北京邮电大学

网络空间安全学院



**测试及结果展示报告**

**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12月10日**

目录

[1 引言 3](#_Toc58503531)

[1.1 目的 3](#_Toc58503532)

[1.2 背景及范围 3](#_Toc58503533)

[1.3 定义、术语和缩写 3](#_Toc58503534)

[2 测试概述 4](#_Toc58503535)

[2.1 测试环境与配置 4](#_Toc58503536)

[2.2 测试内容 4](#_Toc58503537)

[2.2.1 PHP和SQL解析树 4](#_Toc58503538)

[2.2.2 CFG的正确生成 4](#_Toc58503539)

[2.2.3 扫描结果验证 5](#_Toc58503540)

[3 运行结果与分析 5](#_Toc58503541)

[3.1 PHP和SQL解析树 5](#_Toc58503542)

[3.2 CFG的正确生成 7](#_Toc58503543)

[3.3 扫描结果验证 10](#_Toc58503544)

[4 结论 15](#_Toc58503545)

# 1 引言

### 1.1 目的

本测试报告为大三上学期网络分析实践课程的“基于源代码检测和动态执行的二阶SQL注入漏洞检测工具”的测试报告，目的在于测试工具的基本功能并分析测试结果，判断系统是否符合需求。

### 1.2 背景及范围

* + 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

本测试报告预期参考人员包括测试工具的同学、开发工具的同学、验收工具的老师。

### 1.3 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 测试概述

### 2.1 测试环境与配置

* + 操作系统：Windows 10
  + PHP环境：PHP7.4，有x-debug调试测试工具
  + 运行软件：PhpStorm 2020.2.3
  + 虚拟机：kali linux 2020
  + 待扫描的项目：一个自己搭建的demo网站、三个从github网站下载的PHP Web项目（两个具有二阶SQL注入漏洞、一个没有）

### 2.2 测试内容

#### 2.2.1 PHP和SQL解析树

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 要求 |
| 1 | PHP AST的生成 | 对特定PHP文件，对比是否能完全分析PHP文件并生成对应语法分析树 |
| 2 | SQL AST的生成 | 是否能扫描出所有SQL语句，并均能生成对应的语法分析树 |

#### 2.2.2 CFG的正确生成

|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 要求 |
| 1 | PHP内变量的流动 | 查找出所有的变量、并准确的分析变量的来源、去向和变化 |
| 2 | SQL内变量的流动 | 定位数据库的表格和表项；并查找出所有的变量、并准确的分析变量的来源、去向和变化 |

