Devvortex walkthrough

Index

List of pictures

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:

```
# Nmap 7.94SVN scan initiated Fri Jan  5 15:17:39 2024 as: nmap -sT -p- -sV -sC -O -A -oA Devvortex 10.10.11.242
Nmap scan report for 10.10.11.242
Host is up (0.023s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://devvortex.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=1/5%OT=22%CT=1%CU=35172%PV=Y%DS=2%DC=T%G=Y%TM=65978
OS:316%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=106%TI=Z%CI=Z%II=I%TS=A)S
OS:EQ(SP=106%GCD=1%ISR=106%TI=Z%CI=Z%II=I%TS=A)SEQ(SP=106%GCD=1%ISR=107%TI=
OS:Z%CI=Z%II=I%TS=A)OPS(O1=M53CST11NW7%O2=M53CST11NW7%O3=M53CNNT11NW7%O4=M5
OS:3CST11NW7%O5=M53CST11NW7%O6=M53CST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88
OS:%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M53CNNSNW7%CC=Y%Q=)T1(R=Y%DF
OS:=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z
OS:%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=
OS:Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%
OS:RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
OS:IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   25.65 ms 10.10.14.1
2   20.62 ms 10.10.11.242

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jan  5 15:18:30 2024 -- 1 IP address (1 host up) scanned in 50.92 seconds
```

Picture 1 - nMap scan results

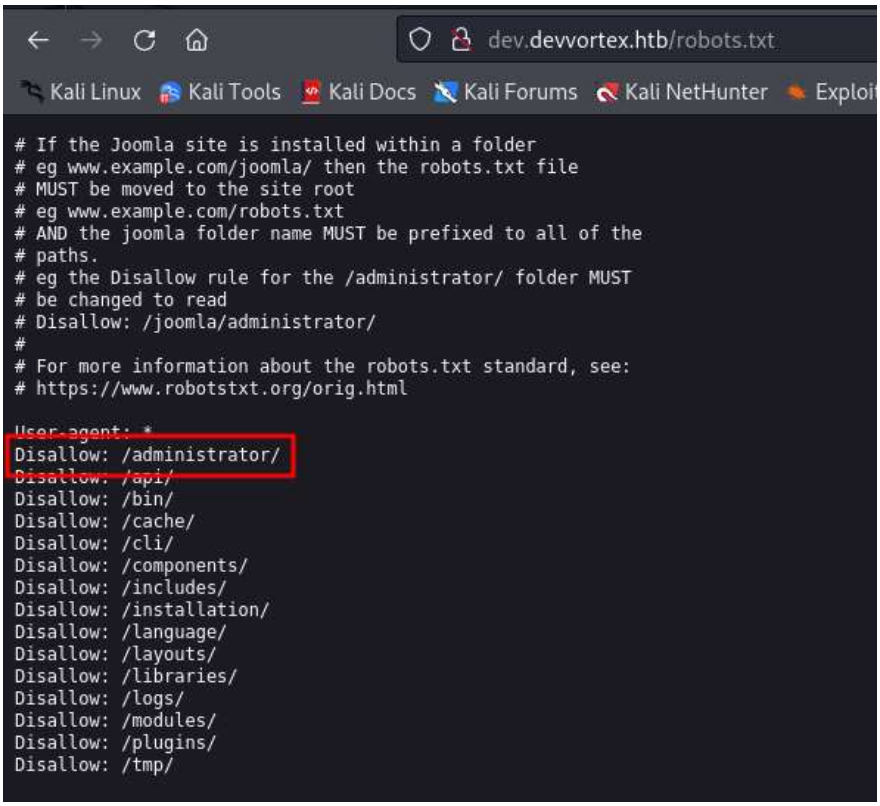
Open ports are 22 and 80. So, the machine has SSH enabled and an application running on port 80. NMap detected that operative system is Linux, but didn't provide any other specific information about it.

Initial foothold

One important step to follow while analyzing a web application, is the subdomain enumeration. In this case, I was able to find a new subdomain as shown in the following picture:

Picture 2 - Subdomain enumeration

So, I started to analyze this new subdomain. Here, ***robot.txt*** is accessible and it provide some useful information. In fact, I found an administrative path:



Picture 3 - Robots.txt file on dev.devvortex.htb

I tried to access to that path and I found a Joomla administrator login form. I looked for some known vulnerabilities on the Internet, and I found **CVE-2023-23752**. This CV is about an ***Improper Access Execution*** vulnerability in the ***/api/index.php/v1/config/application***, ***/joomla/api/v1/config/application?public=true***, ***/api/index.php/v1/config/application?public=true***, ***/api/v1/config/application?public=true*** endpoints of the Joomla server. The public parameter of the

vulnerable endpoint allows an attacker to access the Joomla-related configuration information which eventually leads to the disclosure of sensitive information such as database username and password.

