**Jerry walkthrough**

# **Index**

[Index 1](#_Toc169526767)

[List of pictures 1](#_Toc169526768)

[Disclaimer 2](#_Toc169526769)

[Reconnaissance 2](#_Toc169526770)

[Initial foothold 2](#_Toc169526771)

[User flag 3](#_Toc169526772)

[Privilege escalation 4](#_Toc169526773)

# **List of pictures**

[Figure 1 - nMap scan results 2](#_Toc169526777)

[Figure 2 - Apache Tomcat default page 2](#_Toc169526778)

[Figure 3 - DirBuster search results 3](#_Toc169526779)

[Figure 4 - Apache Tomcat manager page 3](#_Toc169526780)

[Figure 5 - Root shell 4](#_Toc169526781)

# **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’re willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

# **Reconnaissance**

The results of an initial nMap scan are the following:

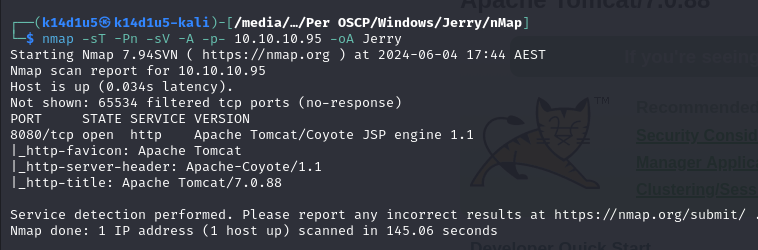


Figure 1 - nMap scan results

Only open port is 8080. There is a web application running on this port. Sadly, nMap didn’t provide any OS information.

# **Initial foothold**

Since I have only one port open, I try to analyze the web application on it. However, any web application is deployed and I found the Apache Tomcat default page:

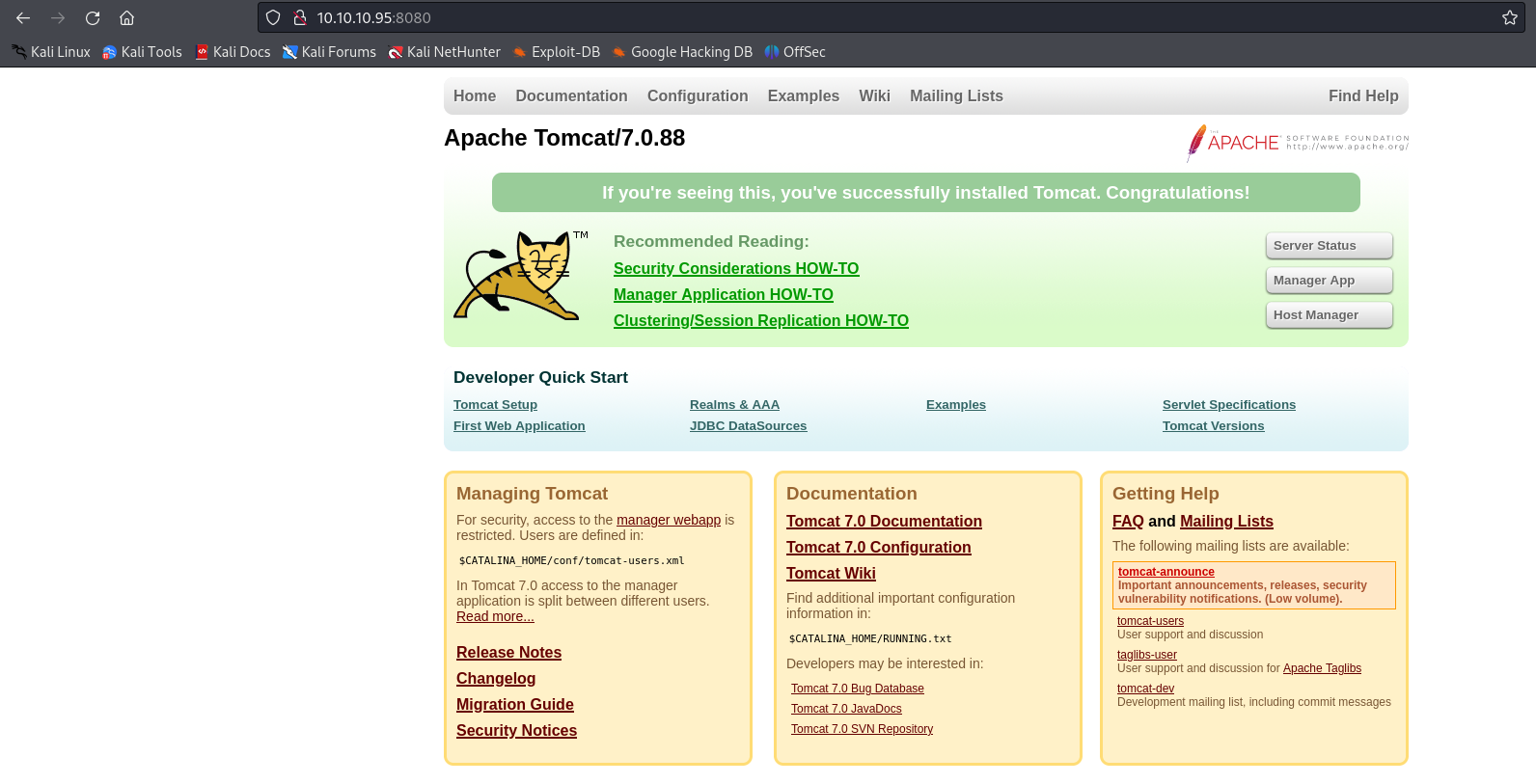


Figure 2 - Apache Tomcat default page

So, I need something else to exploit this box. I run DirBuster and I found a manager page:

