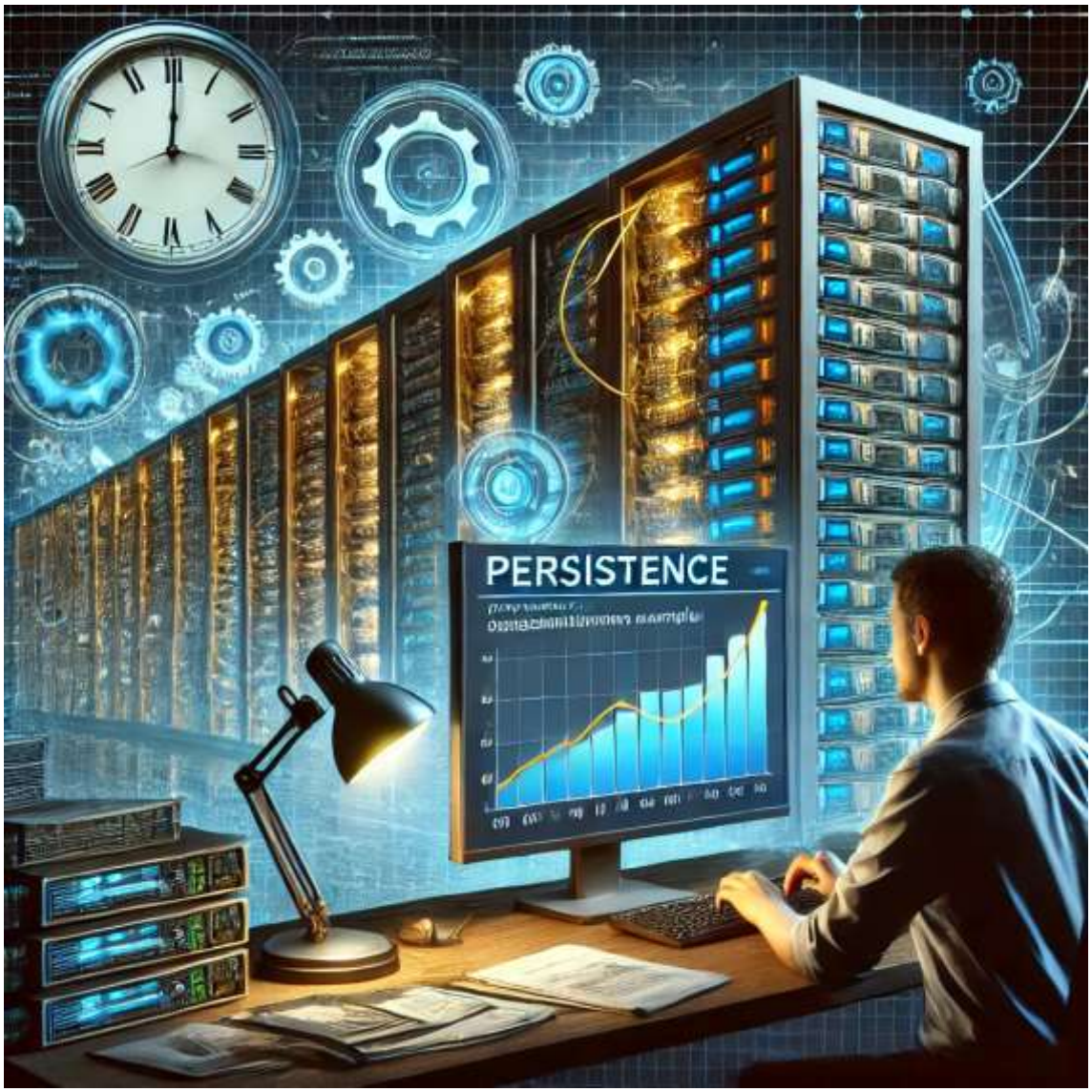
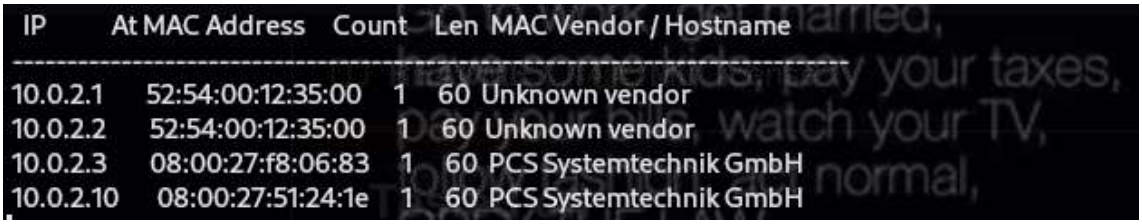


