**Blackpearl walkthrough**

# **Index**

[Index 1](#_Toc204621636)

[List of pictures 1](#_Toc204621637)

[Disclaimer 2](#_Toc204621638)

[Reconnaissance 2](#_Toc204621639)

[Initial foothold 2](#_Toc204621640)

[User flag 3](#_Toc204621641)

[Privilege escalation 4](#_Toc204621642)

[Personal comments 5](#_Toc204621643)

[Appendix A – CVE-2018-17552 5](#_Toc204621644)

[Appendix B – CVE-2018-17553 5](#_Toc204621645)

[References 5](#_Toc204621646)

# **List of pictures**

[Figure 1 - nMap scan results 2](#_Toc204621609)

[Figure 2 - Path to the web application on the domain 3](#_Toc204621610)

[Figure 3 - Application name and version 3](#_Toc204621611)

[Figure 4 - Metasploit exploit module 4](#_Toc204621612)

[Figure 5 - Meterpreter shell 4](#_Toc204621613)

[Figure 6 - Program with unknown SUID set 4](#_Toc204621614)

[Figure 7 - Privilege escalation and root shell 5](#_Toc204621615)

# **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:

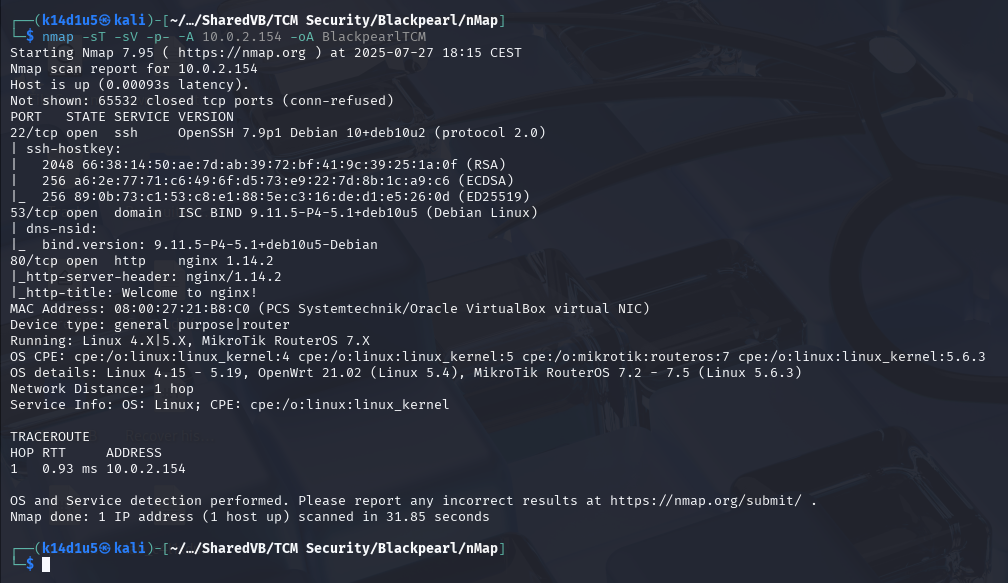


Figure 1 - nMap scan results

Open ports are 22, 53 and 80. Therefore, enabled services are SSH (22) and DNS (53). Also, a web application is running on port 80. Lastly, nMap recognized Linux as operative system.

# **Initial foothold**

First thing I did was analyzing the web application running on port 80. However, I didn’t find anything useful, but an email in the source code of its index page. Therefore, I tried to explore the domain found in the e-mail address. To accomplish this task, I added an entry in my file. At this point, when I browsed to it, I found a new web application. In particular, the index page was the PHPInfo page. Therefore, I analyzed it running ffuf tool and I found the actual application relative to the domain:

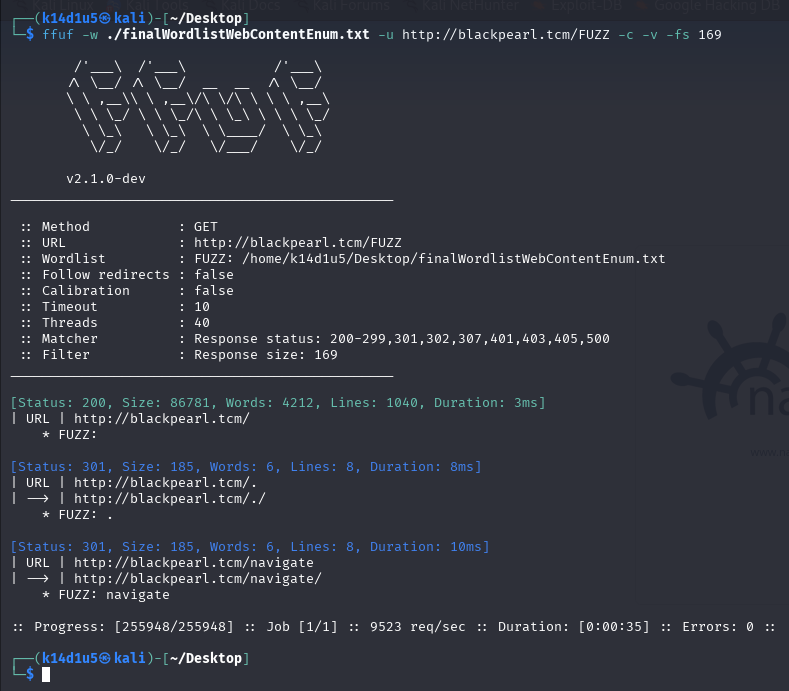


Figure 2 - Path to the web application on the domain

# **User flag**

At this point, I browsed to this web application, and I found out a login page. Also, I saw the application name and version, as shown in the following picture:

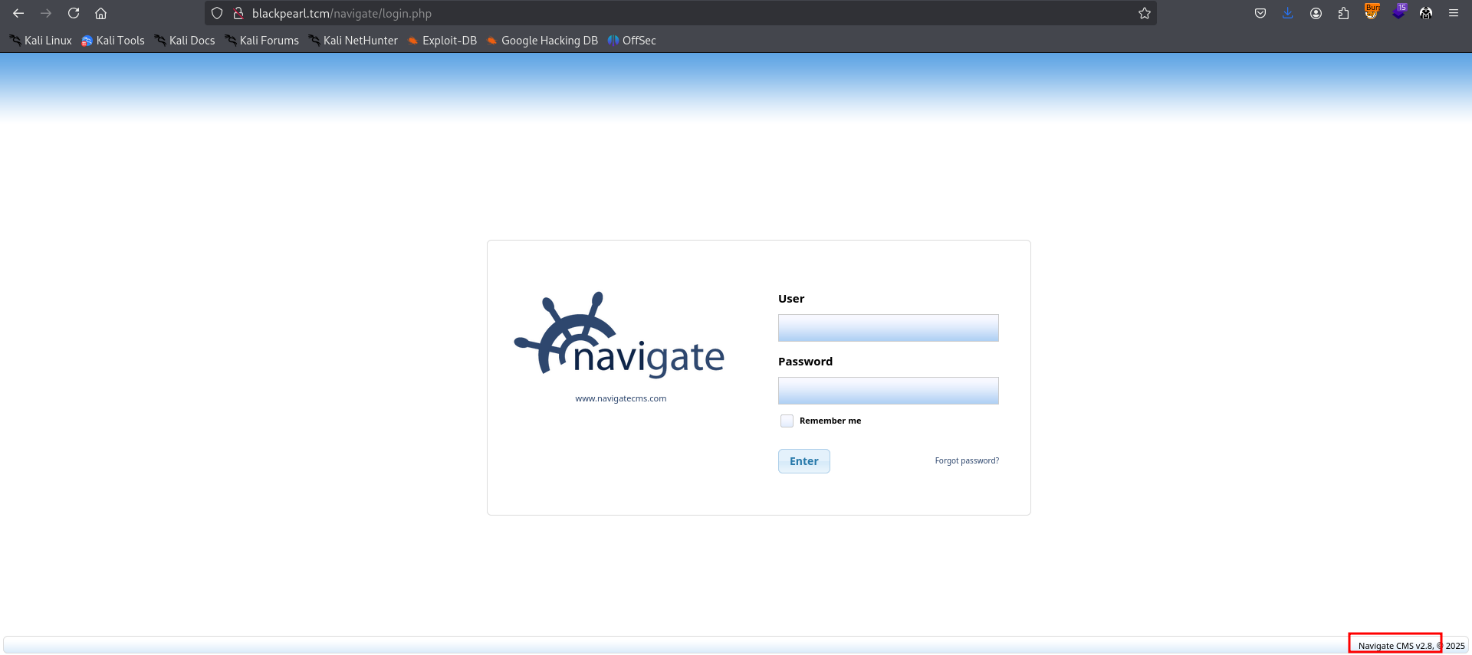


Figure 3 - Application name and version

I looked for an exploit against it and I found one in Metasploit. Therefore, I set up the relative module:

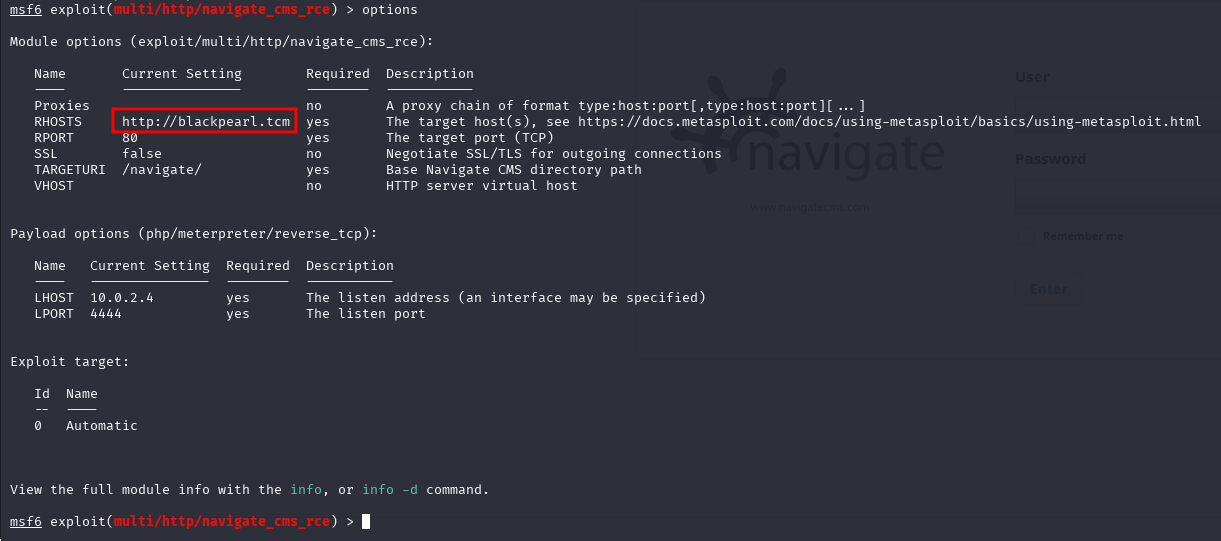


Figure 4 - Metasploit exploit module

Running this module, I obtained a Meterpreter shell as user:

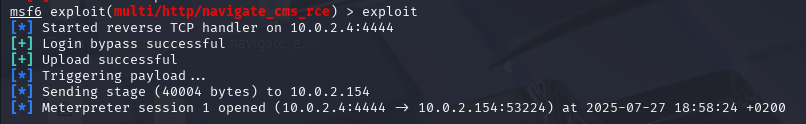


Figure 5 - Meterpreter shell

# **Privilege escalation**

At this point I needed to escalate my privileges. I lost a lot of time for it because I thought I needed to became user before to impersonate the one. Anyway, to escalate my privileges, I run LinPEAS tool. Reading its output, I found out one program with an unknown SUID set:

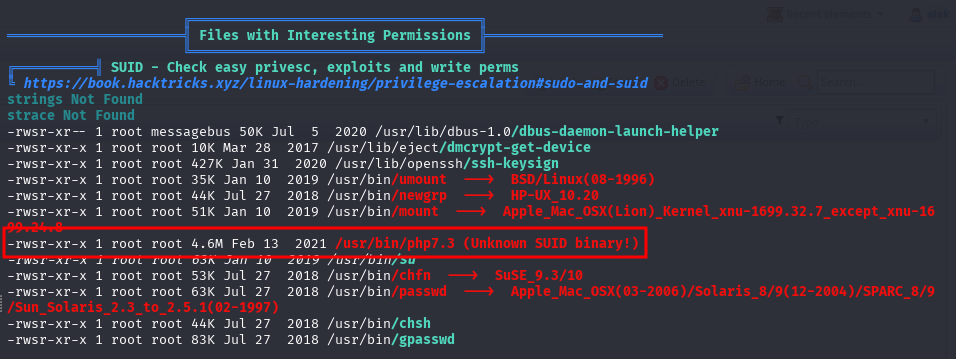


Figure 6 - Program with unknown SUID set

Therefore, I looked on GTFObins website and I found an interesting privilege escalation to run:

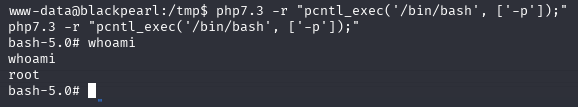


Figure 7 - Privilege escalation and root shell

In this way, I impersonate user.

# **Personal comments**

I just experience a little bit difference to the solution. In fact, I found just one program with the SUID set. Also, the command suddenly didn’t work anymore (but, luckily, I didn’t need it anymore). The actual comment is that it was strange I didn’t need to perform lateral movement to became user. In fact, using the shell as I escalate my privileges directly to the user. Due to this path, I didn’t consider, at first time, some useful information and I spent more time than I needed. It is a very nice box and I evaluate it as an easy one.

# **Appendix A –** **CVE-2018-17552**

This vulnerability affects an unknown code of the file *login.php*. The manipulation as part of a *Cookie* leads to a SQLInjection vulnerability. The product constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component. This is going to have an impact on confidentiality, integrity, and availability. The exploitability is told to be easy. It is possible to launch the attack remotely. The exploitation doesn't require any form of authentication.

# **Appendix B – CVE-2018-17553**

This vulnerability affects an unknown code block of the file *navigate\_upload.php* of the component *File Upload*. The manipulation as part of a *POST Request* leads to an unrestricted upload vulnerability. The product allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment. As an impact it is known to affect integrity, and availability. The exploitation appears to be easy. The attack can be launched remotely. The requirement for exploitation is a single authentication.

# **References**

1. CVE-2018-17552 – <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2018-17552>;
2. CVE-2018-17553 – <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2018-17553>.