**Broker walkthrough**

# **Index**

[Index 1](#_Toc194242734)

[List of pictures 1](#_Toc194242735)

[Disclaimer 2](#_Toc194242736)

[Reconnaissance 2](#_Toc194242737)

[Initial foothold 4](#_Toc194242738)

[User flag 4](#_Toc194242739)

[Privilege escalation 5](#_Toc194242740)

[Personal comments 6](#_Toc194242741)

[Appendix A – CVE-2023-46604 6](#_Toc194242742)

[Appendix B – nginx exploit 6](#_Toc194242743)

[References 7](#_Toc194242744)

# **List of pictures**

[Figure 1 - nMap scan results (part 1) 2](#_Toc194242725)

[Figure 2 - nMap scan results (part 2) 3](#_Toc194242726)

[Figure 3 - nMap scan results (part 3) 3](#_Toc194242727)

[Figure 4 - Version of broker component 4](#_Toc194242728)

[Figure 5 - CVE-2020-46604 exploit 4](#_Toc194242729)

[Figure 6 - User shell 4](#_Toc194242730)

[Figure 7 - nginx server with malicious configuration file 5](#_Toc194242731)

[Figure 8 - SSH keys uploaded on the malicious server 5](#_Toc194242732)

[Figure 9 - Root shell 6](#_Toc194242733)

# **Disclaimer**

I do this box to learn things and challenge myself. I’m not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

# **Reconnaissance**

The results of an initial nMap scan are the following:

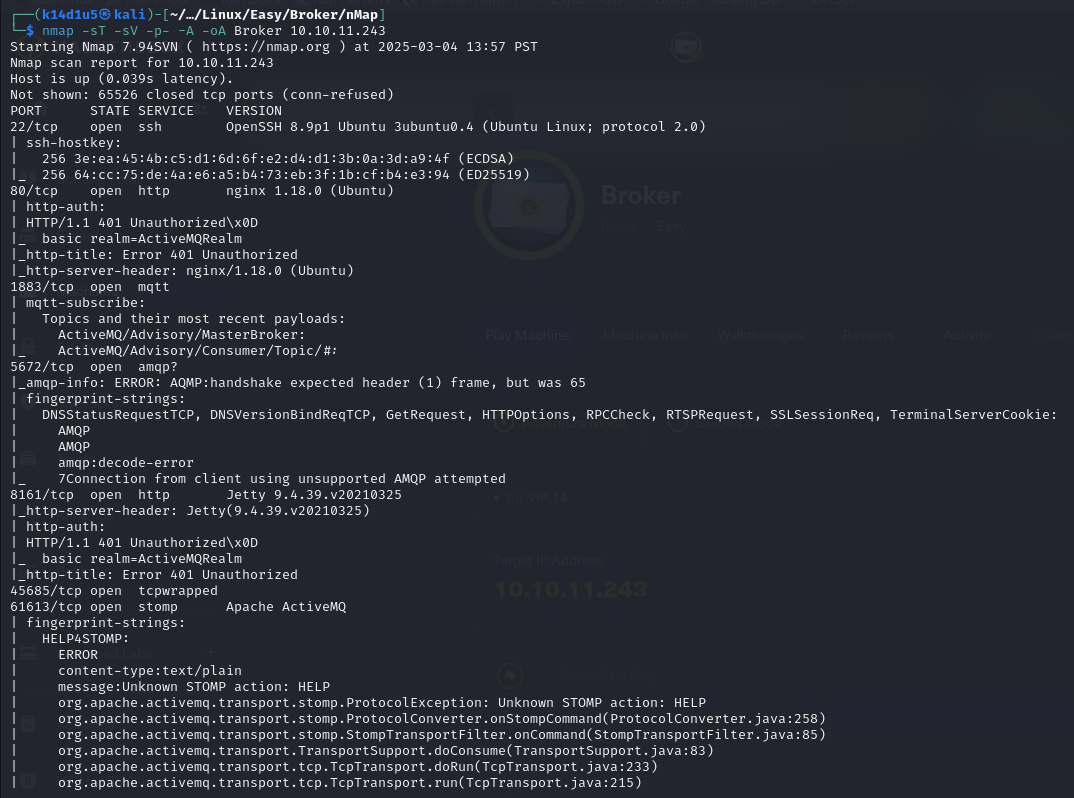


Figure 1 - nMap scan results (part 1)

