

**课 程 实 验 报 告**

**课程名称： 信息安全综合设计与实践**

**专业班级： xxxx**

**学 号： xxx**

**姓 名： xxx**

**报告日期： 2022年9月30日**

**目录**

[一、实验过程 1](#_Toc115394986)

[1.1第1轮Flag获取过程描述 1](#_Toc115394987)

[1.2第1轮防护过程描述 3](#_Toc115394988)

[1.3第2轮Flag获取过程描述 3](#_Toc115394989)

[1.4第3轮Flag获取过程描述 4](#_Toc115394990)

[1.5第2轮防护过程描述 6](#_Toc115394991)

[1.6第4轮Flag获取过程描述 6](#_Toc115394992)

[1.7第3轮防护过程描述 8](#_Toc115394993)

[1.8第5轮Flag获取过程描述 9](#_Toc115394994)

[1.9第4轮防护过程描述 10](#_Toc115394995)

[二、实验心得 10](#_Toc115394996)

[三、对实验内容和过程的建议 11](#_Toc115394997)

## 一、实验过程

### 1.1第1轮Flag获取过程描述

（1）使用xray工具扫描漏洞，如图1-1所示。

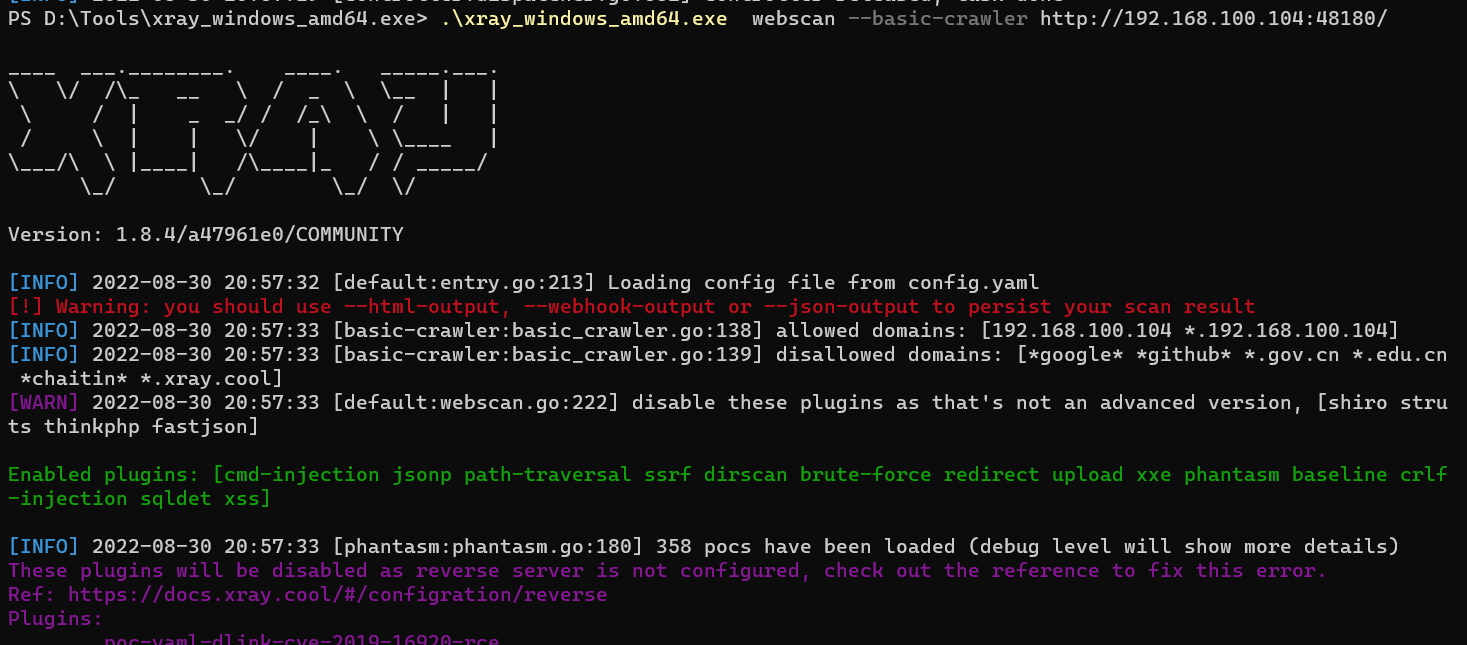