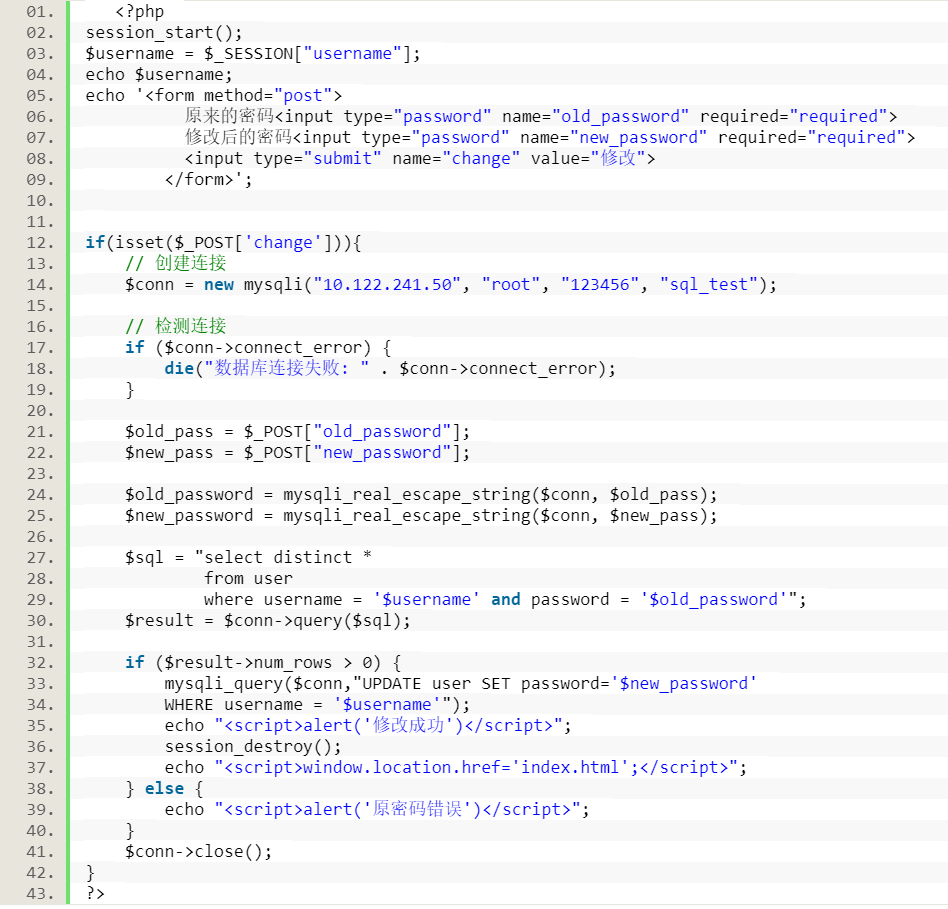
#### 2.2.3 扫描结果验证

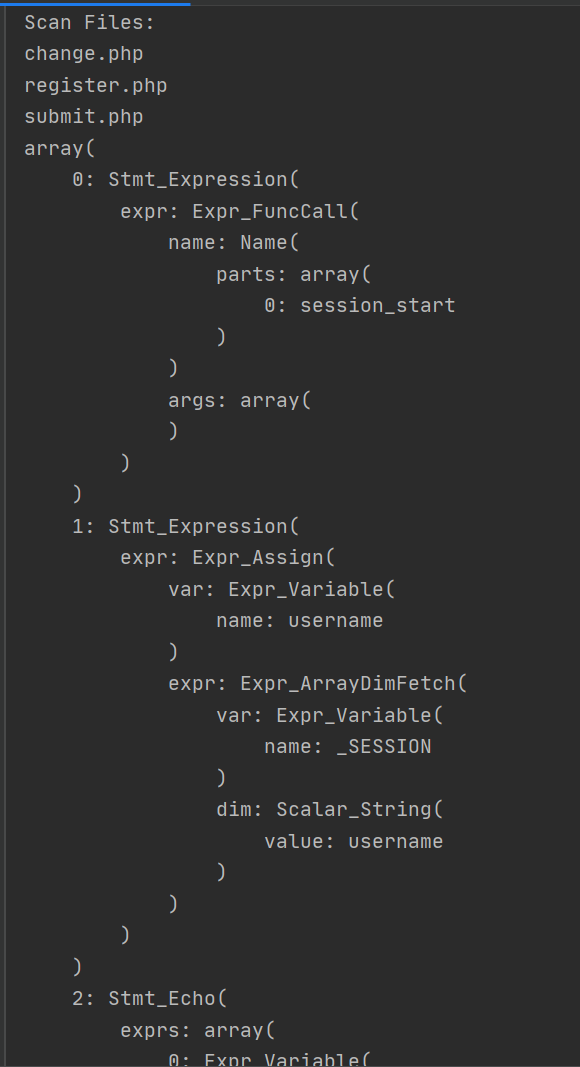
|  |  |  |
| --- | --- | --- |
| 序号 | 功能 | 要求 |
| 1 | 检测出网站内潜在的二阶SQL注入漏洞 | 对存在二阶SQL注入漏洞的网站能准确定位到二阶SQL注入漏洞更多注入点与触发点，对于不存在二阶SQL注入漏洞的网站，报告安全。 |
| 2 | 安全检查 | 用户输入命令不合法，会有回显，不报错；  用户文件项目不存在，提示用户 |

# 3 运行结果与分析

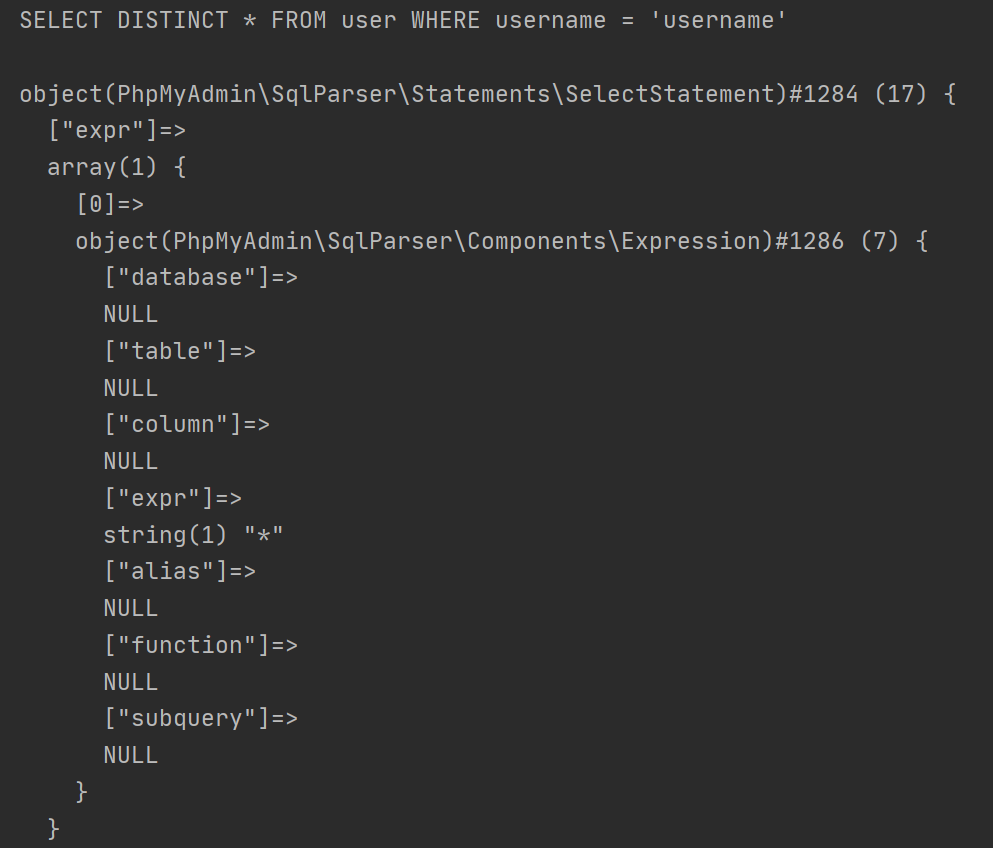
### 3.1 PHP和SQL解析树

（1）PHP文件的AST，测试用例为demo项目内的change.php文件，源码和生成的AST如下：（由于语法分析树很大，故截图只截了一部分，详细的输出保存在文件夹：/输入项目/demo/parser文件夹内对应的语法分析树）

