**Knife walkthrough**

# **Index**

[Index 1](#_Toc157497490)

[List of pictures 1](#_Toc157497491)

[Disclaimer 2](#_Toc157497492)

[Reconnaissance 2](#_Toc157497493)

[Initial foothold 2](#_Toc157497494)

[User flag 3](#_Toc157497495)

[Privilege escalation 4](#_Toc157497496)

# **List of pictures**

[Picture 1 - nMap scan results 2](#_Toc157497500)

[Picture 2 - Application running on port 80 3](#_Toc157497501)

[Picture 3 - Nikto scan results 3](#_Toc157497502)

[Picture 4 - Listener on my attacker machine 3](#_Toc157497503)

[Picture 5 - Payload to get a reverse shell 3](#_Toc157497504)

[Picture 6 - User reverse shell 4](#_Toc157497505)

[Picture 7 - User flag 4](#_Toc157497506)

[Picture 8 - Privilege escalation 4](#_Toc157497507)

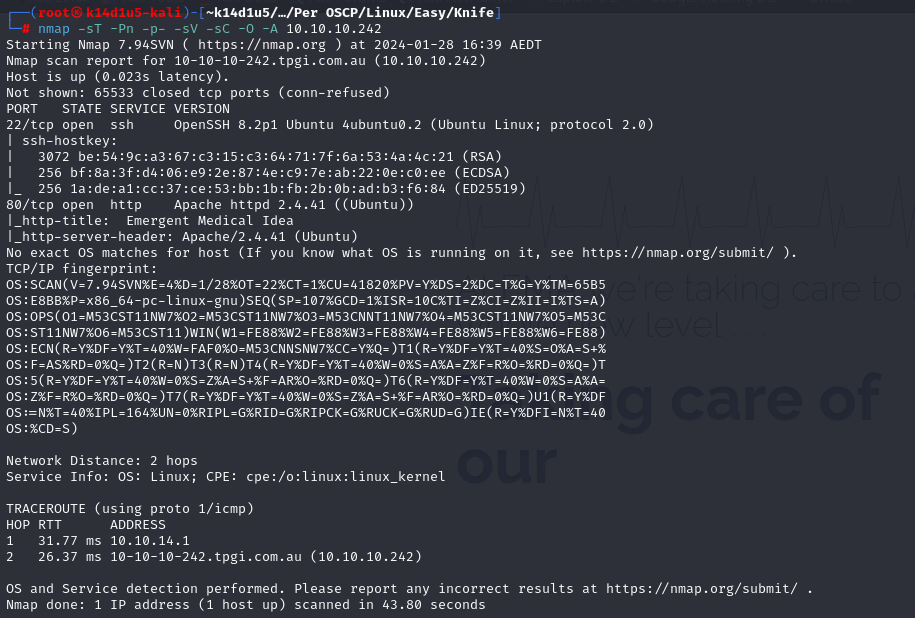
[Picture 9 - Root flag 5](#_Toc157497508)

# **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:

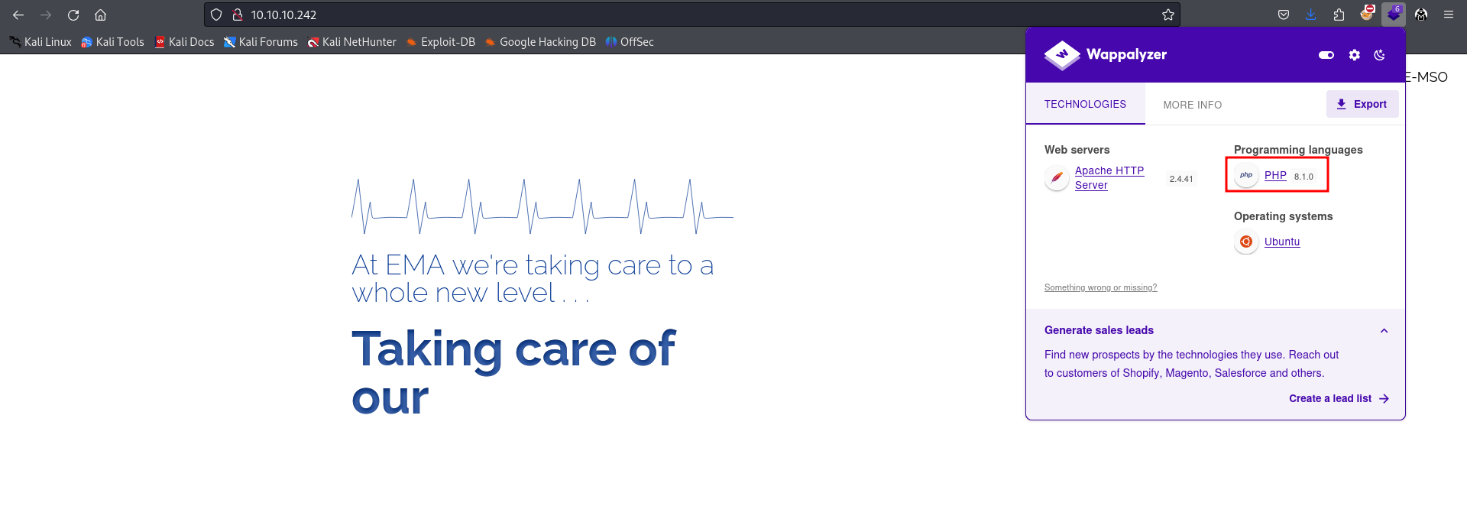


Picture 1 - nMap scan results

Open ports are 22 and 80. So, the machine had SSH enabled and an application running on port 80. NMap recognized the operative system as Linux, probably Ubuntu.