图1-1 xray工具扫描结果

发现文件上传漏洞，如图1-2所示。



图1-2 文件上传漏洞

（2）定位到contact.html页面，分析发现只允许上传图片文件。因此，采用Burp Suite+中国蚁剑利用此漏洞。

准备好一句话木马文件，如图1-3所示。



图1-3 一句话木马

在Burp Suite的浏览器中上传文件，截包，如图1-4所示。

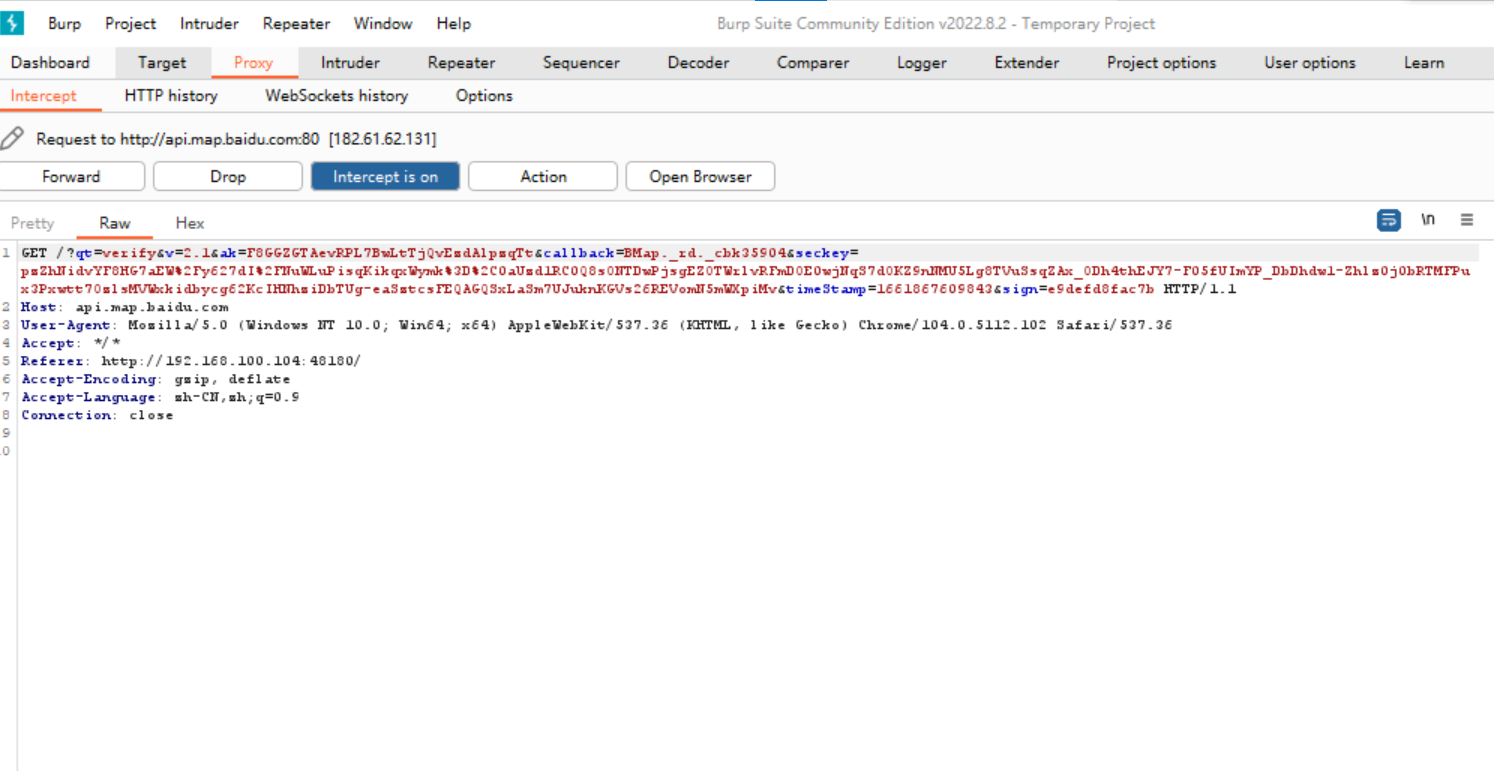


图1-4 BurpSuite截包

修改Content-Type的内容为image/png，如图1-5所示。

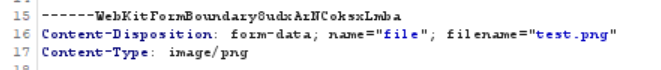


图1-5 修改类型

成功上传文件，如图1-6所示。



图1-6 成功上传文件

使用中国蚁剑连接，如图1-7所示。

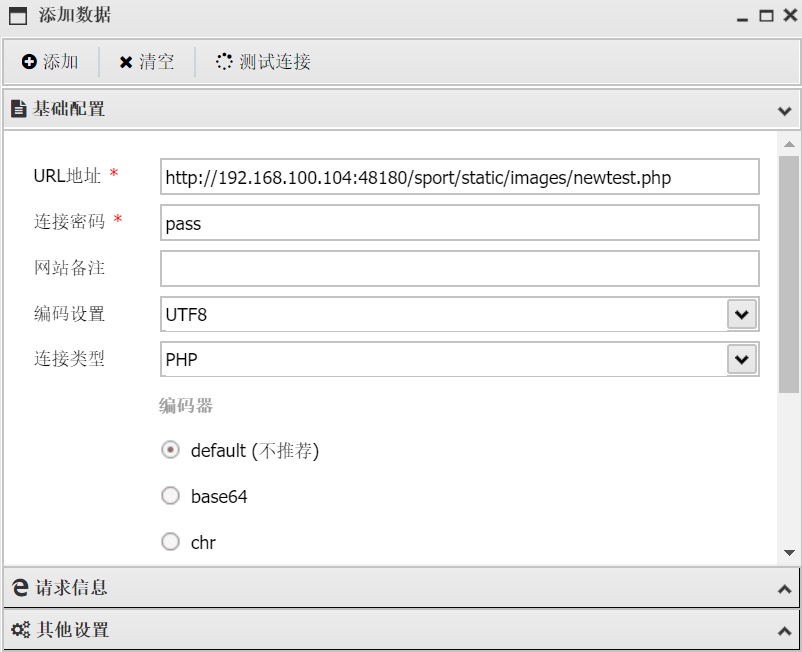


图1-7 蚁剑连接

查看flag，如图1-8所示。

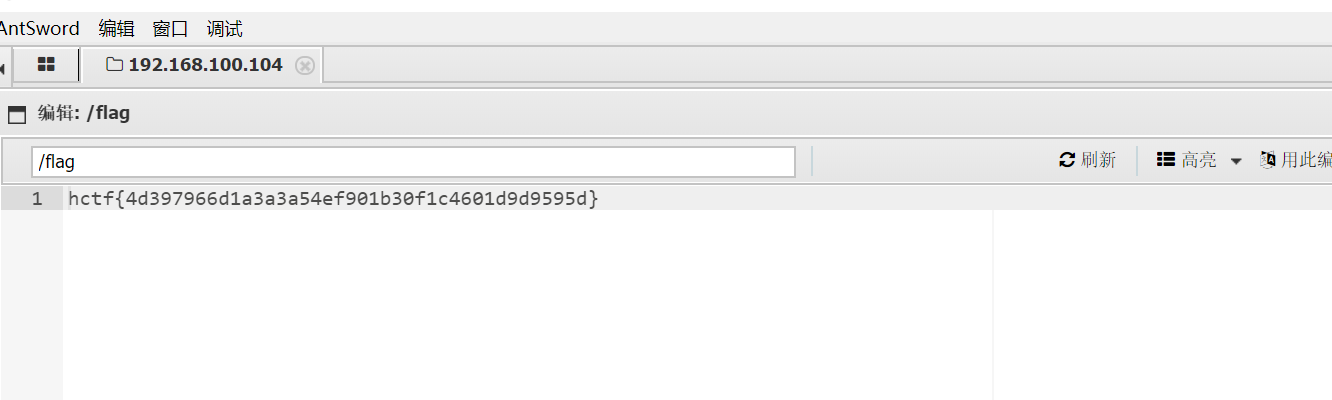


图1-8 获得flag

（3）小结：第一轮Flag获取主要是利用文件上传漏洞。关键在于使用Burp Suite截包修改内容从而绕过文件上传限制，便可使用中国蚁剑连接。

