

# I . 项目总体概述

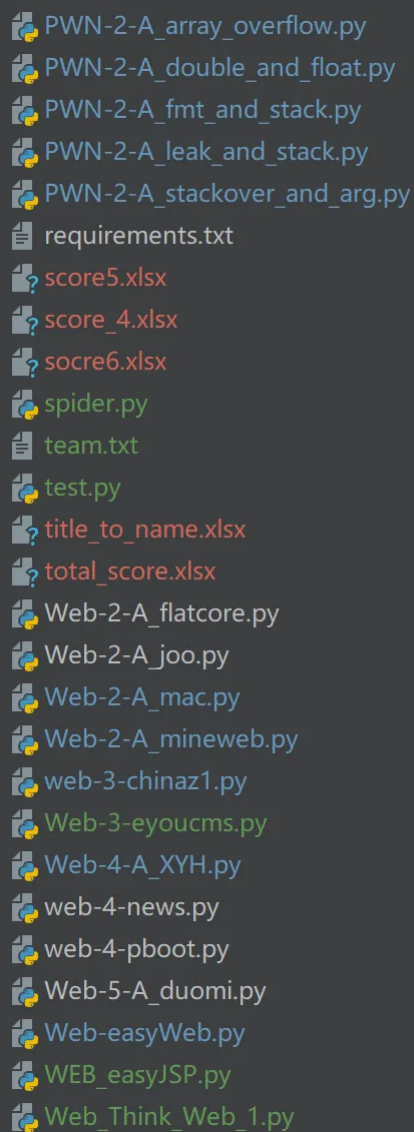
---

项目描述:

页面介绍:

## 项目描述:

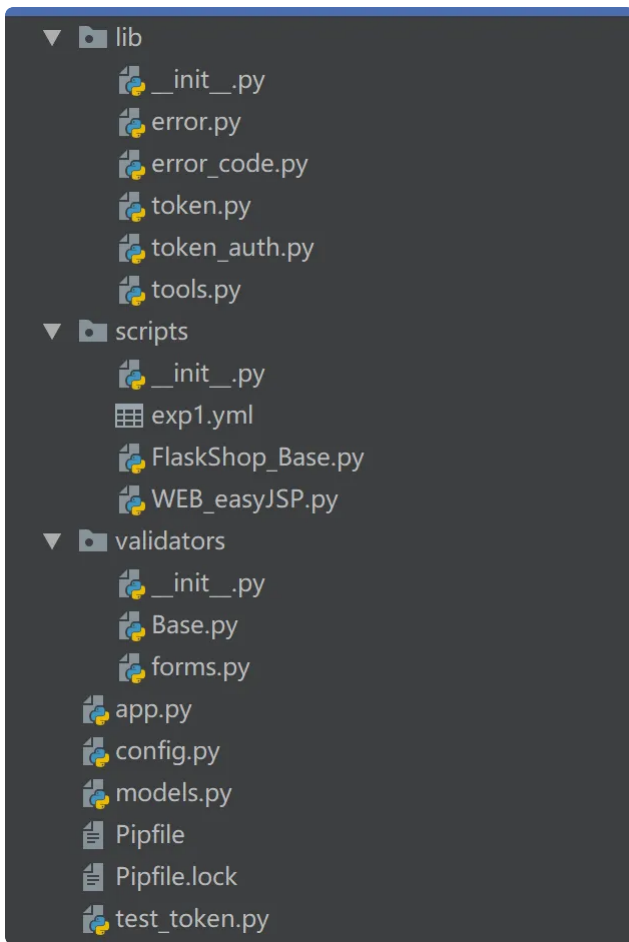
1. 该项目来源于一次AWD比赛。
2. 比赛需要编写漏洞exp脚本，完成对靶机状态的检测。
3. 初期靶机检测脚本是通过命令行的方式调用，数据的传入和输出通过excle表格完成
4. 初期命令行项目目录：



```
PWN-2-A_array_overflow.py
PWN-2-A_double_and_float.py
PWN-2-A_fmt_and_stack.py
PWN-2-A_leak_and_stack.py
PWN-2-A_stackover_and_arg.py
requirements.txt
score5.xlsx
score_4.xlsx
socre6.xlsx
spider.py
team.txt
test.py
title_to_name.xlsx
total_score.xlsx
Web-2-A_flatcore.py
Web-2-A_joo.py
Web-2-A_mac.py
Web-2-A_mineweb.py
web-3-chinaz1.py
Web-3-eyoucms.py
Web-4-A_XYH.py
web-4-news.py
web-4-pboot.py
Web-5-A_duomi.py
Web-easyWeb.py
WEB_easyJSP.py
Web_Think_Web_1.py
```

