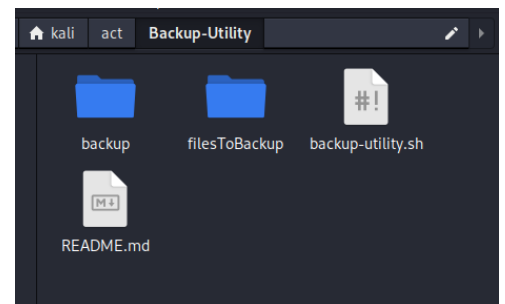
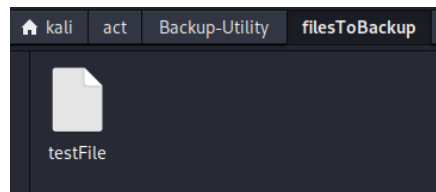
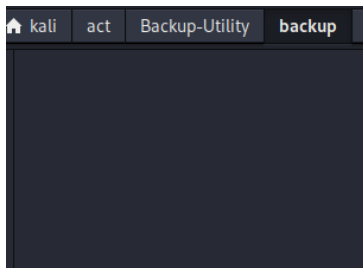


Procedimiento:

Para empezar se prueba el funcionamiento del script haciendo una copia entre carpetas del mismo sistema. Para ello, se crean dos carpetas de prueba, necesarias porque el script no las crea de forma automática. También se crea un archivo vacío de prueba para hacer el respaldo.



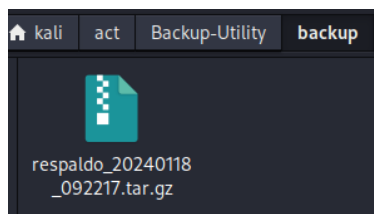
Dentro del script se cambian las variables origen y destino para que correspondan a las carpetas ya creadas.

```
# Directorio de origen que deseas respaldar
origen="./filesToBackup"

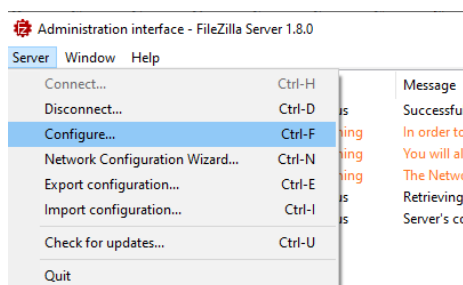
# Directorio de destino donde se almacenarán las copias de seguridad
destino="./backup"
```

Después se ejecuta el script y se ve que funciona correctamente, habiendo creado el archivo comprimido correspondiente.

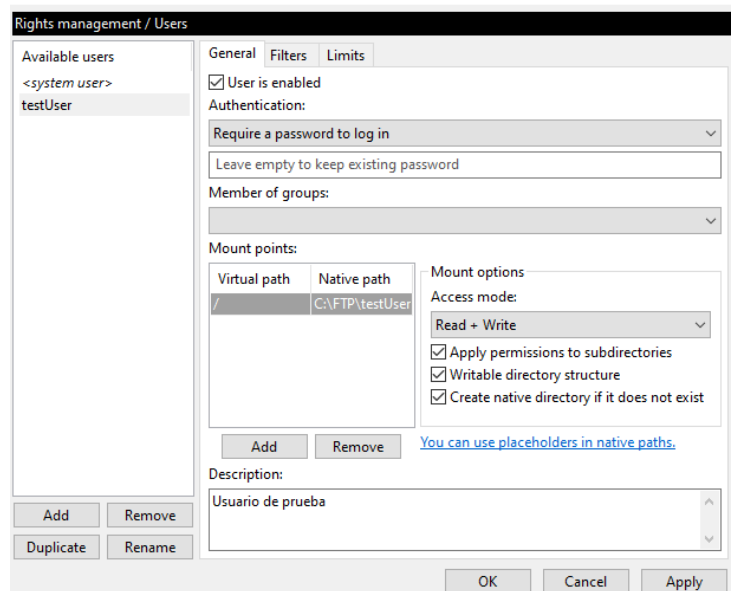
```
(kali@kali)-[~/act/Backup-Utility]
$ bash backup-utility.sh
./filesToBackup/
./filesToBackup/testFile
Copia de seguridad exitosa en ./backup/respaldo_20240118_092217.tar.gz
(kali@kali)-[~/act/Backup-Utility]
$
```