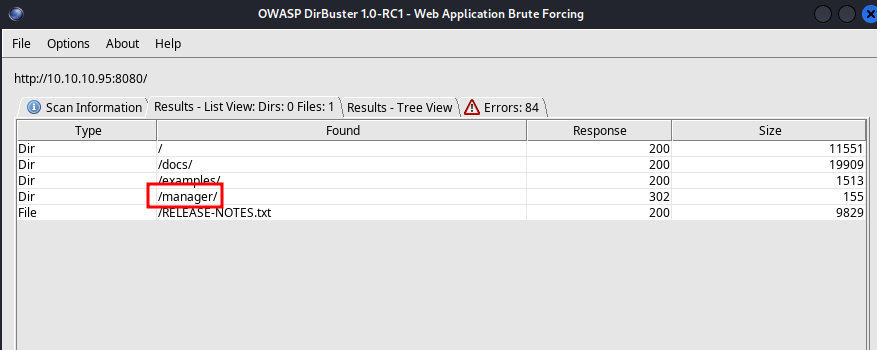


Figure 3 - DirBuster search results

# **User flag**

At this point I tried to access to this manager page, but it requires a login. So, I searched on the Internet the default credential for this Apache Tomcat version and luckily them worked! The manager page is:

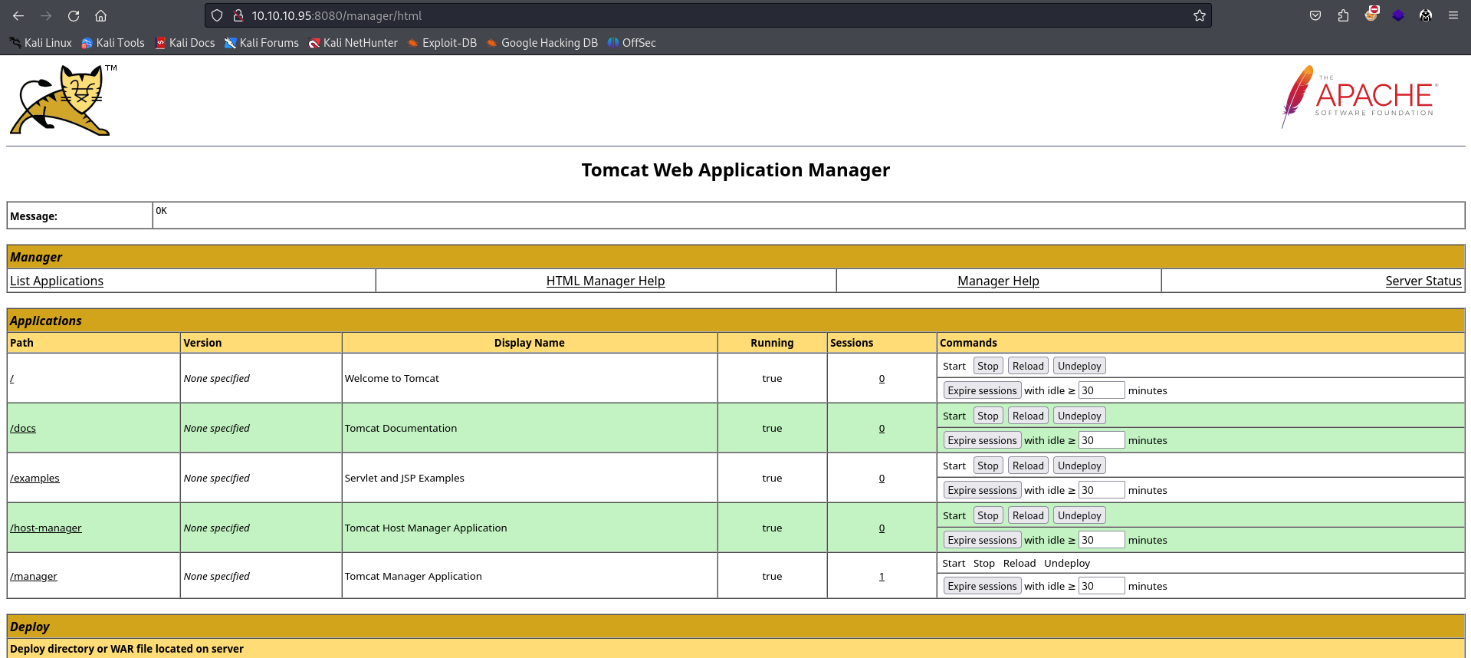


Figure 4 - Apache Tomcat manager page

This page allows me to upload a **.war** file. So, I created a malicious one using **msfvenom** running the command:

At this point I uploaded this malicious file, opened a listener on my Kali machine and invoked the malicious application. In this way, I obtained a reverse shell. Unpredictable, I already was on the system:

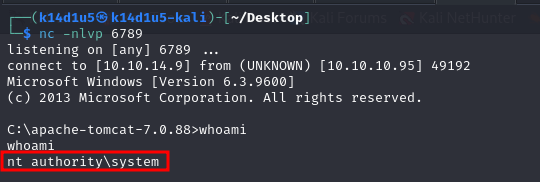


Figure 5 - Root shell

All I need to do was retrieving the user and root flags (I forgot flags screen, sorry).

# **Privilege escalation**

I didn’t need to elevate my privileges due to I obtained a shell as yet and I had already retrieved the root flag too.