### 1.2第1轮防护过程描述

（1）分析负责文件上传的php文件，发现上传文件的特征是文件名前缀固定为new。尝试修改php文件代码，jaruser用户权限不够。

（2）编写并运行shell脚本，不断删除文件上传目录下以new开头的文件，从而实现防护，如图1-9所示。

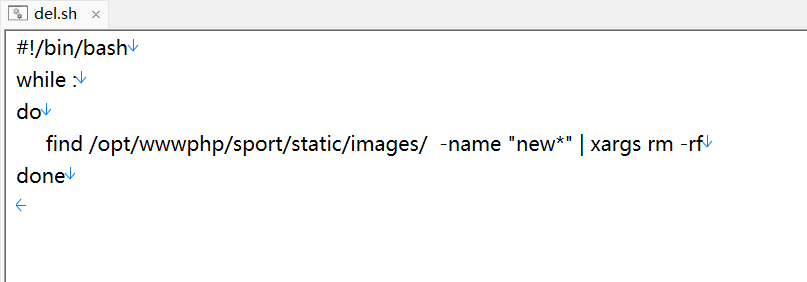


图1-9 批量删除脚本

（3）小结：在用户权限有限的前提下，无法直接修复代码漏洞，只能通过不断删除上传的木马文件来实现防护。

### 1.3第2轮Flag获取过程描述

（1）使用xray工具扫描，发现命令注入漏洞，如图1-10所示。

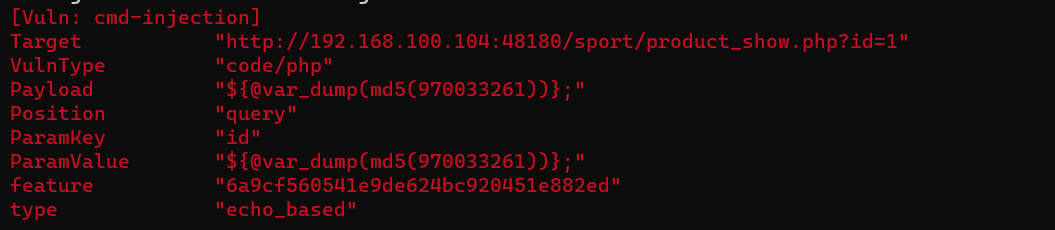


图1-10 命令注入漏洞

查看product\_show.php文件，发现assert函数是代码执行点。

（2）构造特殊的url，查看目标服务器上的flag，如图1-11所示。

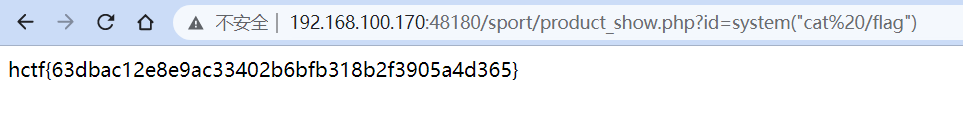


图1-11 构造url攻击

编写批量攻击的脚本文件，如图1-12所示。

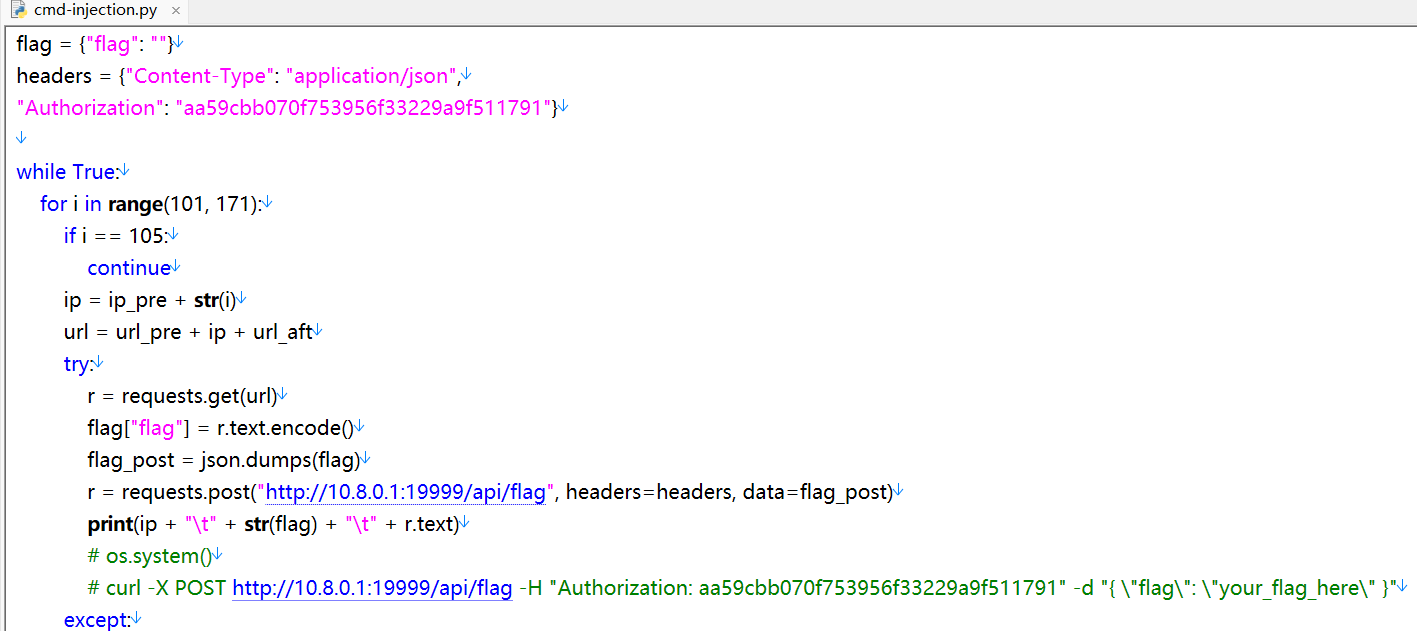


图1-12 批量攻击脚本

批量攻击效果明显，如图1-13所示。

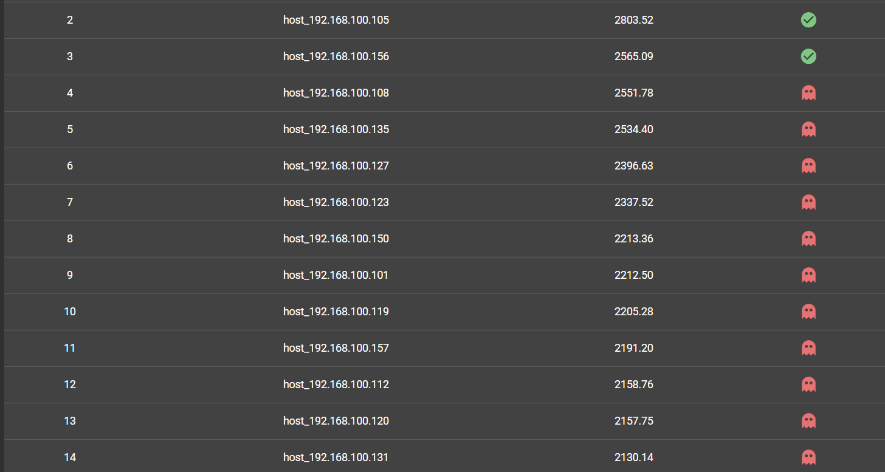


图1-13 批量攻击成功