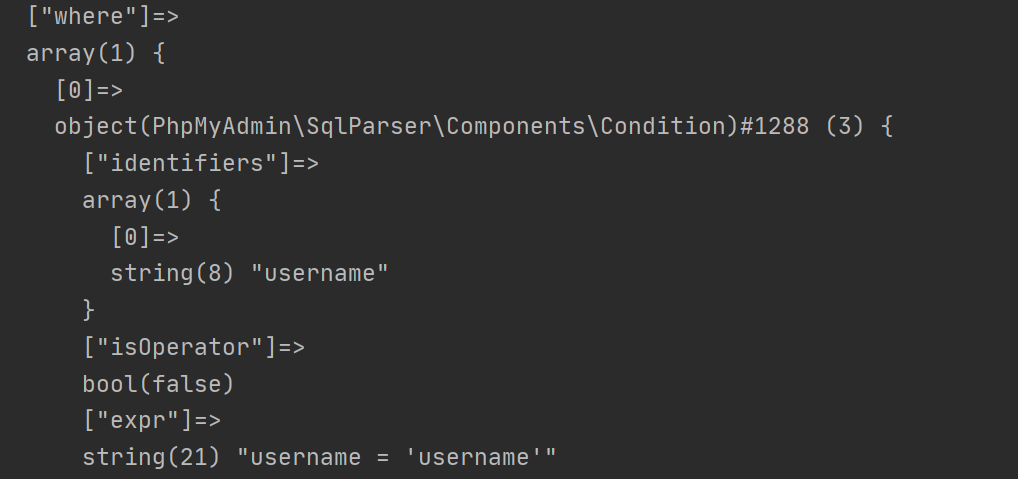




（2）SQL语句的AST，输出如下：

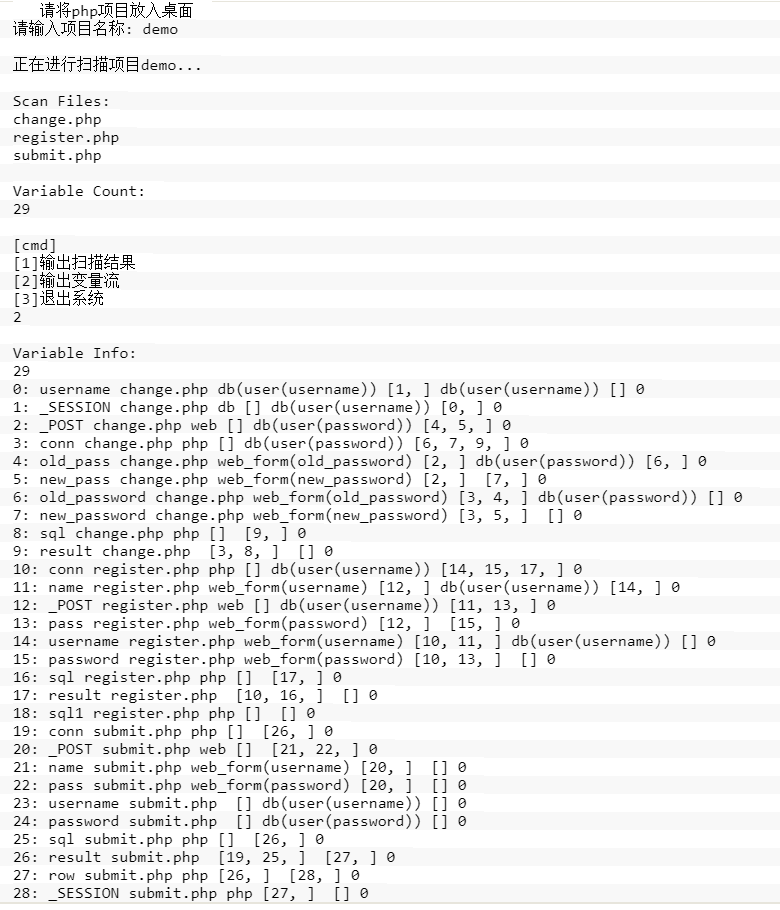






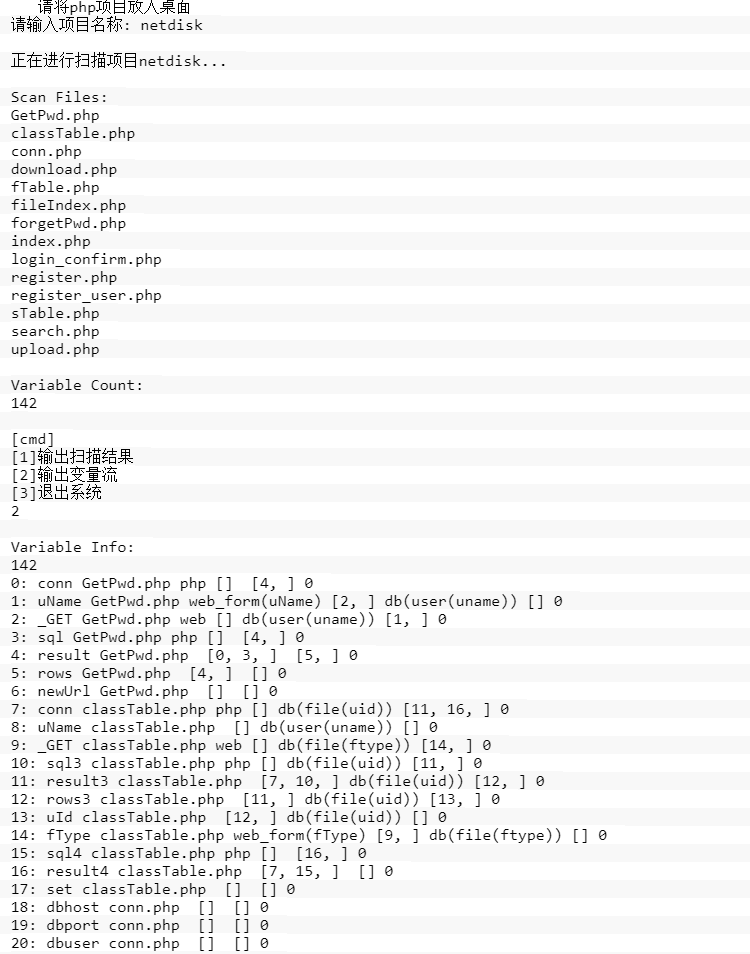
### 3.2 CFG的正确生成

1、测试demo项目，生成的CFG的每个结点的信息打印如下：

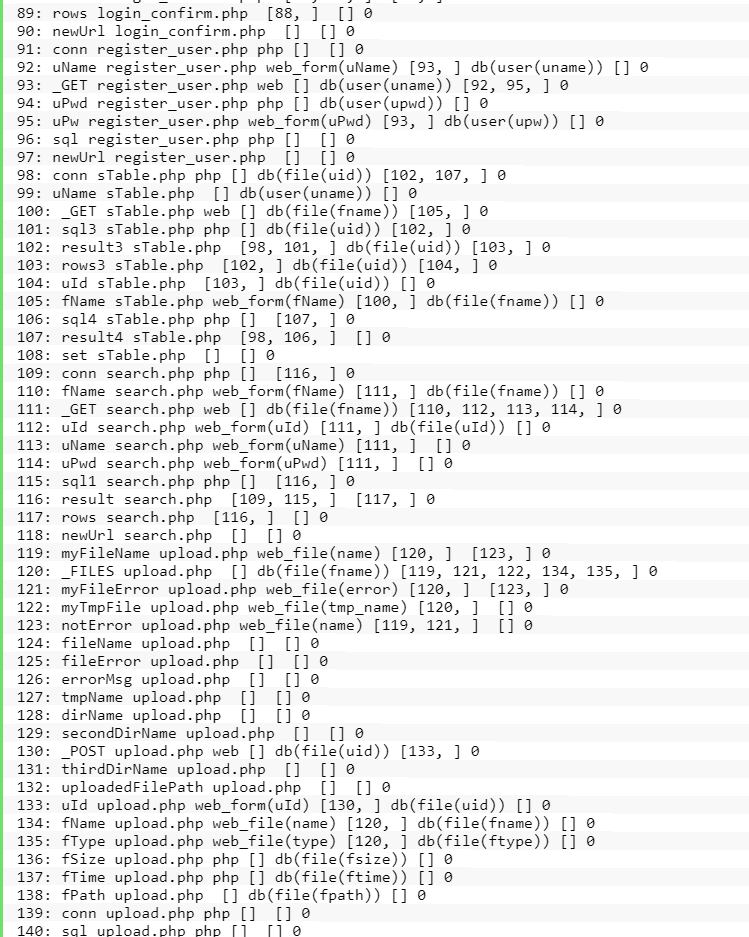


