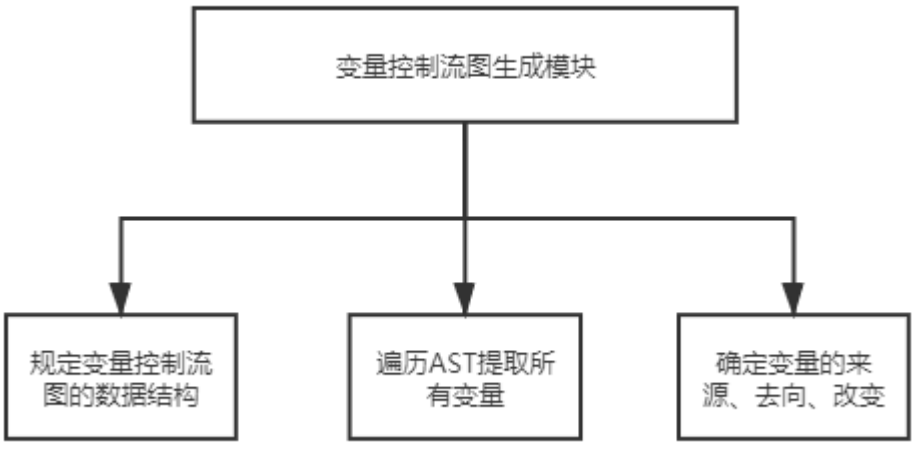
3 功能模块设计

3.1 PHP 文件解析模块



本模块的主要功能为：扫描用户 PHP 网站项目目录，从中提取 PHP 文件，对每个 PHP 文件，进行词法分析、语法分析、语义分析等，最终生成每个 PHP 文件对应的语法分析树，并提供一套递归的遍历方法，从而便于下一个模块从中递归的提取变量，并判断变量的深度（全局、局部、循环内等）