（3）小结：此轮攻击主要是利用命令注入漏洞，先要构造出url获得flag，接着为了快速得分，需要编写批量脚本攻击。

### 1.4第3轮Flag获取过程描述

（1）使用xray工具扫描，发现SQL注入漏洞，如图1-14所示。



图1-14 SQL注入漏洞

（2）使用sqlmap工具进行攻击，爆库获得flag，如图1-15、图1-16所示。

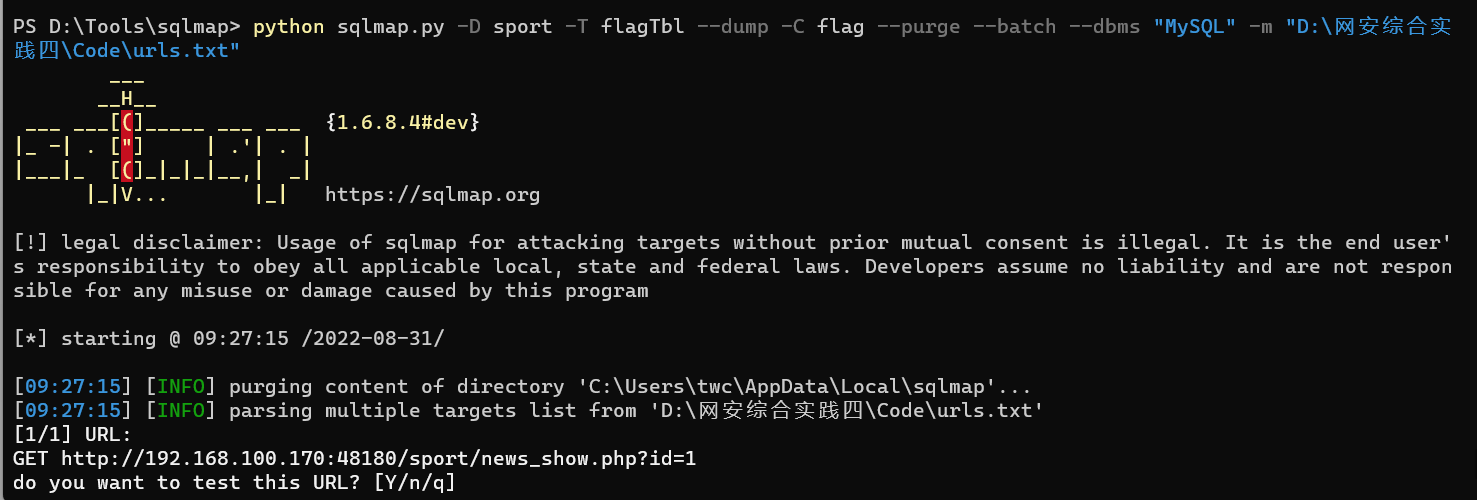


图1-15 sqlmap工具攻击

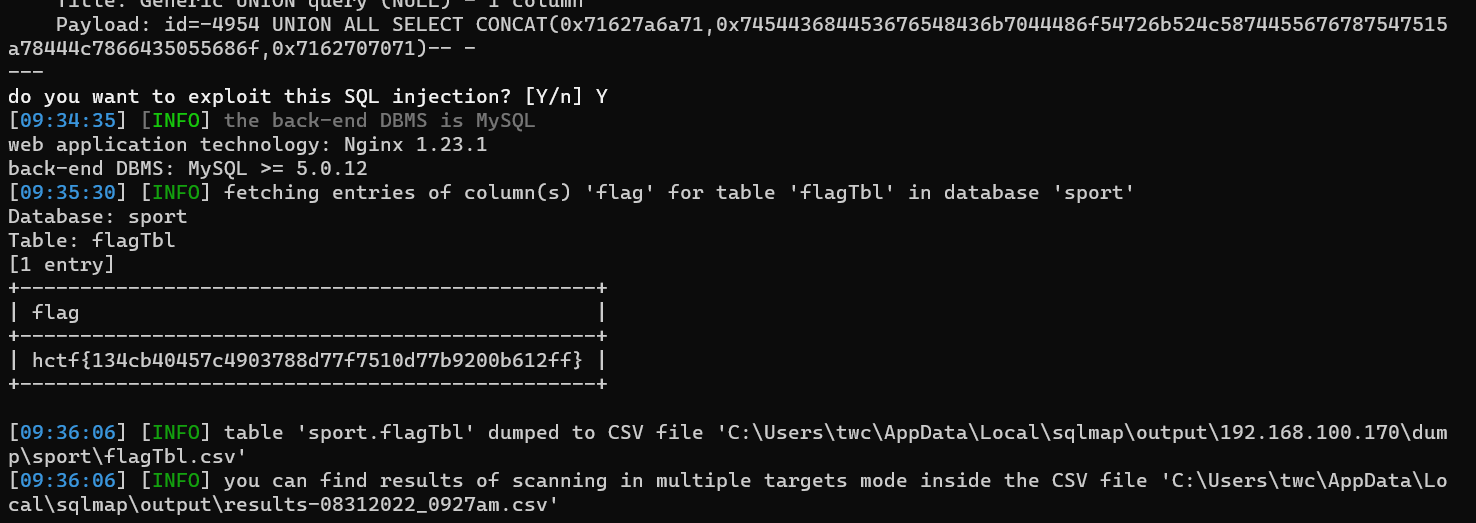


图1-16 爆库获得flag

查看sqlmap工具攻击的PAYLOAD，利用PAYLOAD编写批量攻击脚本，如图1-17所示。



图1-17 批量攻击脚本

（3）小结：此轮攻击主要是利用SQL注入漏洞。可以先利用sqlmap工具查看攻击方法，即PAYLOAD，再编写脚本进行批量攻击。

### 1.5第2轮防护过程描述

（1）开启modsecurity防护。修改nginx的配置文件，并重启nginx，如图1-18、图1-19所示。

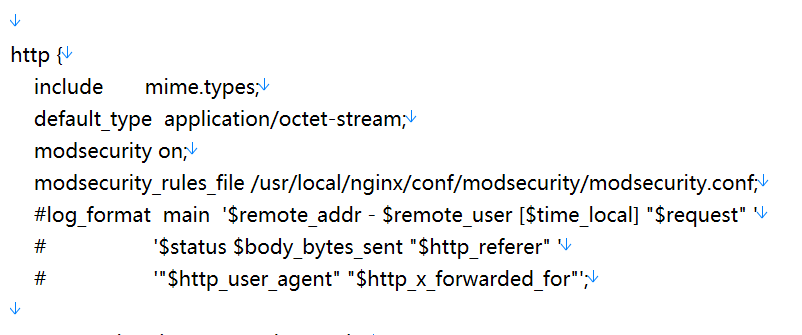


图1-18 修改modsecurity状态

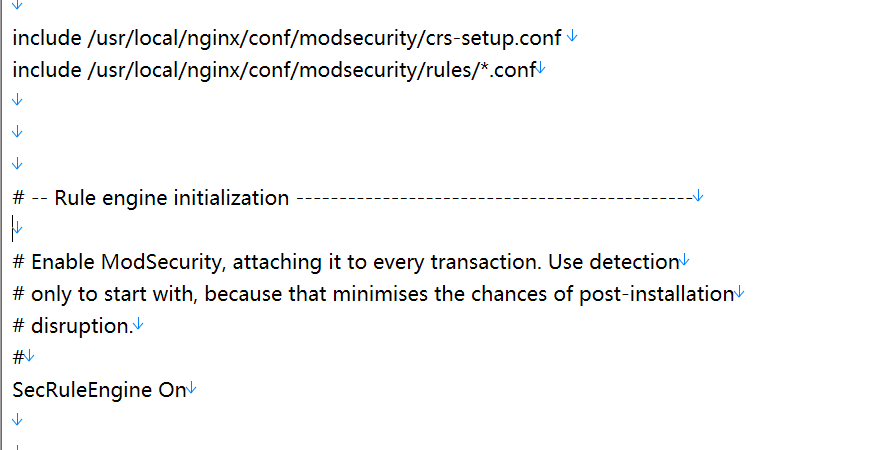


图1-19 修改规则配置文件