通过简单的分析，可以看出此CFG是合法的。

2、测试github项目netdisk，一个简单的网盘系统，生成的CFG的每个结点的信息打印如下：



……



3、测试github项目shopping，输出如下：

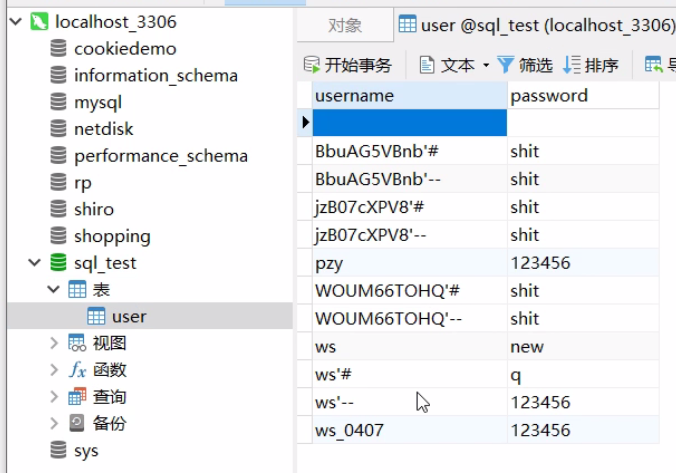
只截取了部分结点信息：



### 3.3 扫描结果验证

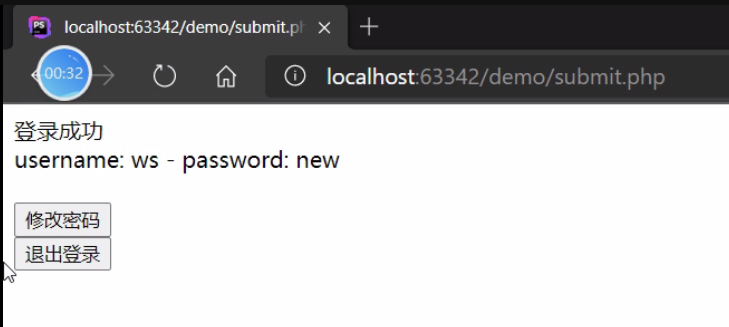
1、demo项目

（1）数据库内容：