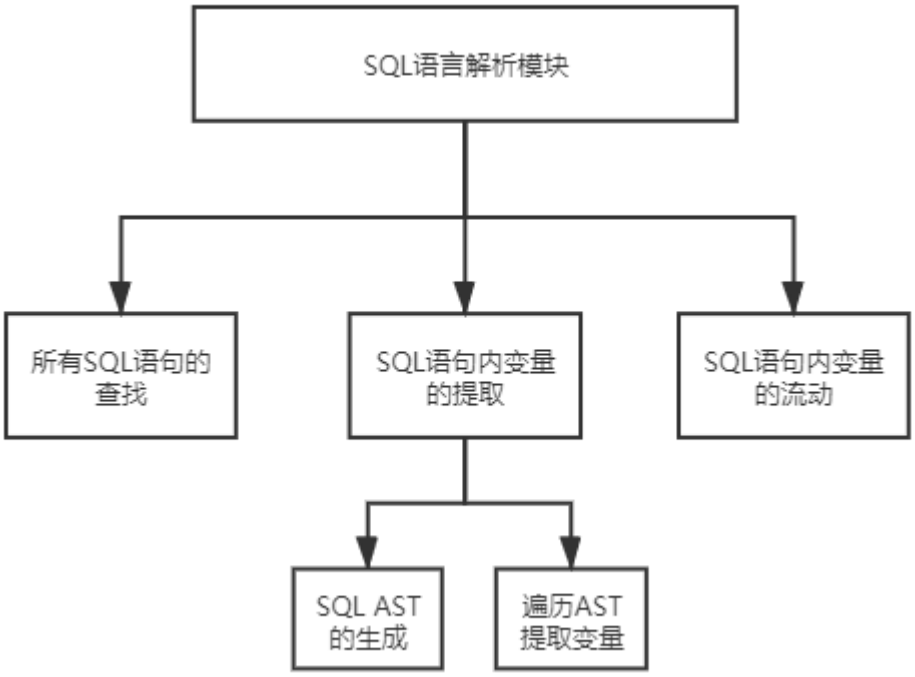
3.2 变量控制流图生成模块



本模块的功能主要是通过 PHP 文件生成的语法分析树以及上一个模块提供的遍历 AST

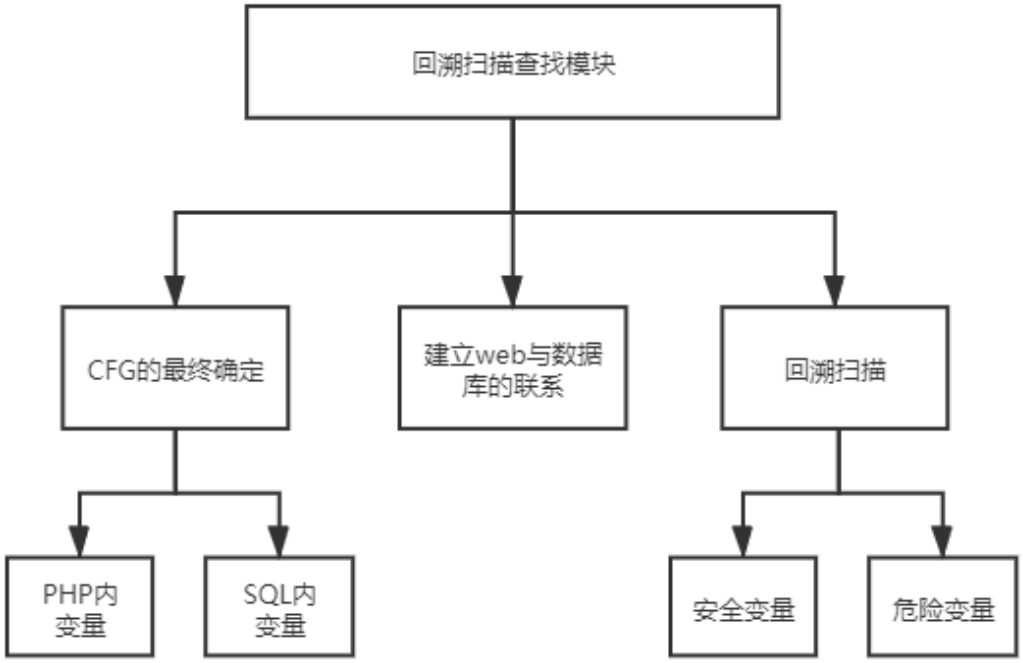
的方法，提取 PHP 文件内部的变量，并通过变量之间的传递，建立变量的控制流图。通过“超全局变量”“引用”“函数调用”等建立各个 PHP 文件之间的关系，从而生成所有变量的控制流图。

3.3 SQL 语言解析模块



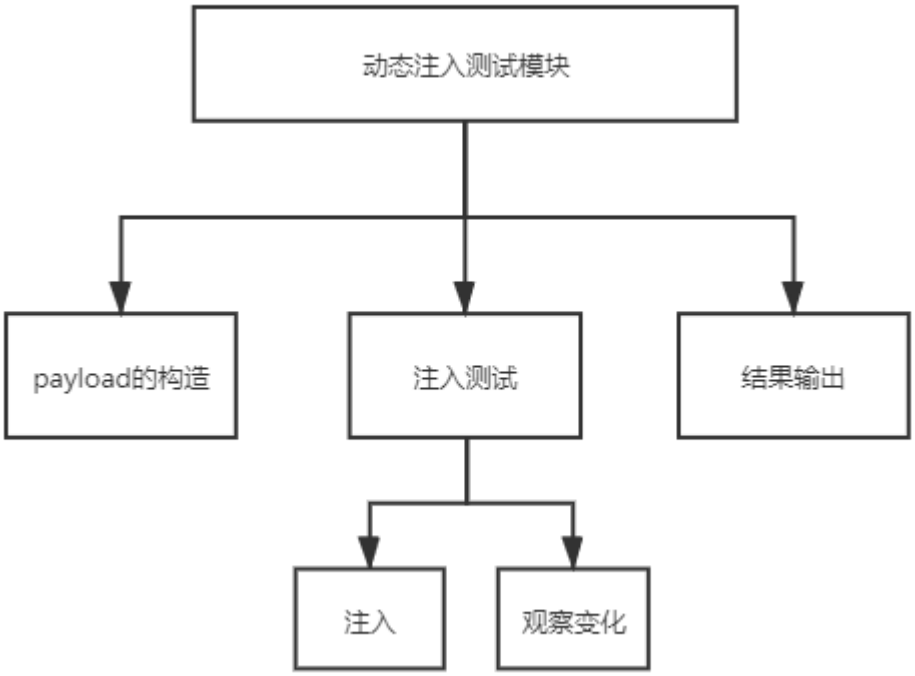
本模块主要是用来获取和解析 php 源代码中的 SQL 语句，首先通过 php-parser 解析树的解析结果获取 sql 语句的信息，然后再利用 sql-parser 解析获取的 sql 语句，将 sql 语句中的关键信息（如变量名、表项名、数据库名、变量流动）提取和整理，保存在数组中，供其他 php 分析模块使用。