（2）重新攻击，消息包含的恶意代码被拦截，防护成功，如图1-20所示。

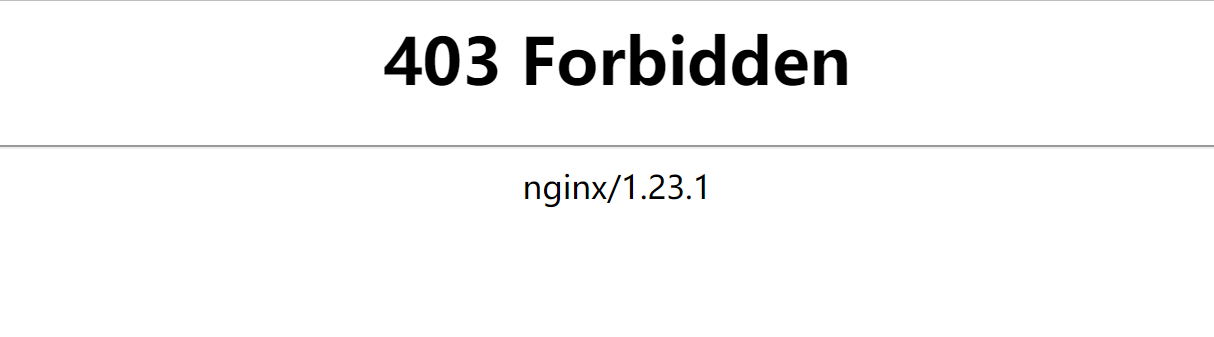


图1-20 防护成功

（3）小结：启用modsecurity，主机的抗攻击能力可以大大提升。

### 1.6第4轮Flag获取过程描述

（1）48181端口的Web服务是flask应用。通过网络搜索，发现flask模板注入攻击的相关资料，如图1-21所示。



图1-21 模板注入攻击

（2）构造url，获得目标主机上的flag，如图1-22所示。

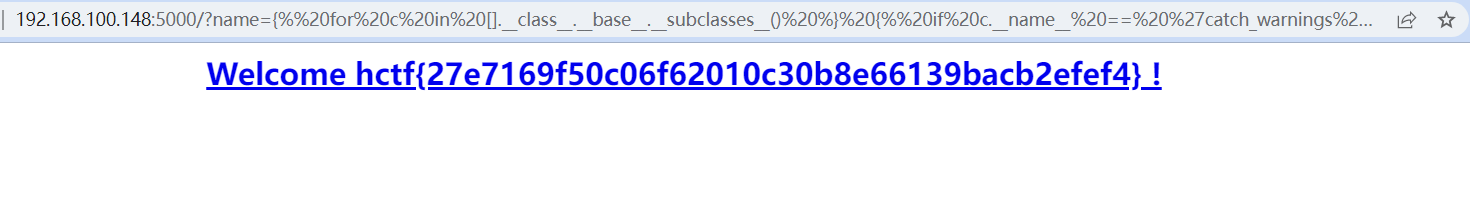


图1-22 构造url攻击

编写脚本批量攻击，如图1-23所示。

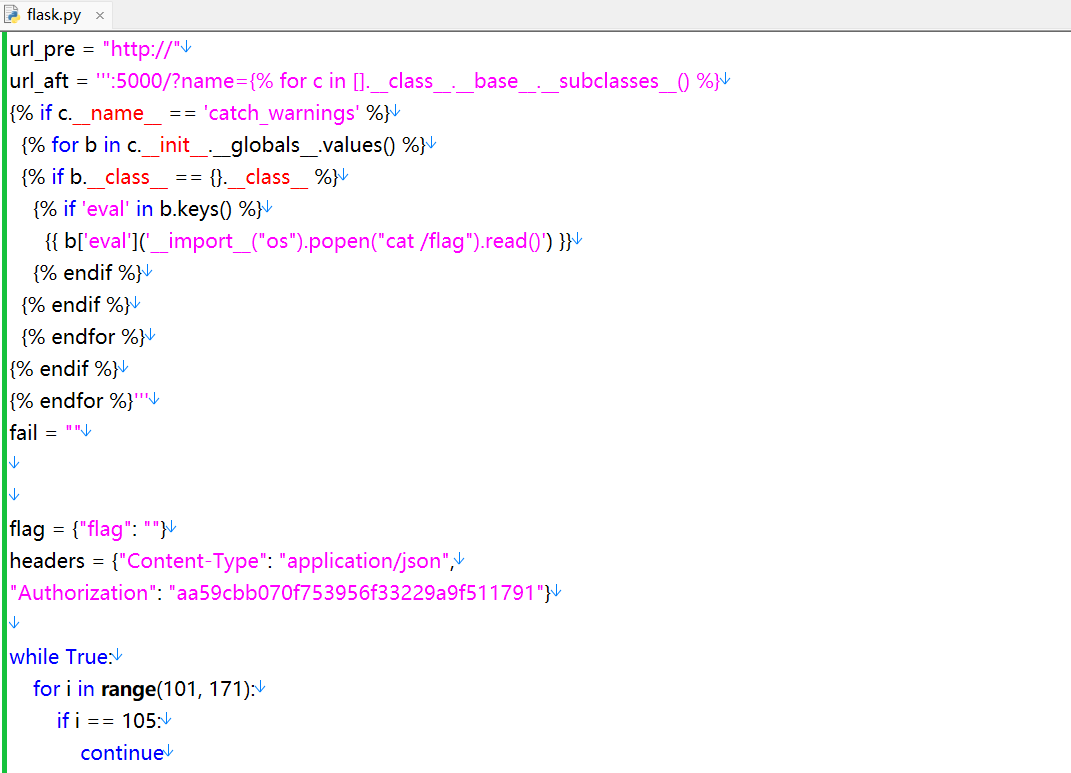


图1-23 批量脚本攻击

（3）小结：此轮攻击为flask模板注入攻击。需要先搜索资料了解大致原理后，构造url获得flag，再编写脚本进行批量攻击。

### 1.7第3轮防护过程描述

（1）开启modsecurity防护。修改nginx的配置文件，并重启nginx。

（2）modsecurity只可以防护48181端口，仍然可以通过5000端口进行flask模板注入攻击。因此，修改flask应用代码，并重新运行flask应用，如图1-24所示。