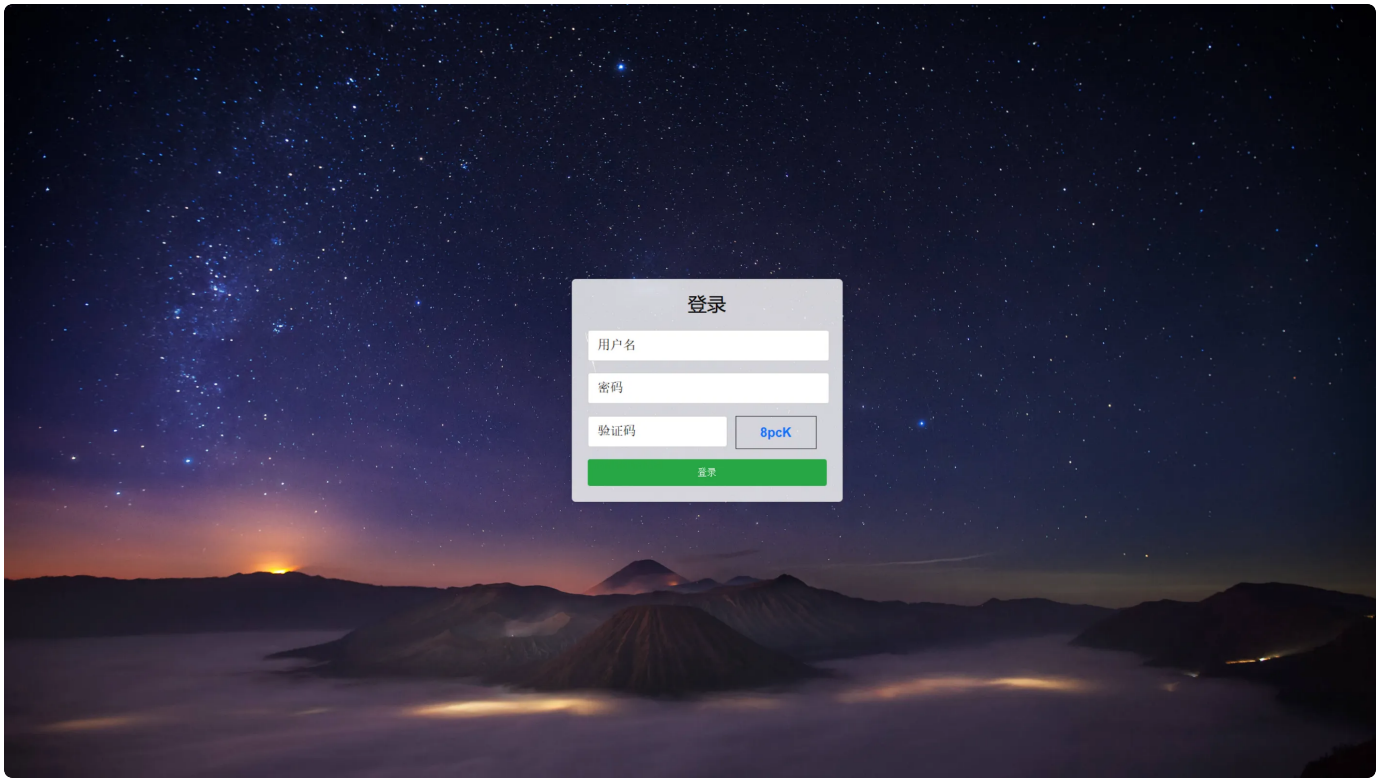
5. 为了更加方便的对脚本进行调用于是做了一个web版本，即为该项目

6. web项目目录



## 页面介绍:

1. 登录界面



## 2. 脚本信息显示及提交

[ip列表](#) [开启所有检测](#) [单个脚本检测记录](#) [批量脚本检测记录](#) [脚本展示](#)

脚本名称:

[新增](#)

[提交脚本](#)

脚本名称: [FlaskShop\\_Base\\_2](#) 漏洞数量: 2

222

4444

## 3. 上传批量检测信息

[ip列表](#)[开启所有检测](#)[单个脚本检测记录](#)[批量脚本检测记录](#)[脚本展示](#)[导入ip](#)名称: 附件:  未选择任何文件

#### 4. 开启多个靶机同时检测，并返回检测记录

[ip列表](#)[开启所有检测](#)[单个脚本检测记录](#)[批量脚本检测记录](#)[脚本展示](#)

脚本名称	轮次名称	队伍名称	漏洞描述	ip	靶机是否访问	靶机主页是否404	漏洞是否存在	漏洞主页是否404
WEB_easyJSP	名称1	队伍1	第一处: /forget.jsp 存在命令执行	192.168.93.165	否	否	否	否
FlaskShop_Base	名称1	队伍1	第一处: /search 命令执行	192.168.93.165	否	否	否	否
FlaskShop_Base	名称1	队伍1	第二处: /upload yaml反序列漏洞	192.168.93.165	否	否	否	否

#### 5. 批量脚本检测记录

[ip列表](#)[单个脚本检测记录](#)[批量脚本检测记录](#)[脚本展示](#)

脚本名称	轮次名称	队伍名称	漏洞描述	ip	靶机是否访问	靶机主页是否404	漏洞是否存在	漏洞主页是否404
WEB_easyJSP	测试-5questions	队伍1	第一处: /forget.jsp 存在命令执行	192.168.93.167	否	否	否	否
FlaskShop_Base	测试-5questions	队伍1	第一处: /search 命令执行	192.168.93.167	否	否	否	否
FlaskShop_Base	测试-5questions	队伍1	第二处: /upload yaml反序列漏洞	192.168.93.167	否	否	否	否

#### 6. 开启对单个靶机检测记录

脚本名称	漏洞检测点	漏洞描述	ip	靶机是否访问	靶机主页是否404	漏洞是否存在	漏洞主页是否404
FlaskShop_Base	1	第一处：/search 命令执行	192.168.93.167	是	否	是	否
FlaskShop_Base	2	第二处：/upload yaml反序列漏洞	192.168.93.167	是	否	是	否