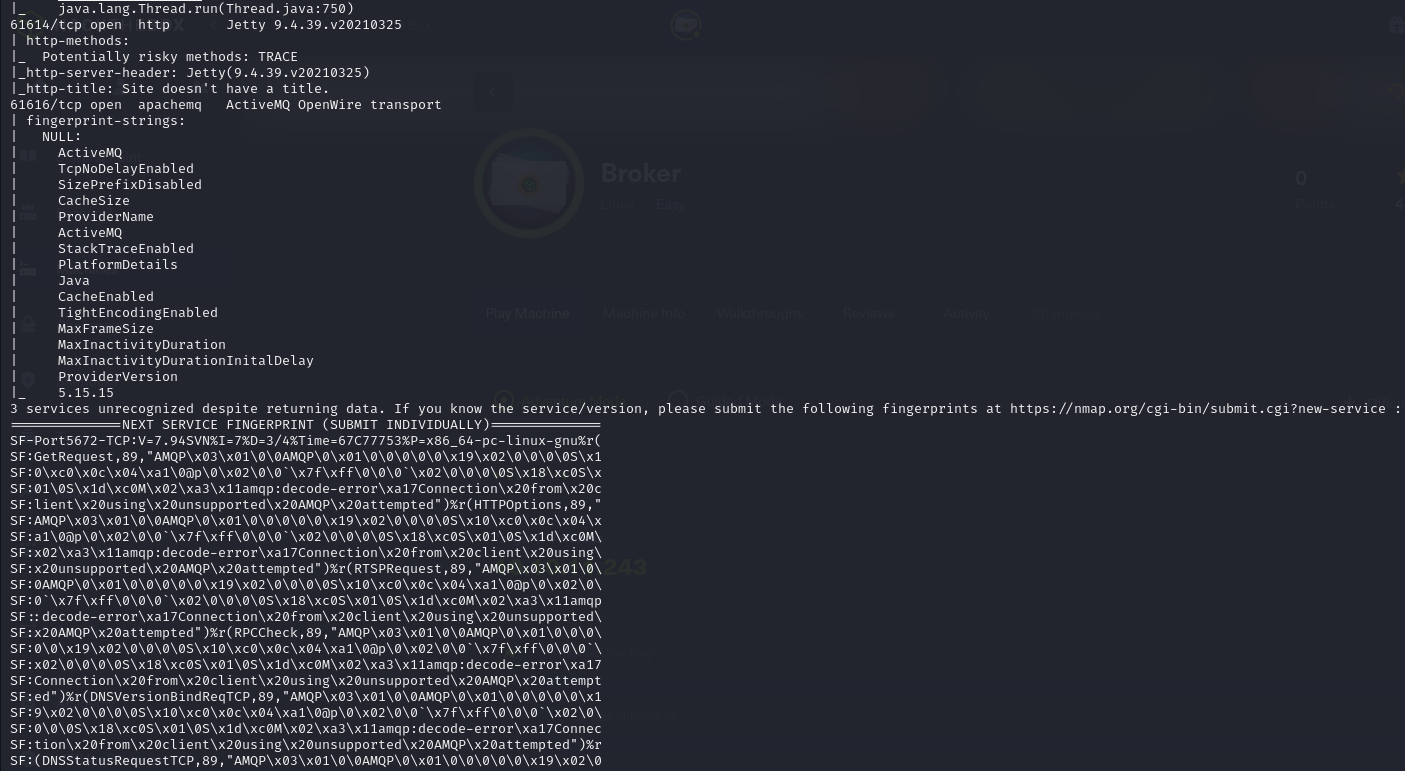


Figure 2 - nMap scan results (part 2)

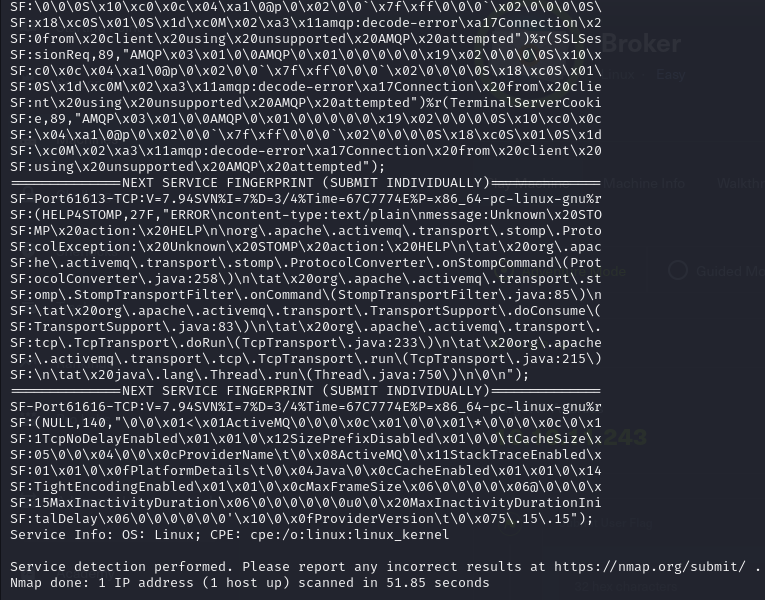


Figure 3 - nMap scan results (part 3)