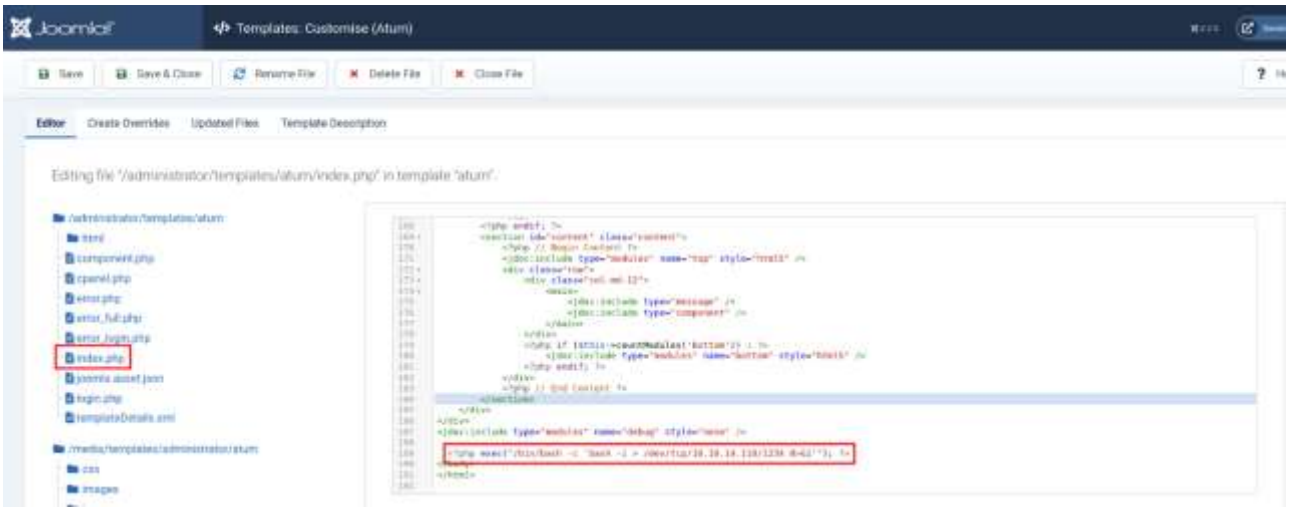
User flag

Since I found an interesting CVE about Joomla, I tried to run its exploit. It works and it provides me the following information:

```
┌──(k14d1u5㉿k14d1u5-kali)-[~/…/Per punti HTB/Linux/Easy/Devvortex]
└─$ ruby ./exploit.rb http://dev.devvortex.htb
Users
[649] lewis (lewis) - lewis@devvortex.htb - Super Users
[650] logan paul (logan) - logan@devvortex.htb - Registered

Site info
Site name: Development
Editor: tinymce
Captcha: 0
Access: 1
Debug status: false

Database info
DB type: mysqli
DB host: localhost
DB user: l[redacted]
DB password: P[redacted]#
DB name: joomla
DB prefix: sd4fg_
DB encryption 0

┌──(k14d1u5㉿k14d1u5-kali)-[~/…/Per punti HTB/Linux/Easy/Devvortex]
└─$
```

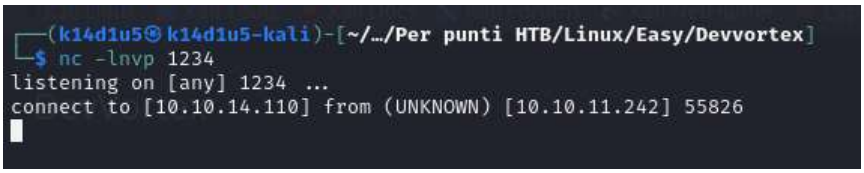
Picture 4 - Databases credentials

I had some credentials at this point, so I tried to use them in the Joomla Administrator login form and they work. In this administrative panel, after a deep inspection, it was possible to modify administrative templates. So, I modified the ***index.php*** page of ***http://devvortex.htb*** to set up a reverse shell:



Picture 5 - Reverse shell command in index.php template page

At this point, I had to set a listener to receive the reverse shell and I needed to reload the index page on ***http://devvortex.htb***. In this way, I obtained a reverse shell with ***www-data*** user:

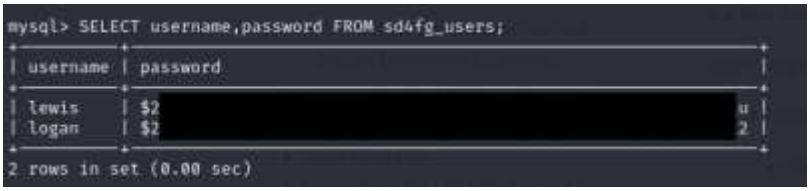


Picture 6 - Reverse shell successful connection

This shell was not good enough to work, so I needed to upgrade and stabilize it with the following steps:

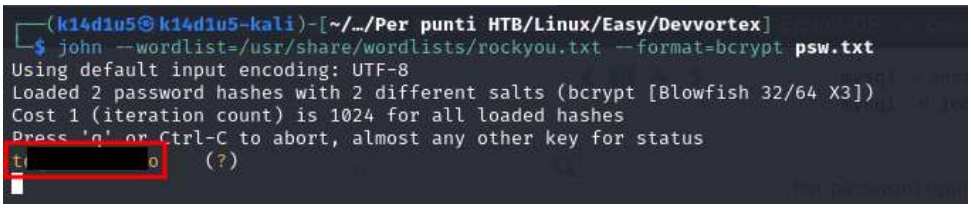
```
script /dev/null -c /bin/bash
CTRL + Z
stty raw -echo; fg
Then press Enter twice, and then enter:
export TERM=xterm
```

At this point, I remembered I found databases credential, so I tried to connect with it. It worked and I inspected the database. At the end, I found some user credentials:



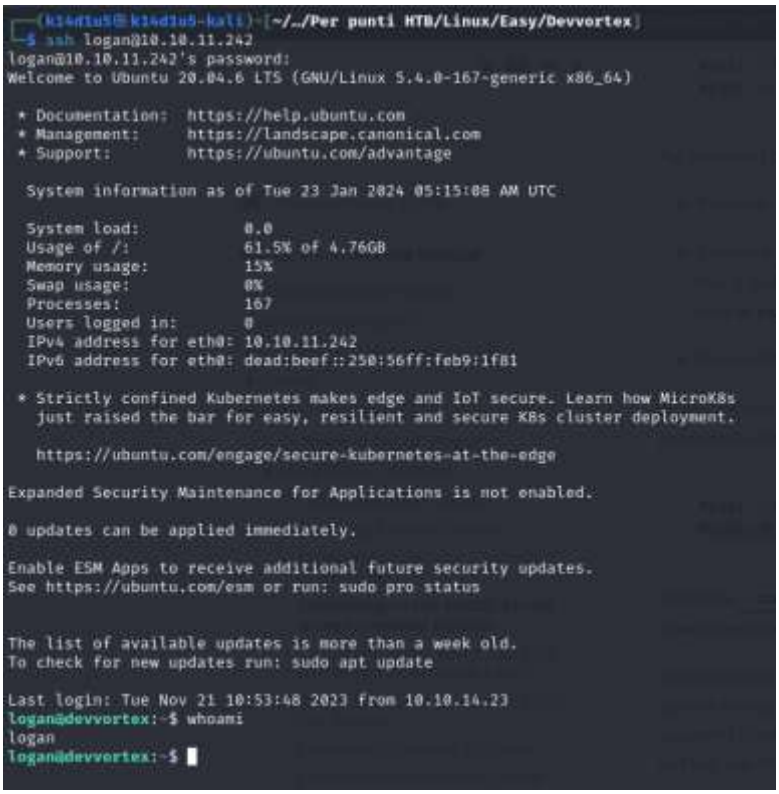
Picture 7 - User credentials in database

I tried to crack this password with ***JohnTheRipper*** tool:



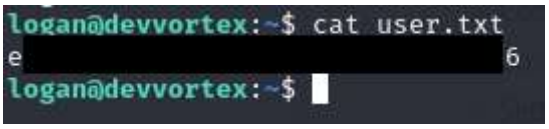
Picture 8 - Password cracked

Luckily, tool cracked one password, the one related to ***logan*** user. So, I connect in SSH with these new credentials:



Picture 9 - SSH connection as logan

At this point, I was able to retrieve the user flag:



Picture 10 - User flag

Privilege escalation

In this case, to escalate my privilege, I saw that ***logan*** user was able to run ***/usr/bin/apport-cli*** as ***sudo***:

Picture 11 - Information useful to escalate privileges

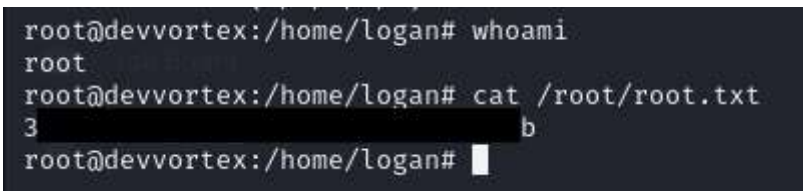
I run this command and it let me to build a report in a text editor similar to ***vim***:

Picture 12 - /usr/bin/apport-cli command execution

So, I re-run this command as ***sudo*** and I exploit it to obtain a root shell. To achieve this goal, I sent a command inside the vim-like text editor opened by the command, in the same way I could in vim. In particular, I sent the command $!/bin/bash$:

Picture 13 - Privilege escalation

At this point, I had just to retrieve the root flag:



Picture 14 - Root flag