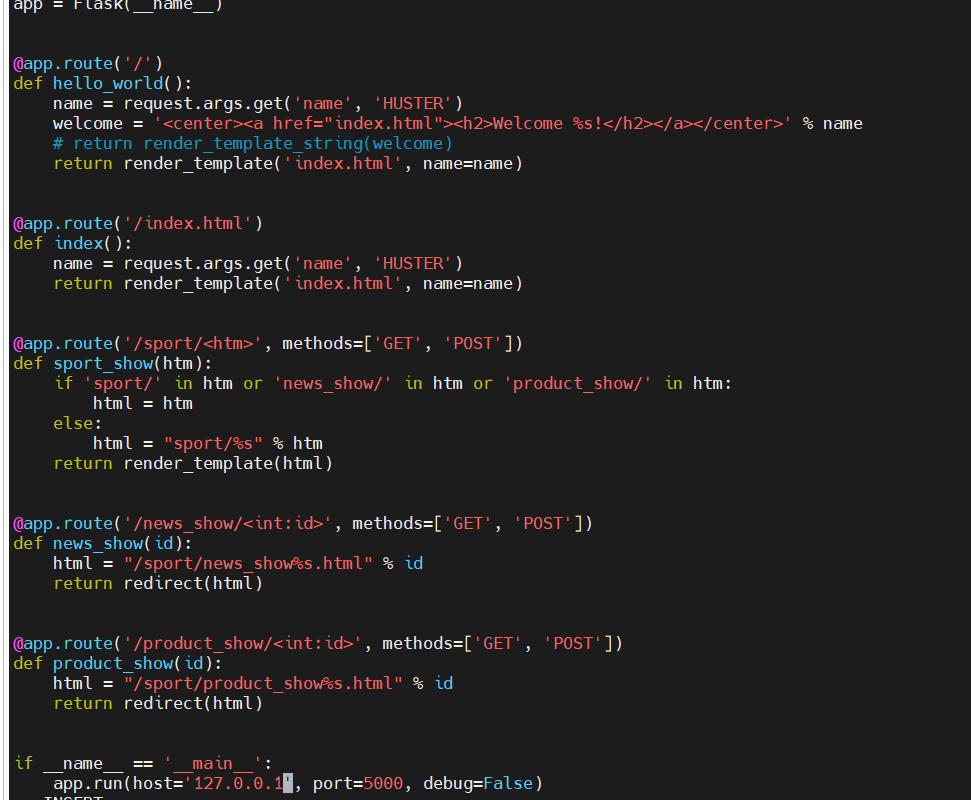


图1-24 修改flask应用运行配置

（3）小结：需要注意的是并非启用modsecurity防护后便可以防御针对flask的攻击。由于flask应用本身运行在5000端口，且modsecurity无法防护5000端口，必须修改代码逻辑，从而实现完全防御flask模板注入攻击。

### 1.8第5轮Flag获取过程描述

（1）首先使用提供的账号登录，同时使用Burp Suite截包，获得cookie信息，如图1-25所示。

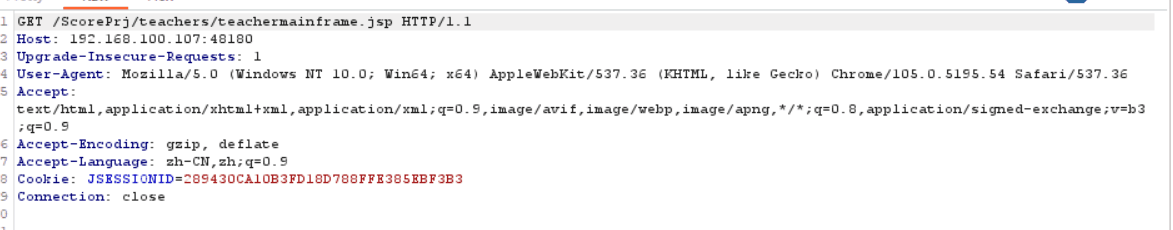


图1-25 获得cookie信息

（2）结合上一步获得的cookie信息，使用sqlmap工具对

<http://192.168.100.107:48180/ScorePrj/teachers/setscore.jsp?couno=001&action=queryscore>

进行爆库，如图1-26所示。执行的命令为： python sqlmap.py -D sport -T flagTbl --dump -C flag --purge --batch --dbms "MySQL" -m ".\urls.txt" -v 3 --cookie="JSESSIONID=F66D16C252328A71368FCBCAE0CB346C"。