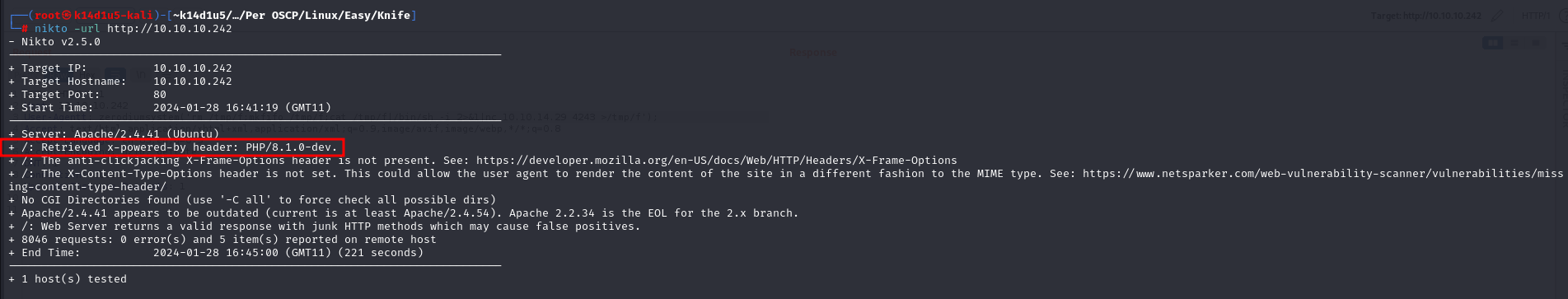
# **Initial foothold**

I access to the application running on port 80 and it looked like:



Picture 2 - Application running on port 80

As shown in the previous picture, this application is developed in PHP version 8.1.0. This information is confirmed by a **Nikto** scans too:

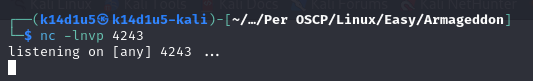


Picture 3 - Nikto scan results

This PHP version is known to be vulnerable to RCE using a custom header in the request.

# **User flag**

To obtain a user shell, I opened a listener on my attacker machine:



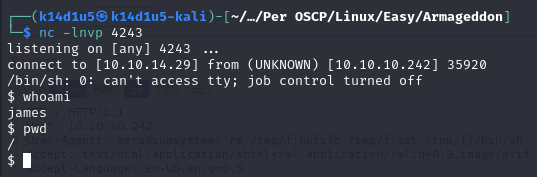
Picture 4 - Listener on my attacker machine

At this point, I used the following payload in request to get a reverse shell:



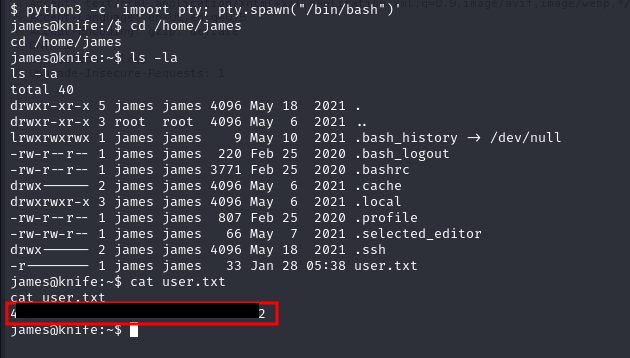
Picture 5 - Payload to get a reverse shell

When I sent this request, I got the reverse shell, as shown in the next picture:



Picture 6 - User reverse shell

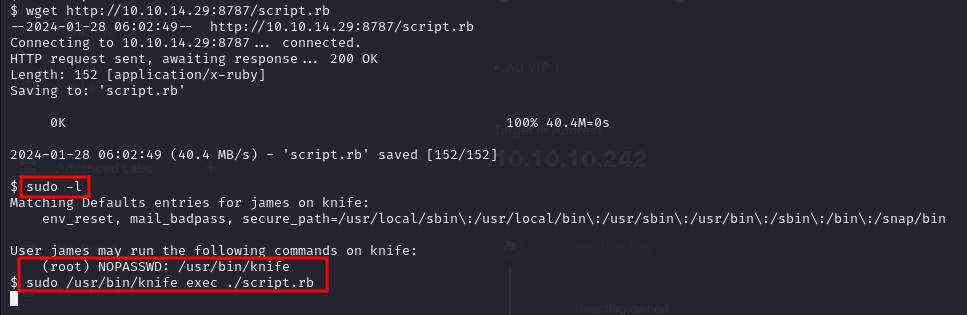
So, I easily got the user flag after stabled the shell:



Picture 7 - User flag

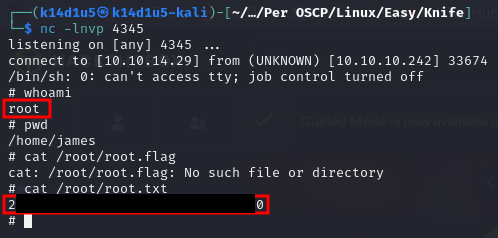
# **Privilege escalation**

At this point I needed to escalate my privileges on the machine. To accomplish this goal, the useful information is the following one:



Picture 8 - Privilege escalation

In the previous picture, I showed the command to escalate my privilege too. In fact, **knife** tool let me to run a ruby script. In particular, **script.rb** file was developed by me and I coded it to open a new shell. Executing this script as root, I got a root reverse shell and I retrieved the root flag:



Picture - Root flag