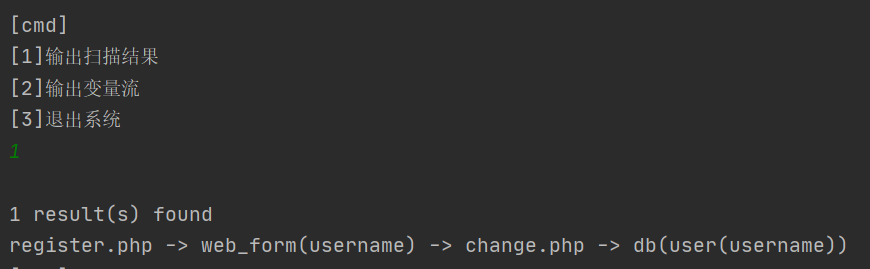


（2）网站运行效果：





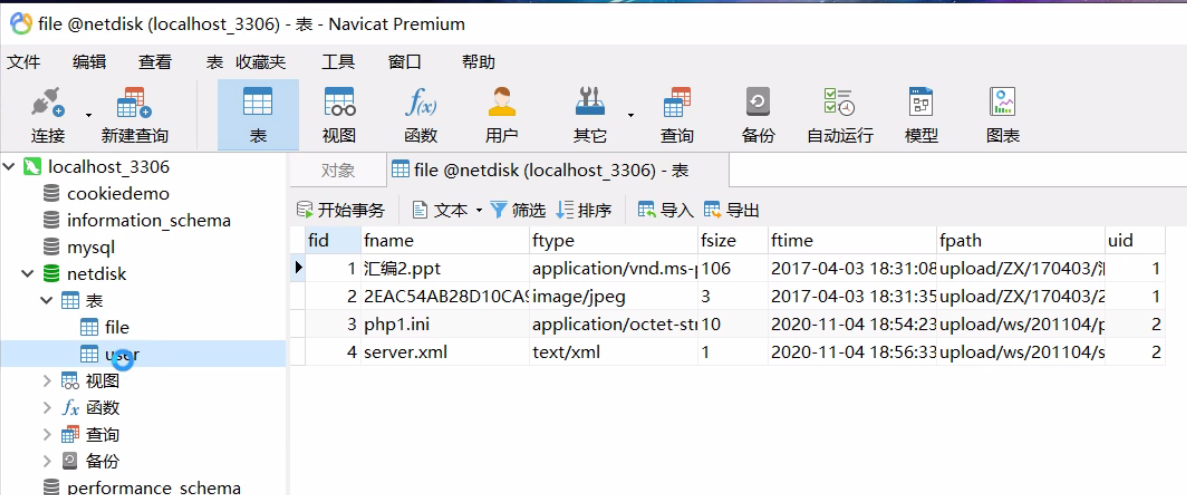
（3）含有一个登陆-注册-修改密码的二阶SQL注入漏洞，检测结果如下：



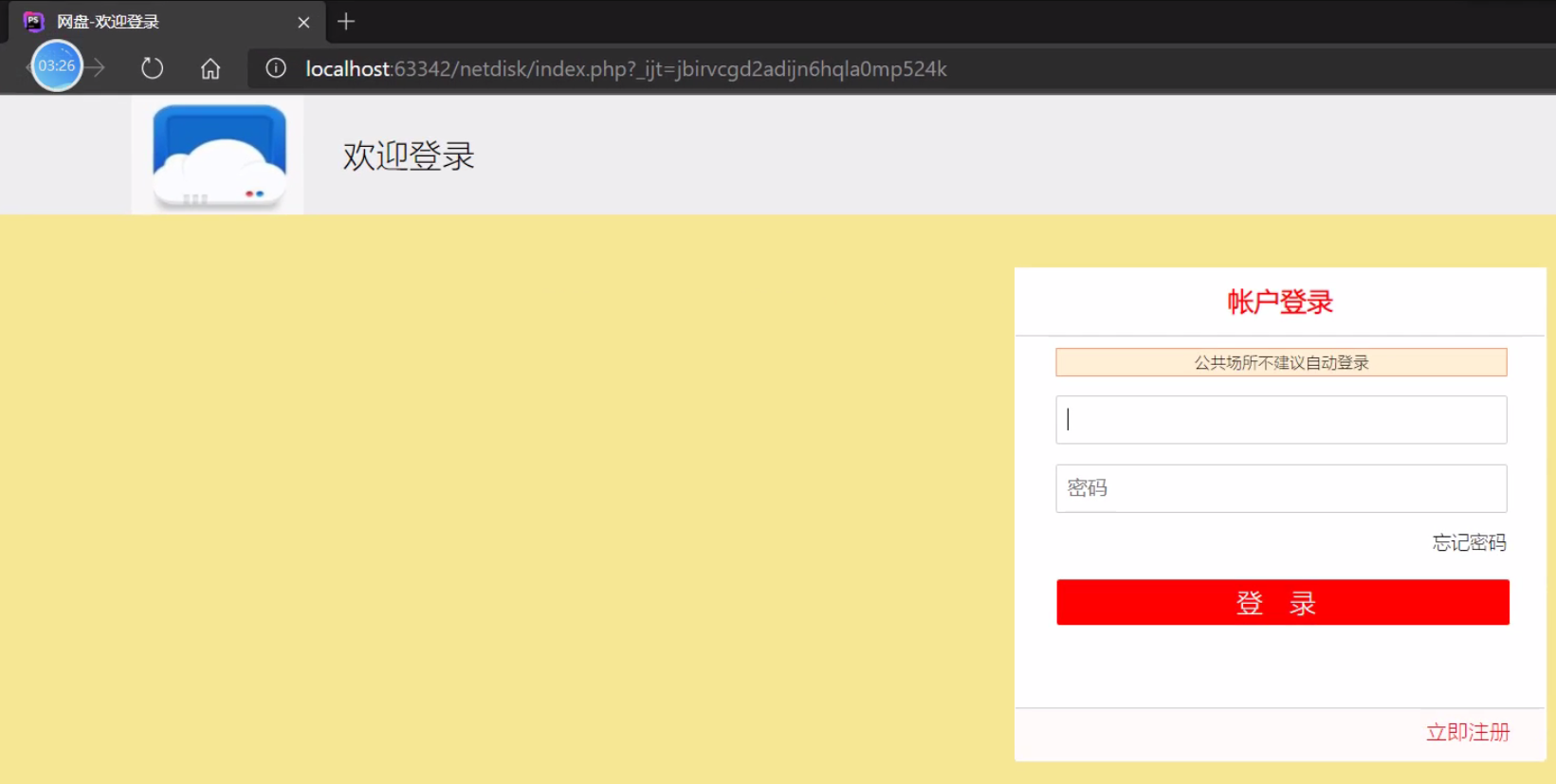
可以得知注入点在register.php界面，触发点在change.php界面，来自用户的输入，最终存到数据库内，又被拿出来执行。故存在二阶SQL注入漏洞。