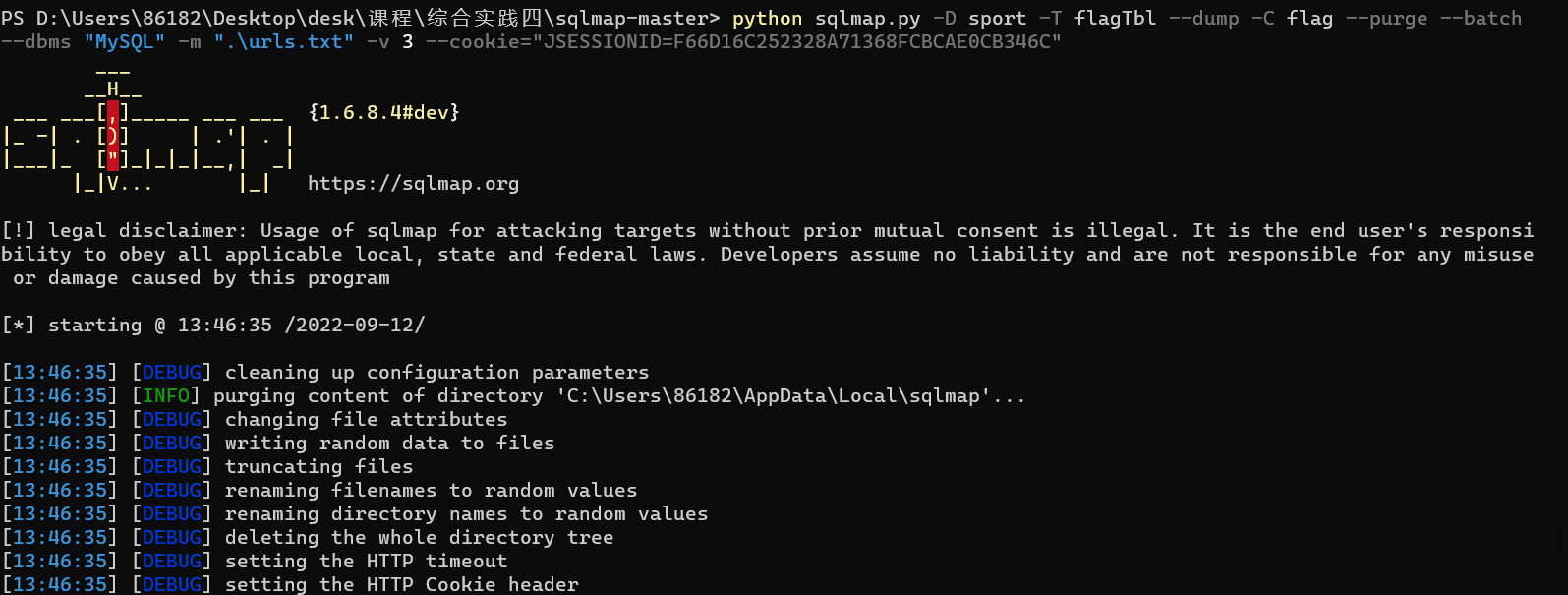


图1-26 基于cookie爆库

获得sqlmap使用的PAYLOAD，如图1-27所示。

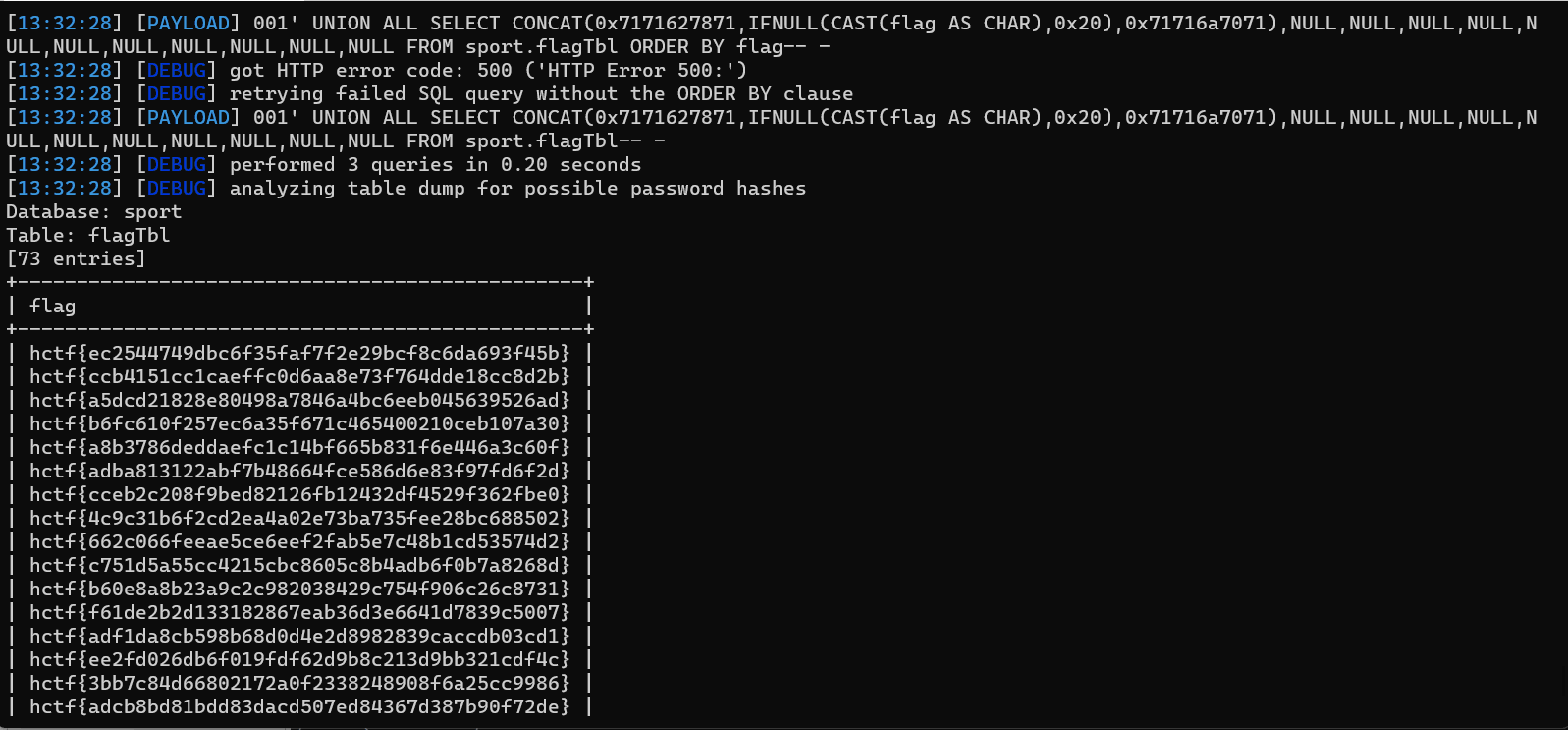


图1-27 爆库并获得PAYLOAD

（3）构造好url后，浏览器访问测试，成功获得flag，如图1-28所示。

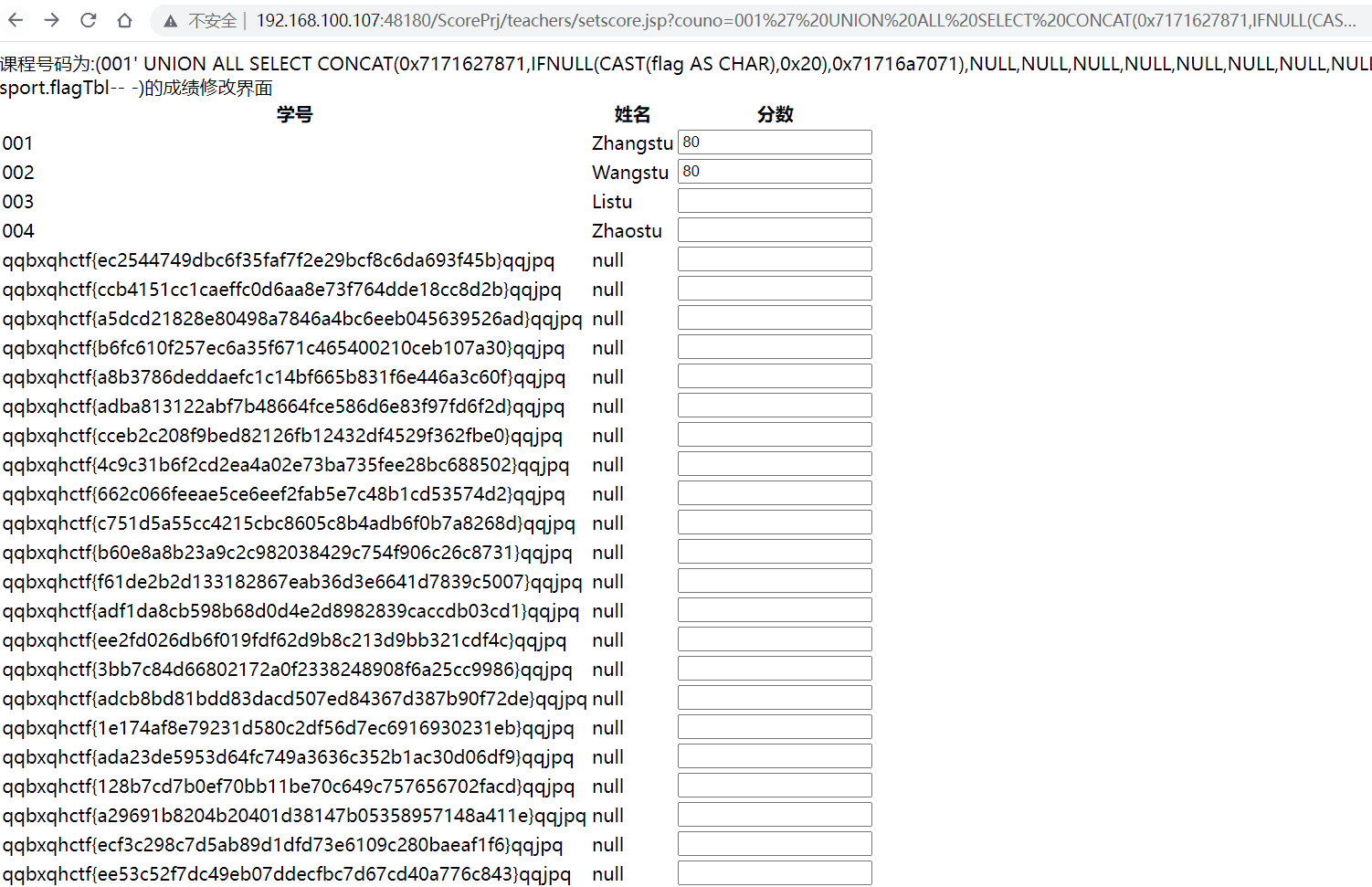


图1-28 构造url攻击

进一步编写批量攻击脚本，快速得分。

### 1.9第4轮防护过程描述

开启modsecurity防护即可，如图1-29所示。

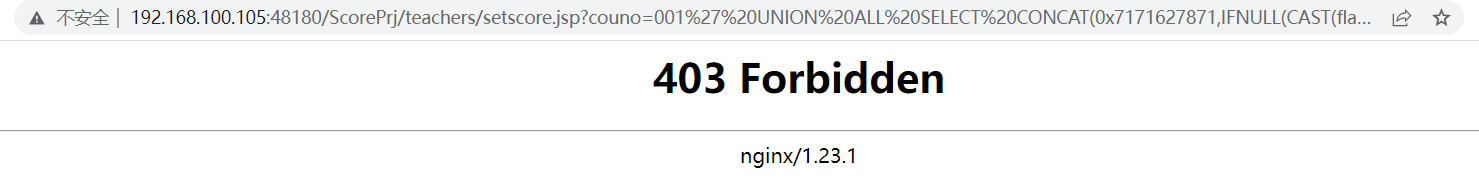


图1-29 防护成功

## 二、实验心得

网安综合实践四是前三次综合实践课知识的综合运用。一方面，通过靶场的仿真环境，将前三次课程内容在实践中应用并加强巩固。另一方面，也掌握了一些新的攻击方法，比如flask的模板注入攻击。

通过此次课程，我在实践中有一些深刻的体会。

（1）需要先对靶机进行分析，利用各种Web扫描工具，发现可能的攻击途径。

（2）对于可能的攻击方法，需要广泛搜寻资料，甚至是现学一些新的内容，想尽一切办法获得flag。

（3）确定攻击途径后，一方面，需要编写批量自动化脚本快速得分；另一方面，要尽快修复靶机漏洞。

总之，通过此次CTF AWD线下赛的形式，在实践中学习Web攻击与防护，收获很多，是一次难忘的经历。同时，我也体会到网络空间安全对国家安全的重要性。因此，我作为网络空间安全专业的学生，应当努力学习科学文化知识，不断提升个人专业技术能力，为国家网络安全事业的发展贡献自己的力量。

## 三、对实验内容和过程的建议

此次课程内容非常有趣且实用。考虑到大多数同学并没有CTF Web方面的基础，且前三次课程内容有些遗忘，希望有更多的引导过程。比如，Web扫描工具的介绍、对个别难的题目提供攻击的方向和切入点。