Open ports are 22, 80, 1883, 5672, 8161, 45685, 61613, 61614 and 61616. So, enabled services are SSH (22), MQTT (1883), probably AMQP (5672), Stomp/Active MQ (61614, 61616). Also, three web application are running on port 80 and 8161, 61614. Lastly, an unknown service is running on port 45685. The last information nMap provided is that the Operative System was Linux.

# **Initial foothold**

First thing I tried to do was accessing to the web application on port 80. In this way I identified the application and, after an Internet search, I found default credentials for it. Also, I browsed to the application on the other ports and tried the credentials on them too. Luckly, the default credentials worked on application running on port 80 and on port 8161. So, I explored the web application on port 8161 and I found the version of the broker component, as shown in the following:

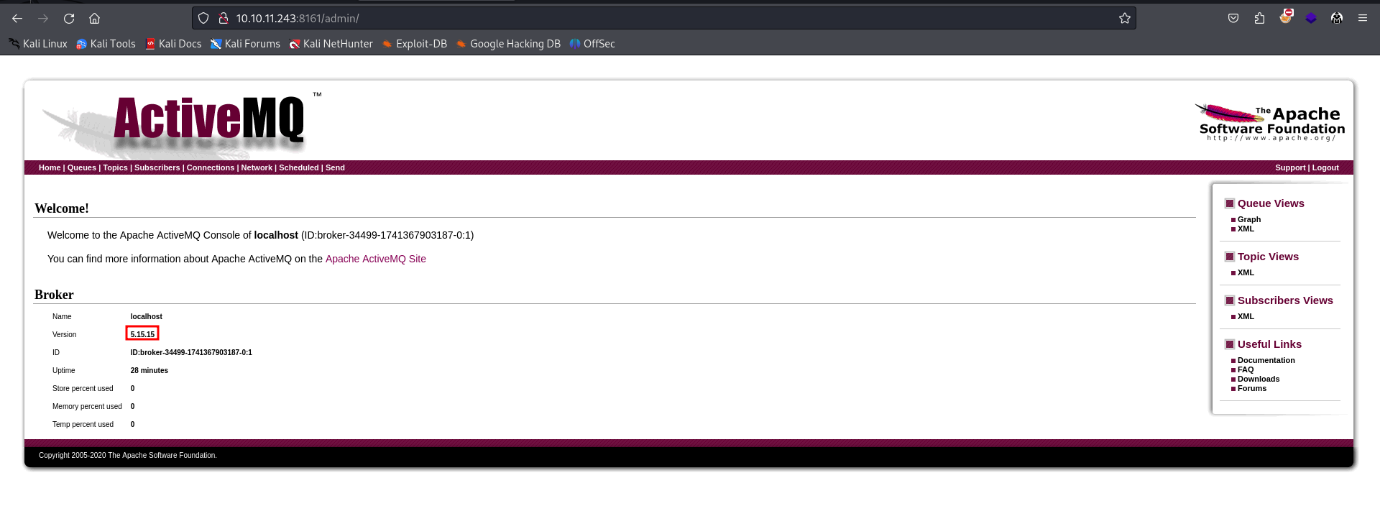


Figure 4 - Version of broker component

# **User flag**

Since I found the version of the broker component, I looked for an exploit on the Internet. In particular, I found the CVE-2023-46604 and I downloaded the exploit. This exploit was developed in GO, so I needed to install it and configure the relative environment variables. Also, I needed to change the payload in the file so it downloaded a malicious payload generated using MSFVenom, add the execution privileges to the file and executed it. Now that all was set, I run the exploit:

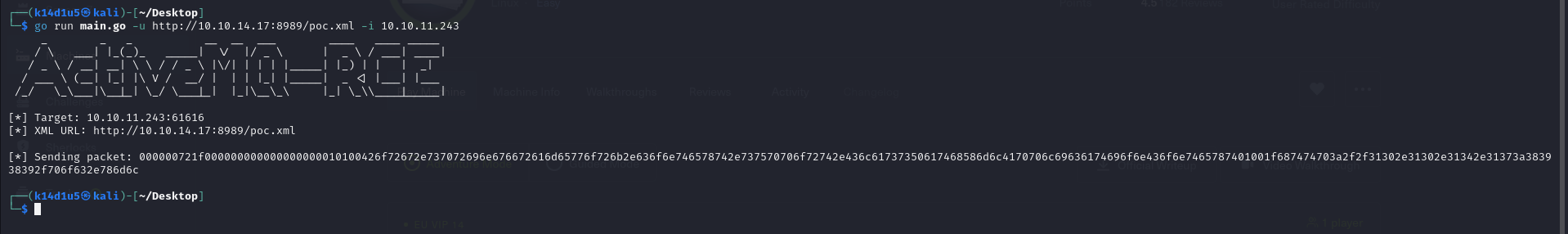


Figure 5 - CVE-2020-46604 exploit

In this way, I obtained the user shell, as shown in the following picture:

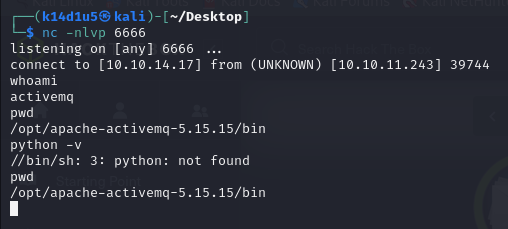


Figure 6 - User shell

Using it, I was able to retrieve the user flag, even if I forgot the screenshot.

# **Privilege escalation**

One of the first tests I do to perform privilege escalation was checking the sudoers running the command. In this way I found out that I was able to execute a ngnix server as root. I looked for an exploit on the Internet again and I found one. To execute it, I needed to upload a malicious configuration file and run a new server that use the malicious configuration:

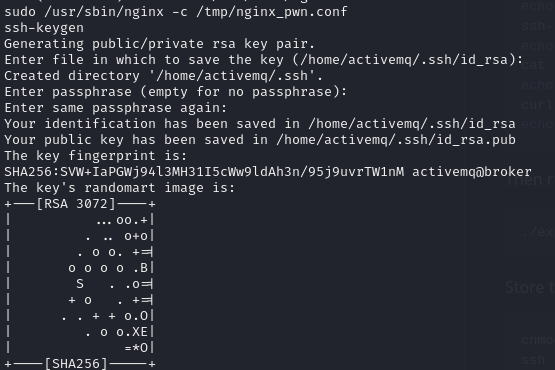


Figure 7 - nginx server with malicious configuration file

At this point I needed to upload the SSH keys on the new server:

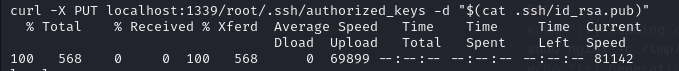


Figure 8 - SSH keys uploaded on the malicious server

So, all I needed to do was establishing an SSH connection using the SSH keys:



Figure 9 - Root shell

Again, I forgot to take a screenshot about the root flag.

# **Personal comments**

This box was very simply and linear, in my opinion. It was nice and I hadn’t any problem to solve it. It was a good exercise and can be a good point for beginners. I evaluate it as Easy on the HackTheBox platform.

# **Appendix A – CVE-2023-46604**

CVE-2023-46604 vulnerability classified as critical was found in [Apache ActiveMQ and ActiveMQ Legacy OpenWire Module up to 5.15.15/5.16.6/5.17.5/5.18.2](https://vuldb.com/?product.apache:activemq). This vulnerability affects an unknown code of the component **OpenWire Protocol Handler**. The manipulation with an unknown input leads to a deserialization vulnerability. The product deserializes untrusted data without sufficiently verifying that the resulting data will be valid. As an impact it is known to affect confidentiality, integrity, and availability. In this way, a malicious user can remotely execute arbitrary commands.

# **Appendix B – nginx exploit**

The exploit against nginx I run worked because I was able to run nginx as root. In particular, I was able to use a custom configuration file where I can specify **root** as user to run a new nginx server. Also, the logic behind the exploit is:

* Run a new nginx server instance as root using a custom configuration file;
* Create SSH keys correlated to the attacker;
* Upload the public key in the root’s SSH folder via the new nginx server.

At this point, a malicious user just needs to connect via SSH to the target machine to obtain a root shell.

# **References**

1. CVE-2023.46604: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-46604>;
2. Nginx exploit: <https://gist.github.com/DylanGrl/ab497e2f01c7d672a80ab9561a903406?permalink_comment_id=5322813>;
3. Nginx configuration file: <https://www.html.it/pag/377241/configurazione-il-file-nginx-conf/>.