2、netdisk项目

（1）数据库信息：

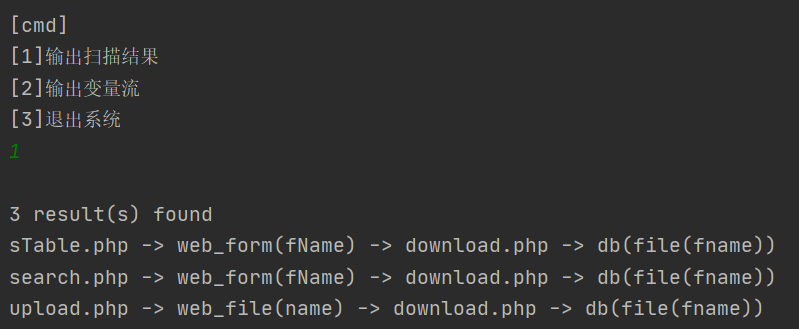


（2）网站运行效果：



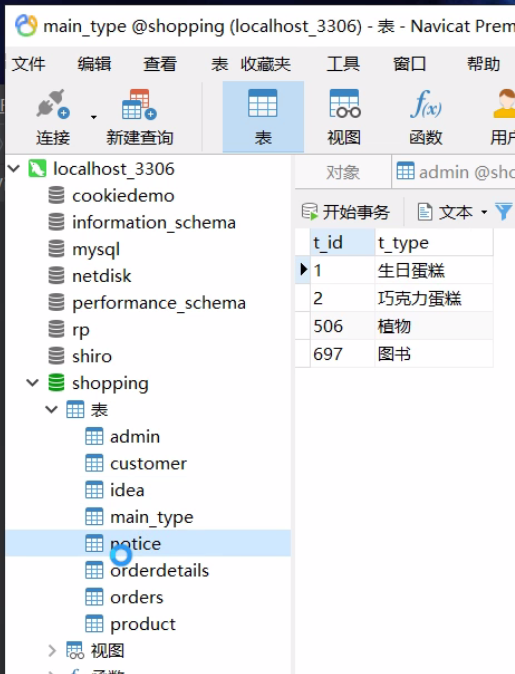


（3）网站含有一个二阶的文件SQL注入漏洞，文件名作为用户的上传的文件，直接被后端读取存到数据库内，又重新在下载界面获取到下载的文件的文件名，到后端查找文件的存储地址，故存在二阶SQL注入漏洞。