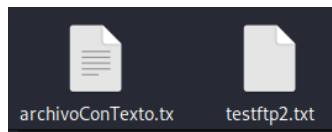
A continuación se procede a buscar un servidor donde guardar el backup, en este caso se va a usar un servidor FTP con FileZilla. Luego de instalar el software correspondiente, se abre la ventana de configuración del servidor para configurar usuarios y la dirección IP. Se crea un usuario de prueba, se le da una contraseña y se le asigna permisos de lectura y escritura. Después se configura la IP del servidor, la dirección a la cual deberán conectarse los equipos que quieran acceder, se recomienda que sea estática con el fin de evitar tener que configurarlo constantemente. El puerto es el 21 (puerto de control de FTP)



Server listeners		
Address	Port	Protocol
0.0.0.0	21	Explicit FTP over TLS a
::	21	Explicit FTP over TLS a
192.168.0.200	21	Explicit FTP over TLS a



Una vez hecho esto, probamos con éxito la conexión al servidor con nuestro usuario de prueba y a la IP configurada anteriormente. Luego, creamos varios archivos para testear si se suben correctamente.



```
(kali@kali)-[~]
$ ftp 192.168.0.200
Connected to 192.168.0.200.
220-FileZilla Server 1.8.0
220-Please visit https://filezilla-project.org/
220 Conectado!
Name (192.168.0.200:kali): testUser
331 Please, specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Se usaron dos comandos distintos:

- **put <nombre archivo a subir>** (requiere que ya se esté logeado en el servidor). Es un comando propio de ftp.

```
ftp> put testftp2.txt
local: testftp2.txt remote: testftp2.txt
229 Entering Extended Passive Mode (|||51891|)
150 Starting data transfer.
0 0.00 KiB/s
226 Operation successful
ftp>
```

- **curl -T --user <usuario>:<contraseña> <url>** (no requiere un logeo previo) El parámetro -T indica que se va a subir un archivo. Es un comando de bash.

```
(kali@kali)-[~]
$ curl -T testftp2.txt --user testUser:mostsecurepassword ftp://192.168.0.200
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
```

Habiendo probado que se pueden subir archivos correctamente, modificamos el script para que suba el backup ni bien se crea.

```
# Nombre del archivo de copia de seguridad con marca de tiempo
archivo_destino="$destino/respaldo_$(date +%Y%m%d_%H%M%S).tar.gz"

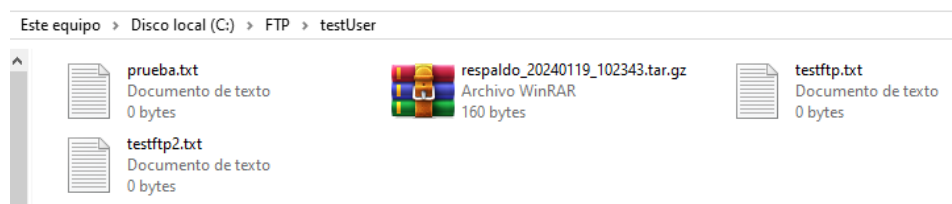
# Comprimir y copiar los archivos de origen al directorio de destino
tar -czvf "$archivo_destino" "$origen"

curl -T $archivo_destino --user testUser:mostsecurepassword ftp://192.168.0.200
```

Y ejecutamos el script.

```
(kali@kali)-[~/act/Backup-Utility]
$ bash backup-utility.sh
./filesToBackup/
./filesToBackup/testFile
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 160 0 0 100 160 0 1438 --:--:-- --:--:-- --:--:-- 1441
Copia de seguridad exitosa en ./backup/respaldo_20240119_102343.tar.gz
```

Ahí vemos que se subió correctamente el archivo del backup.



Podemos crear una tarea programada si escribimos el comando **crontab -e** y dentro ponemos cada cuánto se debería ejecutar y el comando a ejecutar. Para cambiar el editor de texto ponemos el comando **select-editor**.

```
# For more information see the manual pages of crontab(5) and
#
# m h dom mon dow command
* * * * * /home/kali/act/Backup-Utility/backup-utility.sh
```