1. **需求分析**

**1.1 引言**

在sql注入的过程中，其大部分都是按照流程一步一步实现注入并且拿到webshell的，本系统的目的就是降低sql注入的难度，实现自动化失去了注入流程。

**1.2 任务概述**

**1.2.1 需求分析的任务**

本系统的目标是：通过用户的输入判断web系统是否存在sql注入漏洞，并获取数据库信息。

**1.2.2 总体需求**

本系统主要通过获取系统的url连接，对链接进行解析处理，然后利用sql注入漏洞检测技术对获取到的url进行漏洞检测，已完成判断web系统中是否存在sql注入漏洞，并通过漏洞进行对数据库的爆破，从而拿到数据库信息。

**1.3 功能需求**

**1.3.1功能划分**

该系统大致需要完成三部分工作，分别是用户输入处理功能、Sql注入漏洞检测功能、数据库爆破功能，信息显示功能。

**1.3.2功能描述**

（1）用户输入处理功能，用户输入需要检测的url，系统进行解析处理，并进行目标存活检测，方便后面功能的实现。

（2）Sql注入漏洞检测功能，通过构造sql语句进行web站点的漏

洞检测，并获取到web站点的基本信息。

（3）数据库爆破功能，利用收集到的信息构造sql注入语句，对web站点的数据库进行爆破，获取数据库的名称，版本，里面创建的表等信息，并进行存储。

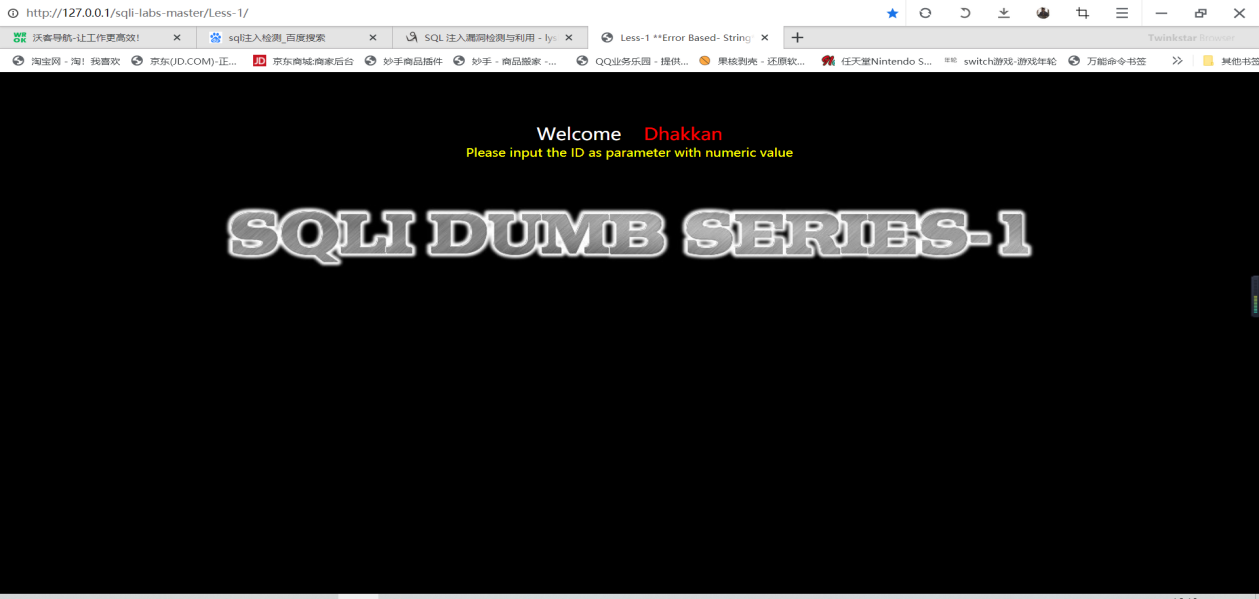
（4）信息显示功能，对于收集到的信息，采用友好界面的形式进行显示。

**二、系统框架**

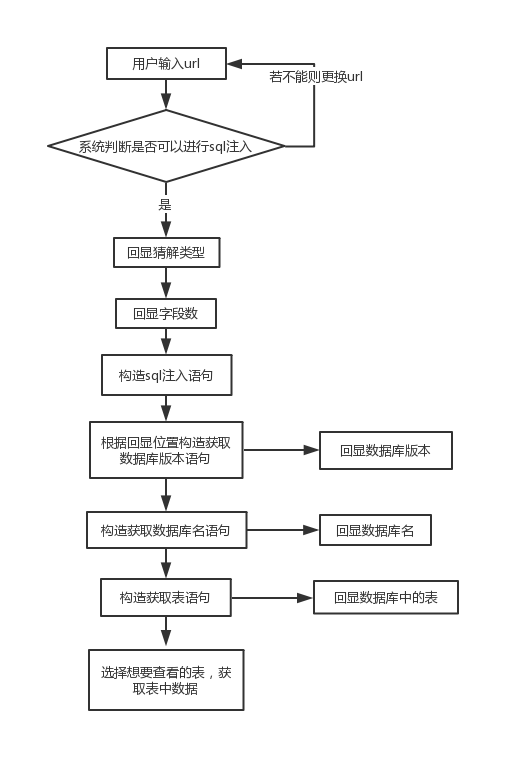
实验环境是使用PHP study搭建的php调试环境，PHP study软件包中集成了最新的Apache+php+MySQL，在本次实验中需要用到Apache和MySQL才能使用sqli-labs靶机。我们需要设计程序脚本来对靶机进行sql注入。



实验环境需要配置Apache2.4.39、MySQL5.7.26，直接在PHP study中启动就好。然后安装好sqli-labs，在本地浏览器就可以对靶机进行渗透了。



**三、系统流程图**

****

1. **总体设计**

根据模块之间的关联性，将整个程序分为了4个模块。

1.检测url合法性模块。

class page\_validity:

def judge(url):

判断用户输入的url是否可以访问。

2.注入判断模块。

此模块的功能是判断该url是否存在注入漏洞，如果存在那么得到它的注入类型，字段数等信息。

class information:

def built\_url(url):

#判断是否存在注入点，以及注入类型

def get\_in\_str(url,url1):

#构造注入语句

def get\_counts(url,url2):

#得到字段数

3.构造注入语句模块，用来获得数据库内信息。

此模块的功能是构造注入语句，来获取回显位置，数据库名，版本，表名等信息。

class exploit():

#构造select语句，用来拼接union select

def get\_string\_counts(counts):

#构造union语句

def exploit\_string(url,string\_counts):

#构造可以爆出版本的语句

def exploit\_version(url,position,string\_counts):

#构造可以爆出数据库的语句

def exploit\_database(url,position,string\_counts):

#构造可以爆出表的语句

def exploit\_tables(url,position,get\_database,string\_counts):

#构造可以爆出列的语句

def exploit\_columns(url,position,get\_database,get\_tables,string\_counts):

#爆数据

#参数，url，位置，数据库名，表名，列名

def exploit\_data(url,position,db\_and\_table,columns,string\_counts):

4.获取信息模块。

此模块的功能是根据构造的注入语句来获取相应的信息。

class information:

#获得回显位置

def get\_position(url,exploit\_string):

#获取信息

def get\_infor(url,url1):

#获得数据

def get\_data(url,url1):