3.4 回溯扫描查找模块



本模块的主要功能为：建立 PHP 和 SQL 的变量关系，生成最终的变量控制流图，并回溯扫描将变量分类，分成安全变量和危险变量，并记录二阶 SQL 注入漏洞的注入点和触发点，作为静态扫描的结果。

3.5 动态注入测试模块



本模块的主要功能为：由于静态的扫描误报率很高，所以动态的去验证是必要的。通过构造 payload，进行数据库的注入，并观察网站和数据库内的变化，如果构造的 payload 与正常的数据产生的变化不同，那么就存在二阶 SQL 注入漏洞。并将结果保存，输入。

4 界面设计

界面最好是以图形化界面的形式展示，但是考虑到时间问题，可以采用命令行的方式进行交互，但是要注意以下几点：

1） 命令行界面应当友好，即提示语句排列清楚，输入人性化
2） 对于软件运行过程中会有实时输出，避免卡顿/运行不容易区分
3） 对于用户错误输入，应当能够检测，并有所提示，然后为用户提供再次输入的机会

5 运行环境设计

5.1 开发环境

程序开发阶段需要进行 PHP 文件的解析、网站服务器的搭建运行等，考虑到时间和小组成员条件等因素，故对开发环境有一定的要求如下：

1） 操作系统：Windows 10 / MAC 系统
2） PHP 环境：php7.4
3） 开发工具：phpstorm、x-debug、notepad++等
4） 服务器：Apache Tomcat 2019
5） 浏览器：Chrome、Firefox 等
6） 数据库：MySQL 8.0
7） 联网要求：无要求（能连接本地服务器、数据库即可）

5.2 运行环境

程序运行阶段需要网站实际运行起来，故要求和开发环境类似，故对程序运行环境（测试环境）有下列要求：

1） 操作系统：Windows 7 及以上 / MAC 系统
2） PHP 环境：php7.4
3） 运行工具：phpstorm
4） 服务器：Apache Tomcat 2019
5） 浏览器：Chrome、Firefox 等
6） 数据库：MySQL 8.0
7） 联网要求：无要求（能连接本地服务器、数据库即可）

6 安全设计

6.1 系统备份设计

为防止因硬件、软件原因导致的程序异常中断而造成数据丢失，因对数据具有备份以及及时保存措施，包括

1） 数据库原有数据的备份
2） 网站源代码的备份
3） 程序不能处理的内容的记录与保存
4） 源代码扫描结果的保存和动态执行结果的保存与备份

6.2 系统容错设计

为提高系统容错能力，防止由于网络数据包复杂、用户输入不合法、检测文件异常等问题，开发程序时应注意以下内容

1） 用户的错误命令输入，应当有所检查，并给予用户提示
2） 当用户没有将网站原项目文件放置到正常目录时，应该有所检查，并给予用户提示
3） 当用户网站没有运行起来时，即服务器连接失败时进行检查与提示，并终止程序
4） 当用户误删项目中的一些输出、输入文件夹、检测备份文件时，应当有所检查、并重新生成这些文件。