**Pit walkthrough**

# **Index**

[Index 1](#_Toc202884716)

[List of pictures 1](#_Toc202884717)

[Disclaimer 2](#_Toc202884718)

[Reconnaissance 2](#_Toc202884719)

[Initial foothold 3](#_Toc202884720)

[User flag 4](#_Toc202884721)

[Privilege escalation 7](#_Toc202884722)

[Personal comments 9](#_Toc202884723)

[Appendix A – More details on privilege escalation method 9](#_Toc202884724)

[Appendix B – CVE-2019-12744 9](#_Toc202884725)

[References 10](#_Toc202884726)

# **List of pictures**

[Figure 1 - nMap scan results (part 1) 2](#_Toc202884727)

[Figure 2 - nMap scan results (part 2) 2](#_Toc202884728)

[Figure 3 - nMap UDP scan results 3](#_Toc202884729)

[Figure 4 - Domain found 3](#_Toc202884730)

[Figure 5 - Application found 4](#_Toc202884731)

[Figure 6 - Userame found 4](#_Toc202884732)

[Figure 7 - Security considerations SeedDMS v. 5.1.15 4](#_Toc202884733)

[Figure 8 - Web shell uploading 5](#_Toc202884734)

[Figure 9 - Webshell invocked 5](#_Toc202884735)

[Figure 10 - Downloading Web application 6](#_Toc202884736)

[Figure 11 - Credentials found 6](#_Toc202884737)

[Figure 12 - User flag 6](#_Toc202884738)

[Figure 13 - Command found 7](#_Toc202884739)

[Figure 14 - Script “monitor” code 7](#_Toc202884740)

[Figure 15 - ACLs on monitoring folder 7](#_Toc202884741)

[Figure 16 - SNMP configuration allow monitoring script execution 8](#_Toc202884742)

[Figure 17 - Privilege escalation exploit 8](#_Toc202884743)

[Figure 18 - Root flag 8](#_Toc202884744)

[Figure 19 - Attempt to write file in /root/.ssh folder 9](#_Toc202884745)

[Figure 20 - Attempt results 9](#_Toc202884746)

# **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:

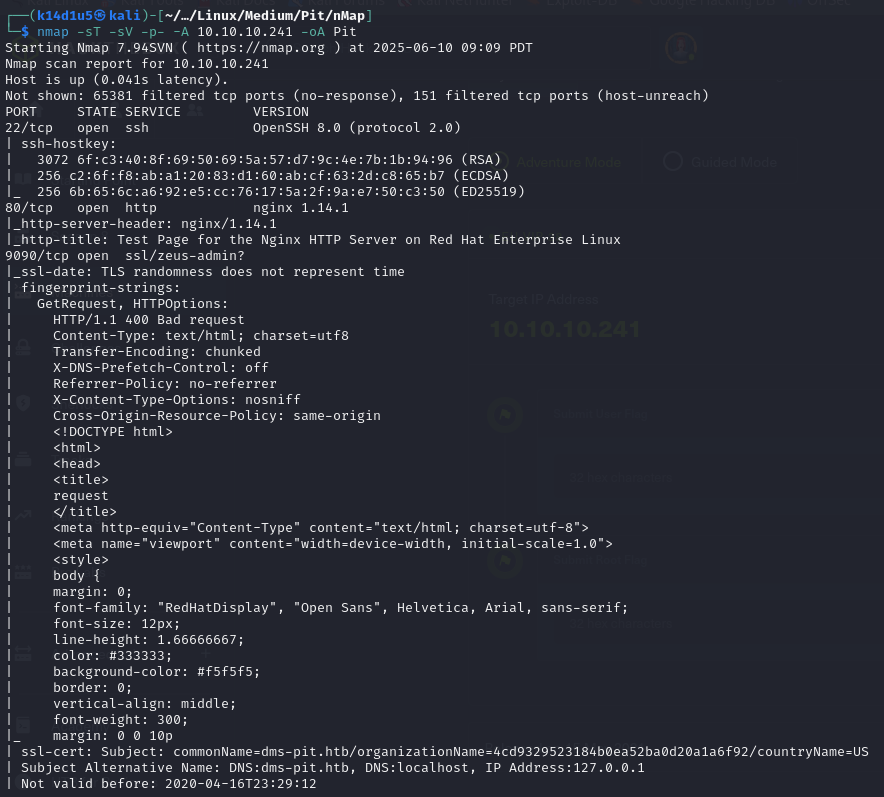


Figure 1 - nMap scan results (part 1)