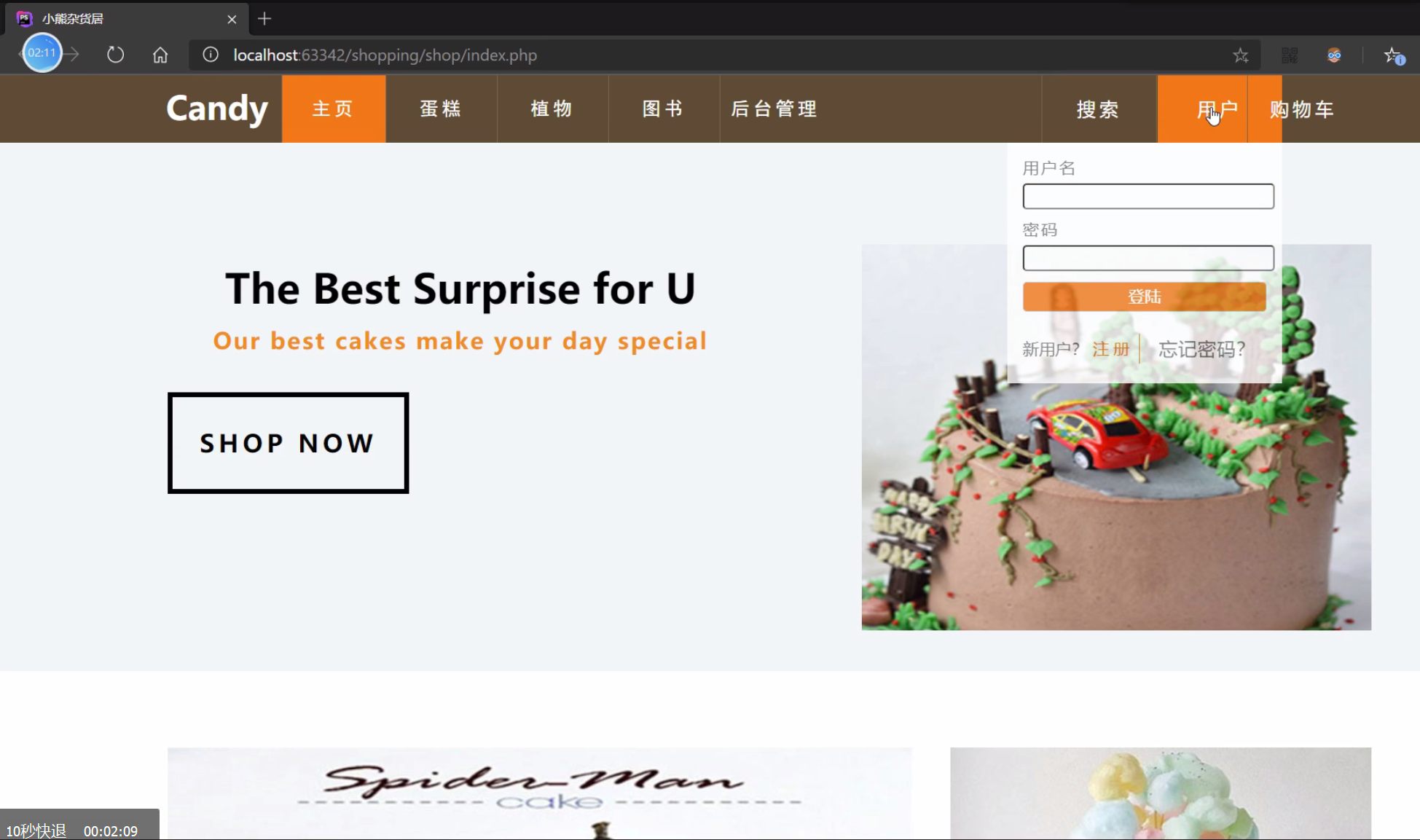


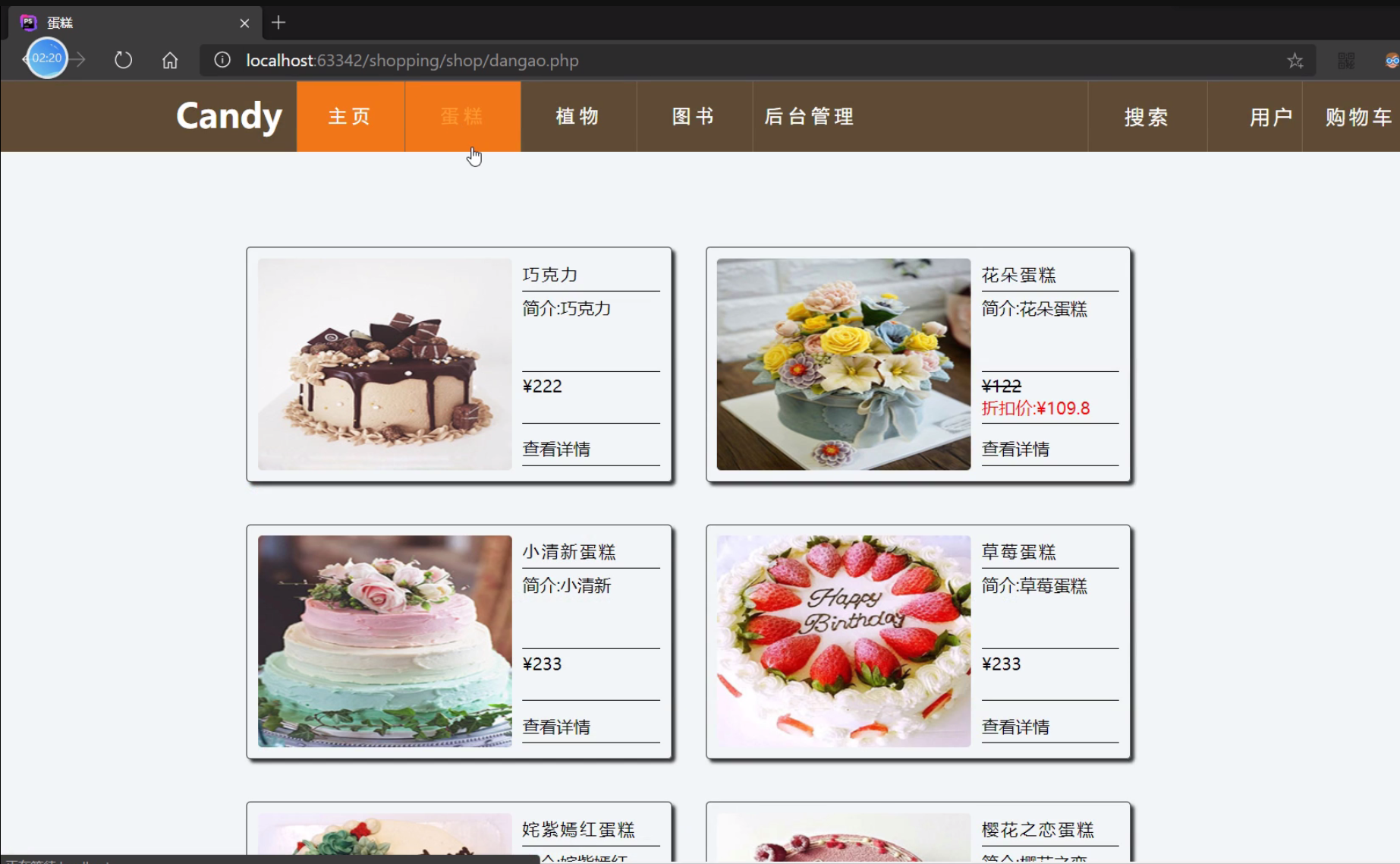
3、shopping项目

（1）数据库信息：

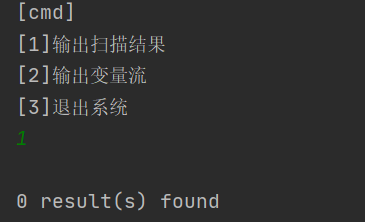


（2）网站运行效果：



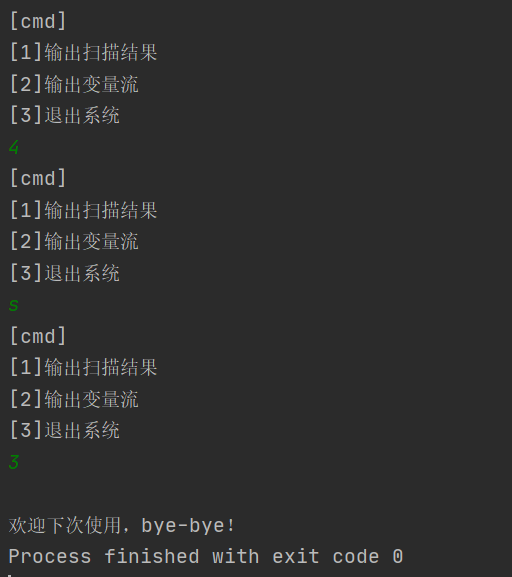


（3）由于网站只是一个简单的加购物车，没有涉及到数据库内的操作，所以0 result，结果正确：

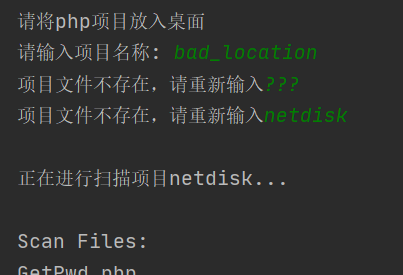


4、最后测试项目的安全性

（1）错误的命令输入：



（2）没有项目文件：



# 4 结论

1. 项目功能达到要求，对于提供的项目文件能够准确的扫描出存在的二阶SQL注入漏洞。
2. 能够定位到二阶SQL注入漏洞的注入点和触发点，提醒项目管理者注入防范。
3. 项目运行良好，不会出现报错，用户友好性程度高。