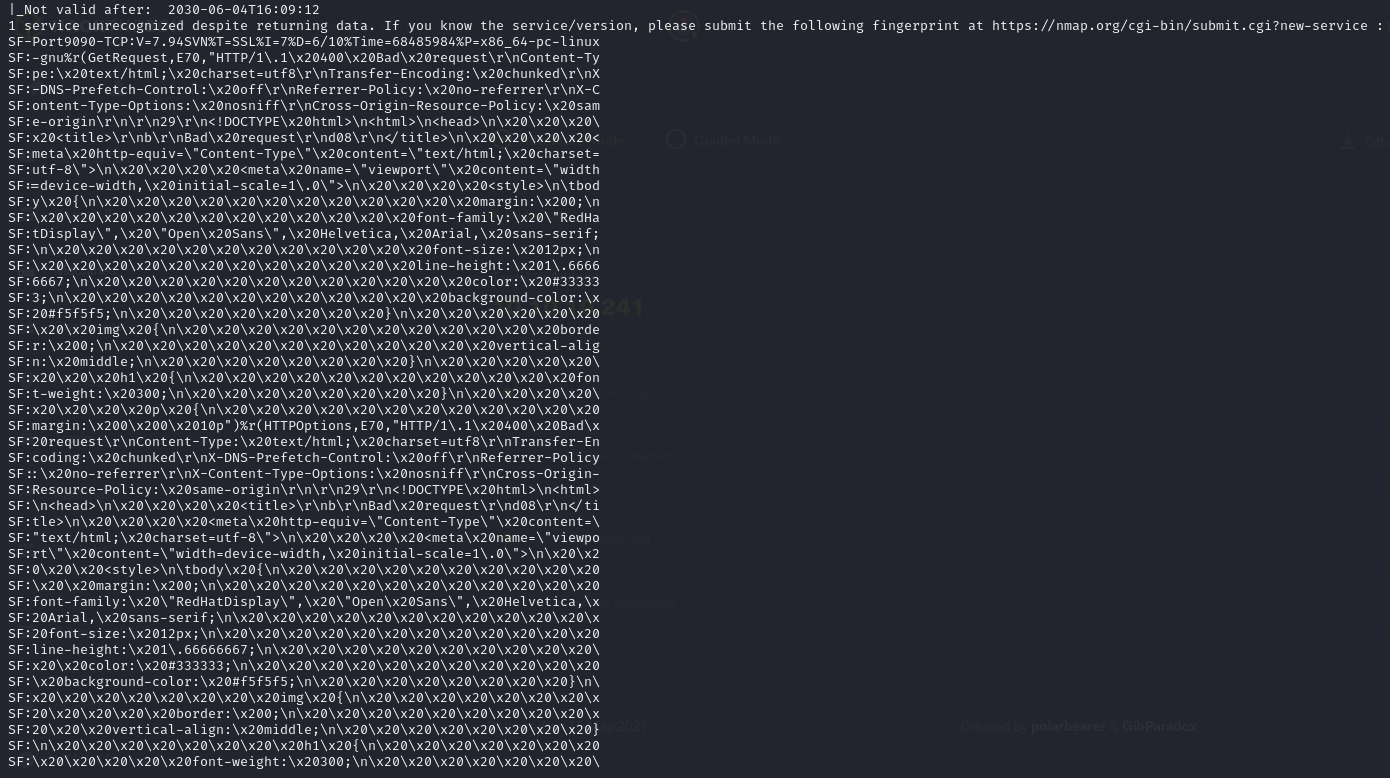


Figure 2 - nMap scan results (part 2)

Open ports are 22, 80 and 9090. Therefore, SSH service is enabled on port 22 and there are two web application on ports 80 and 9090. Lastly, nMap recognized Linux as operative system. In this case, an UDP scan was useful:



Figure 3 - nMap UDP scan results

I scanned the first 1000 ports in UDP and I just found SNMP service on port 161.

# **Initial foothold**

First of all, I browsed to the web applications. The one on port 80 is empty, I just received the Nginx default index page. The one on port 9090 has a login form. Also, I found a new domain in its certificate:

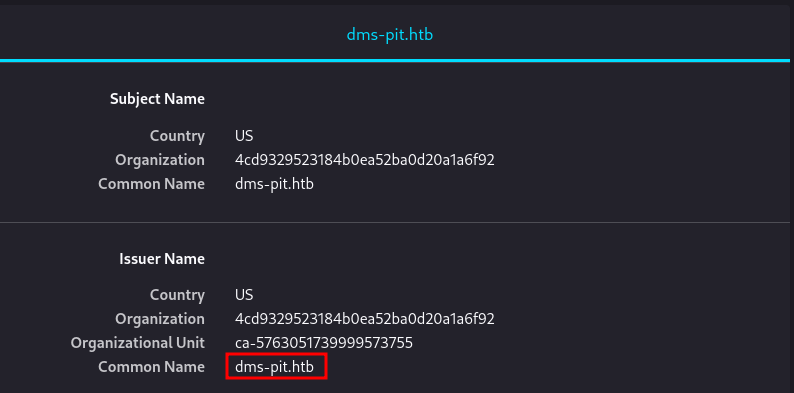


Figure 4 - Domain found

I tried to access to this domain, but I received a forbidden message. At this point I started to analyze the SNMP service. Using SNMPWalk tool, I found a new application:

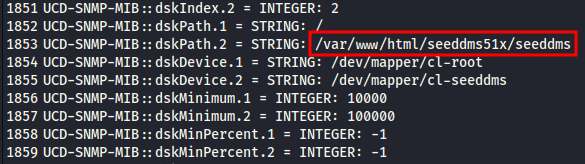


Figure 5 - Application found

Also, I found a plausible username too, as shown in the following figure:

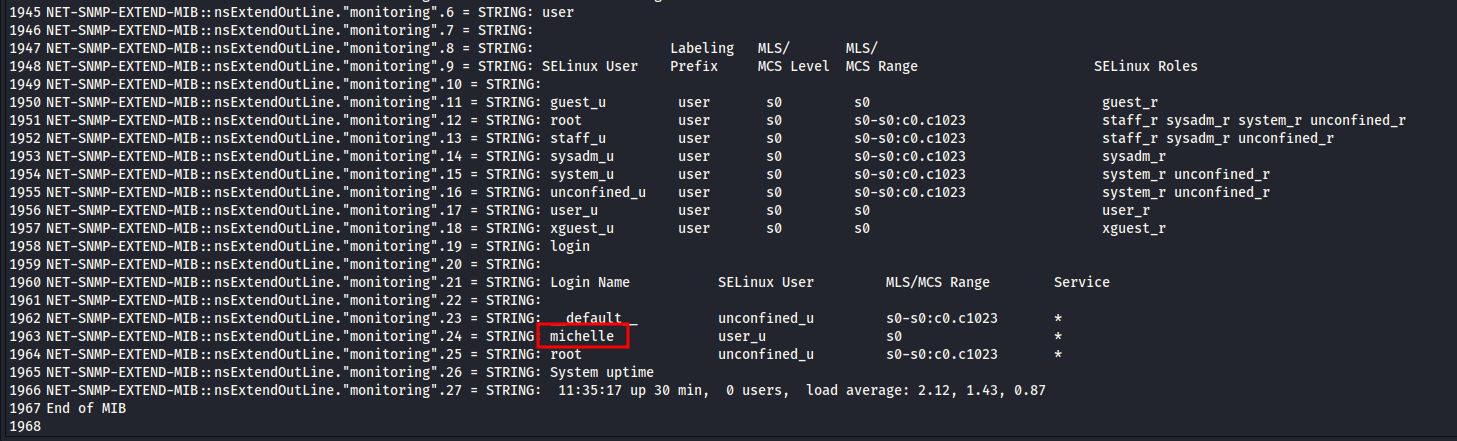


Figure 6 - Username found

# **User flag**

Since I found an application path, I tried to access to it using the URLs I had. I successful accessed to it on <http://dms-pit.htb> URL, where I found a new login form. Also, since I found a plausible username, I tried to use it in both login forms I found. However, I hadn’t a password. Therefore, I tried to use the username as password too. Luckily, I was successful on <http://dms-pit.htb> URL. The application running on this URL was SeedDMS version 5.1.15. I found a changelog file on the portal and it seems to be patched against CVE-2019-12744. However, the SeedDMS 5.1.15 readme file shows that a vulnerability could be still present:

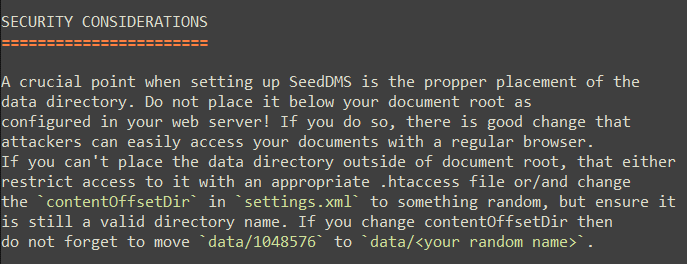


Figure 7 - Security considerations SeedDMS v. 5.1.15

At this point, I looked for some interesting exploit on the Internet. Luckily, I found one and I was able to exploit it. Therefore, I uploaded a webshell using the portal at <http://dms-pit.htb> URL:

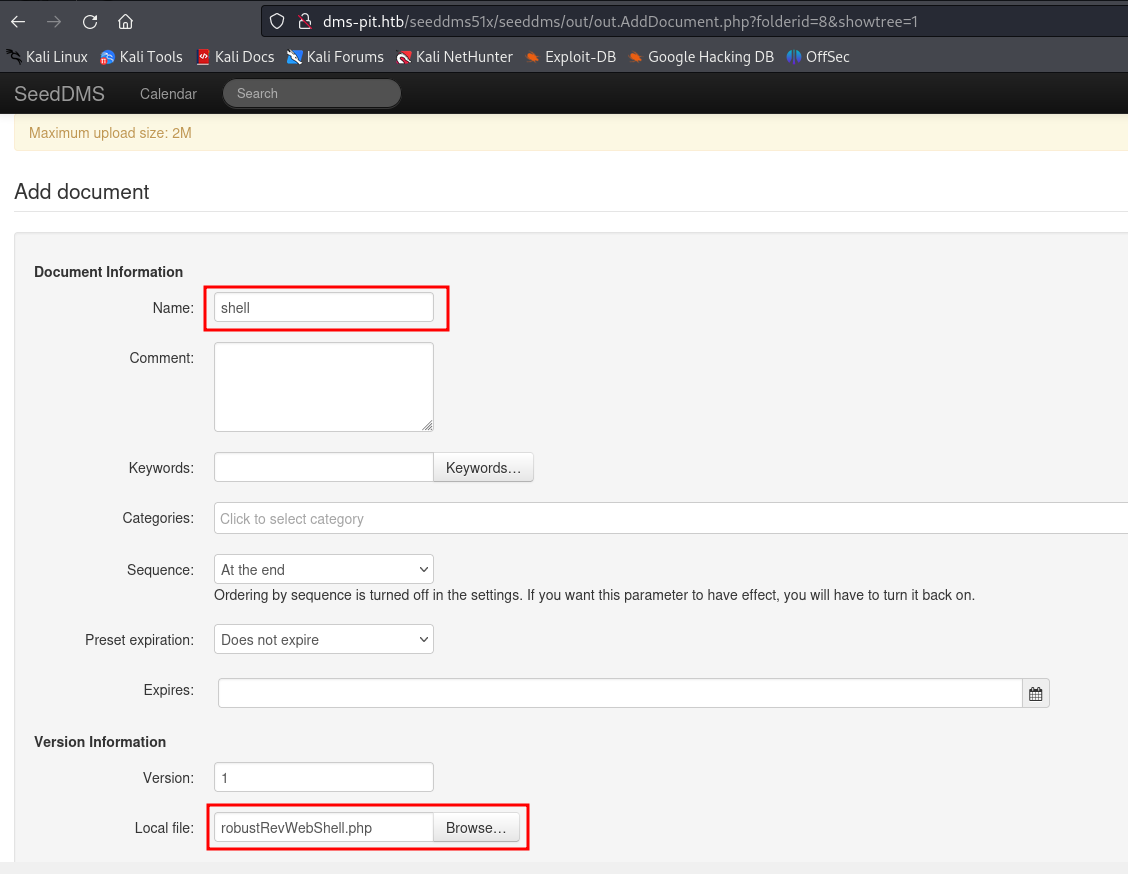


Figure 8 - Web shell uploading

After, I can access to my web shell just uploaded using the <http://dms-pit.htb/seeddms51x/data/1048576/29/1.php?cmd=hostname>, as described by the exploit:

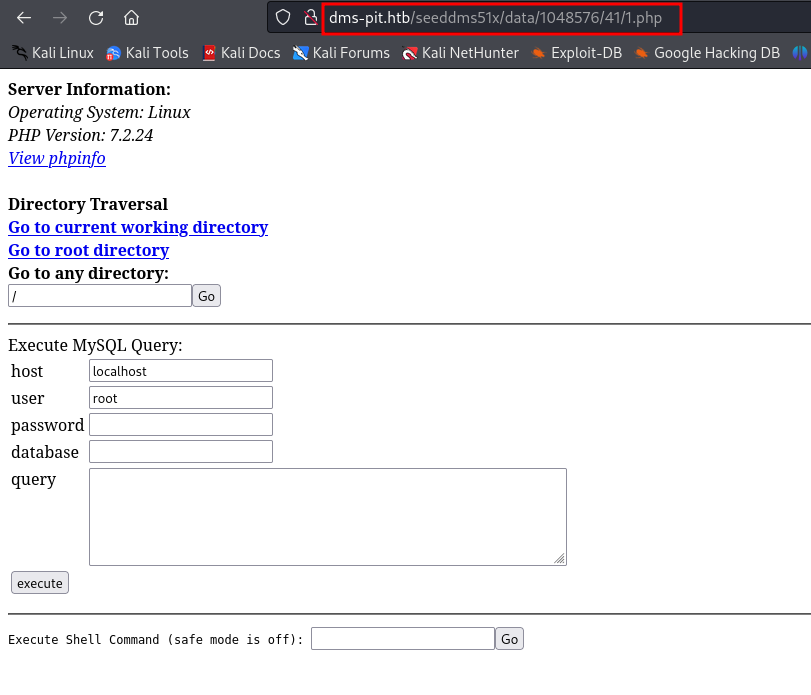


Figure 9 - Webshell invoked

Sadly, the web shell was deleted every few minutes. Therefore, to work without interruption, I decided to open a connection to my Kali and download all application files:

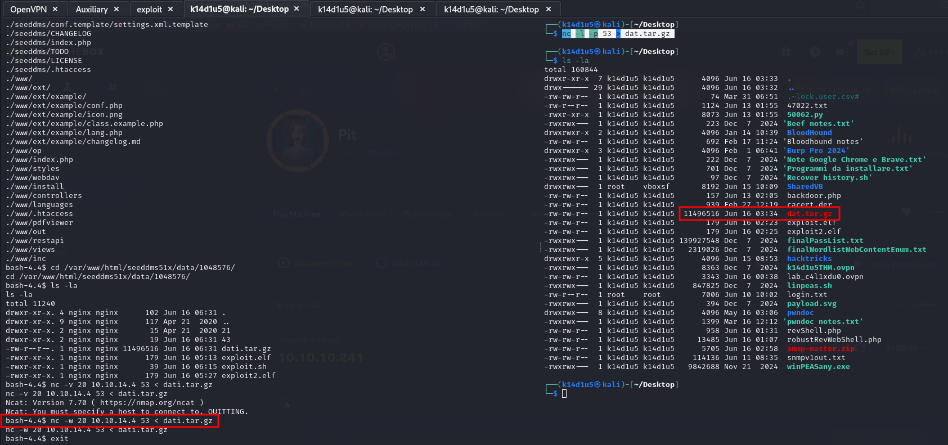


Figure 10 - Downloading Web application

In this way I was able to explore all application files on my local Kali machine. Looking in all these files, I found new credentials:

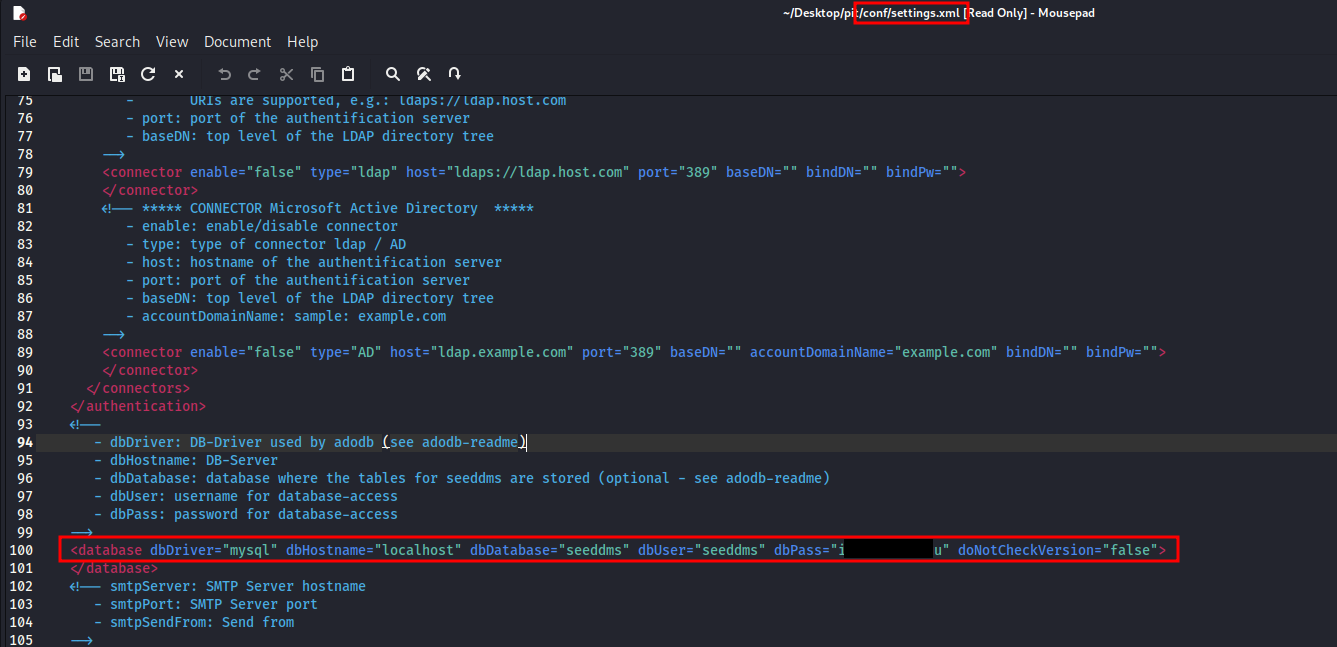


Figure 11 - Credentials found

Finally, the new password found was useful to login as user on the web application running on port 9090 and I was able to retrieve the user flag:

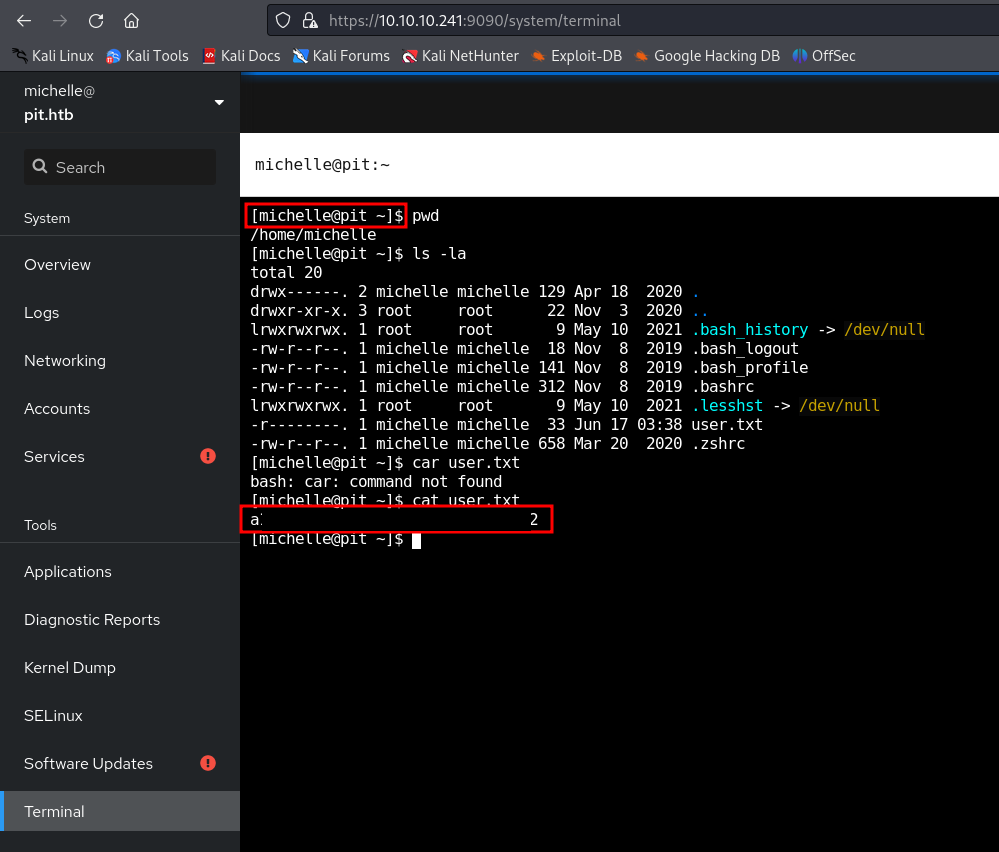


Figure 12 - User flag

# **Privilege escalation**

Since I wanted to work on my local Kali machine, the first thing I did was create SSH key for user, so I was able to connect via SSH to the target. At this point I was looking for a way to escalate my privileges. It was a very though task, because I didn’t find anything using the common techniques. After a while, I checked again the full SNMPWalk output. Just in that moment finally I found something useful. In that results I found a script name, set in an extended command field:

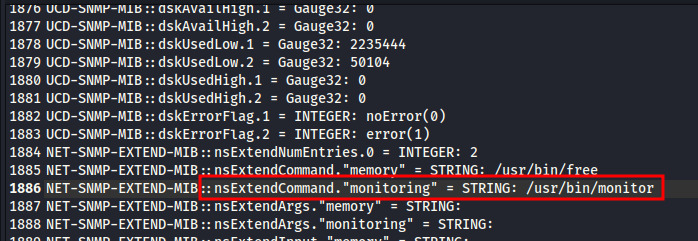


Figure 13 - Command found

At first glance, I thought it was an executable file. But when I checked it, I found out that it was a script:

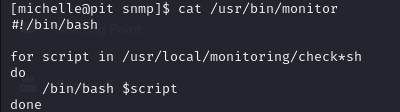


Figure 14 - Script “monitor” code

This script executes all scripts contained in and that its name begins with and ends with . All I needed to do was create a proper script I the right folder. However, I was not able to read that path. Luckily, I was able to write in it because of the specific limited ACL rules set:

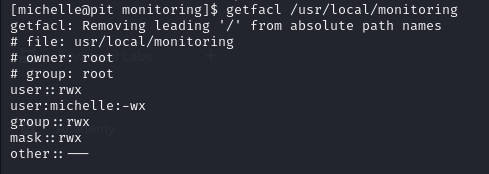


Figure 15 - ACLs on monitoring folder

At this point, I needed to understand how to run my malicious script. I tried to create a reverse shell script and waited for its automatic execution. I hoped that monitor script was periodically invoked, but my assumption was wrong. After a while, I did research on the Internet and I found out I can run scripts via SNMP. I had confirmation about it analyzing the SNMPWalk output:

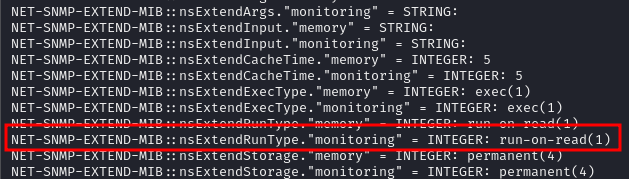


Figure 16 - SNMP configuration allow monitoring script execution

To do it, I just needed to run an SNMP enumeration to run the monitor script again. However, I was not able to open a reverse shell. It was due to SELinux. At this point, I thought to use again the same technique I used before and install SSH key for the user. Therefore, I created SSH key on my local Kali machine and I developed a script to download it and save it in the folder. At this point, I just needed to run again an SNMP enumeration to run the monitor script and, consequently, my malicious script:

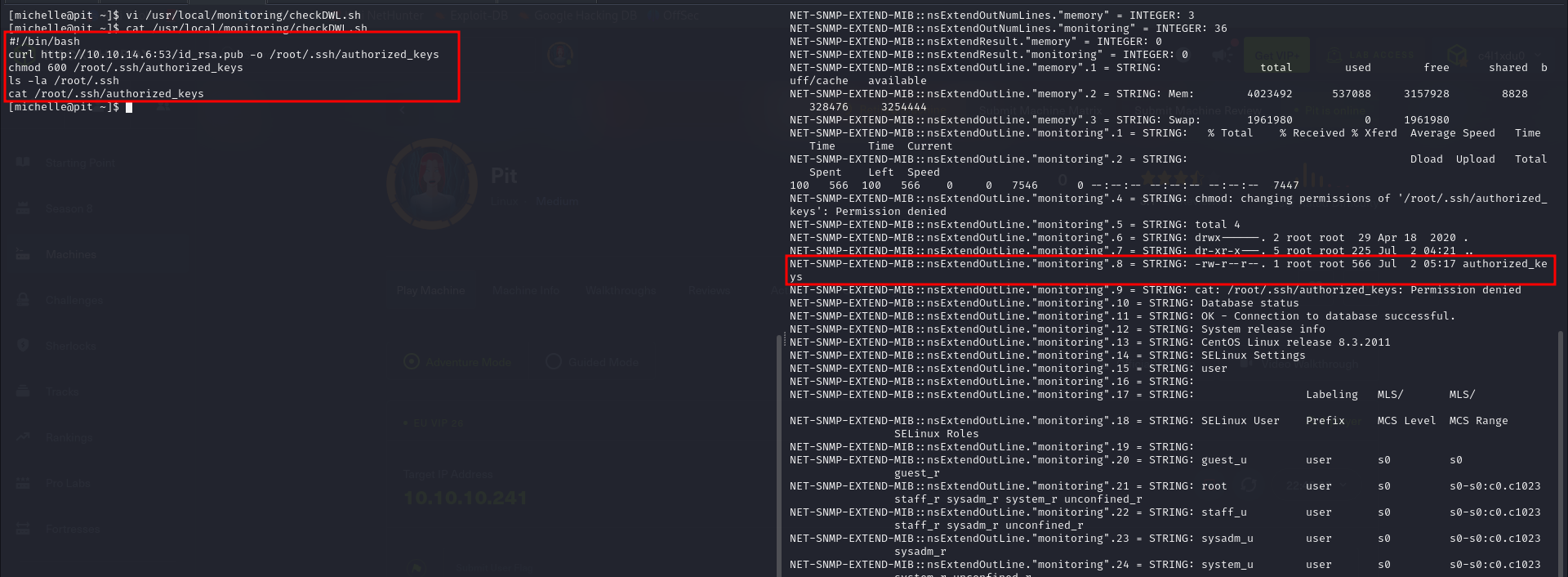


Figure 17 - Privilege escalation exploit

Finally, I can use the SSH key to access as user on the target and retrieve the root flag:

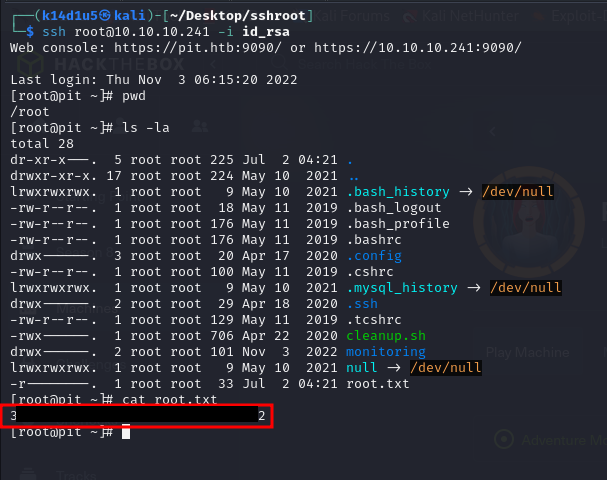


Figure 18 - Root flag

# **Personal comments**

This box was very challenging for me due to at least three reasons. The first one is that you need to thoroughly analyze and explore software documentation even if you find notes that issues are fixed. In this case, anything makes you sure a plausible exploit works, but it is always better to try. In this case, it worked. The second reason is that the backdoor you upload on <http://dms-pit.htb/> portal will die after few minutes. It is very frustrating and don’t let you to work steadily. Last reason is that it was my first box which use again something that was useful before. In this case, I needed to go deeper in SNMP output in different points of progress. Usually, each thing let you to get some information in a specific phase of progress and after that it is not useful again. In conclusion, this box is very interesting and allowed me to learn several things. I liked it, but in my opinion is a little bit harder than classic medium difficulty.

# **Appendix A – More details on privilege escalation method**

When I tried to escalate my privileges, I tried to write a different file in the folder suing a custom script as I did in the walkthrough:

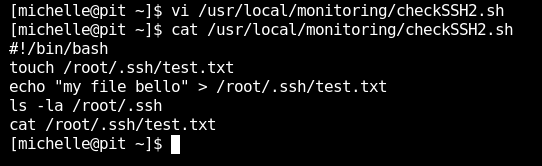


Figure 19 - Attempt to write file in /root/.ssh folder

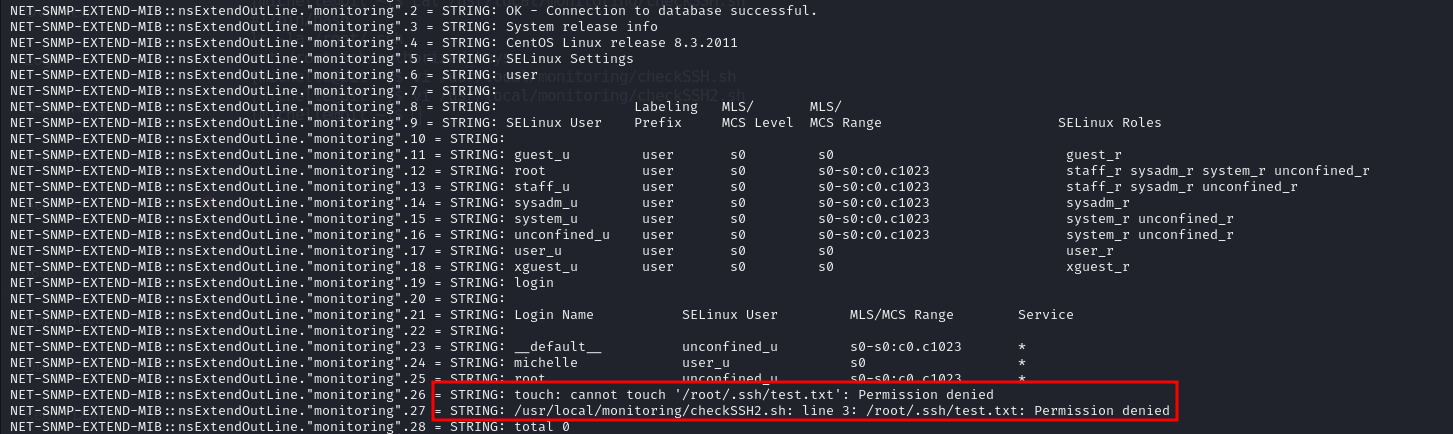


Figure - Attempt results

However, I was unsuccessful and I am not sure why. In fact, the malicious script was executed by root. Therefore, it must be able to write files in that folder, in my opinion. Since this was the behavior, why the script was able to write the SSH key? As I said, I have not the specific answer, but I noted that the file already exists in that folder. This means that the script didn’t need to create a new file (task that failed as I showed), but just write it (overriding it). I am very surprised, but it worked. I hope this appendix could be useful for you in same way.

# **Appendix B – CVE-2019-12744**

Some unknown functionalities of component are affected by CVE-2019-12744. The manipulation with an unknown input leads to a command injection vulnerability. The attack can be remotely launched. This vulnerability has been declared as critical. The product constructs all or part of a command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended command when it is sent to a downstream component. As an impact it is known to affect confidentiality, integrity, and availability.

# **References**

1. CVE-2019-12744: <https://cve.mitre.org/cgi-bin/cvename.cgi?name=2019-12744>;
2. CVE-2019-12744 exploit guide: <https://bryanleong98.medium.com/cve-2019-12744-remote-command-execution-through-unvalidated-file-upload-in-seeddms-versions-5-1-1-5c32d90fda28>.