EJERCICIO UNO – SPRINT 12

PERSISTENCIA - LINUX

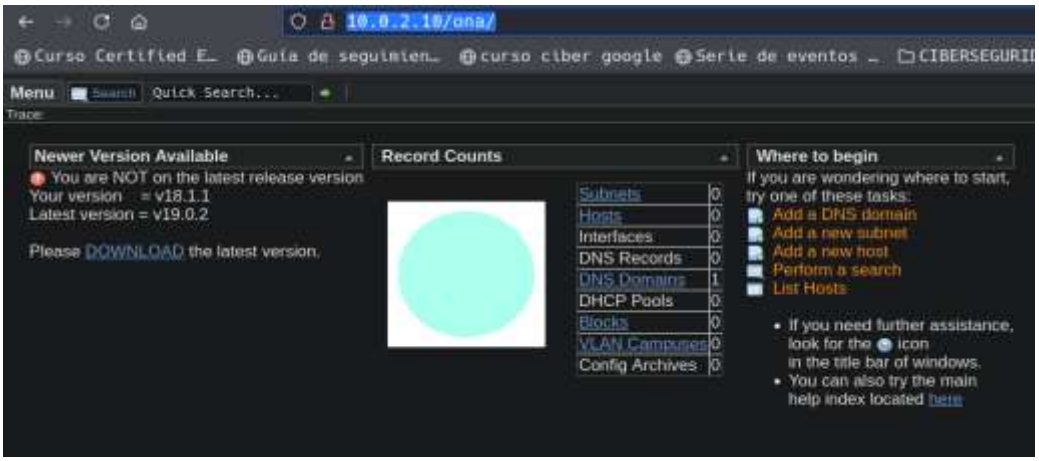
En el presente ejercicio se ha realizado la post-explotacion sobre la maquina Linux “Five86_1” aportada por el equipo docente durante una review, realizando las gestiones necesarias para conseguir persistencia en el sistema, siendo estas:

1. La IP de la maquina objetivo es la terminada en .10:



2. Se realiza consulta mediante Gobuster a los directorios del sistema con el diccionario *“rockyou”,* obteniendo muchos errores “*parse”* y falsos positivos, siendo el único interesante:

3. A través de la web http://10.0.2.10/ona/ encontramos un administrador de redes, dns, hosts etc, y tras una búsqueda, se comprueba que *“ona”* es el acrónimo de *“OpenNetAdmin*”, el cual es una aplicación de gestión de redes basada en web que permite a los administradores de sistemas y redes gestionar y documentar la infraestructura de red de una organización, simplificando su administración y documentación, especialmente en entornos complejos.



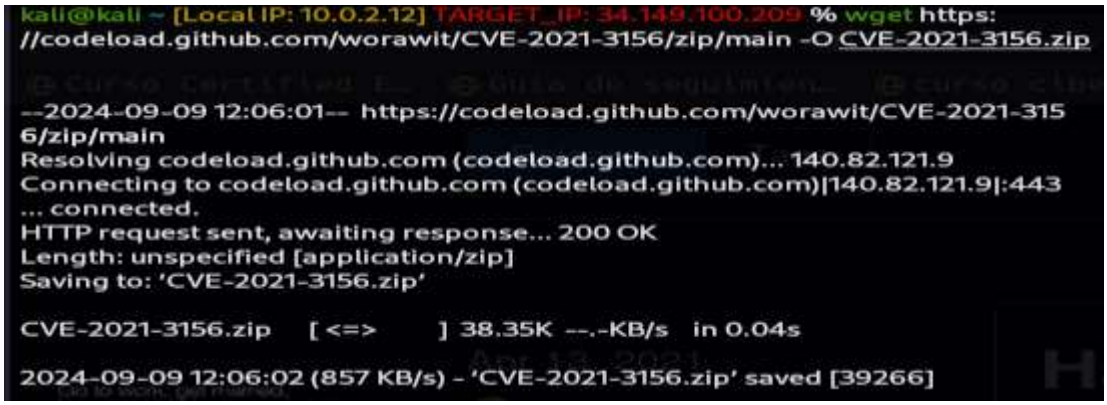
4. Mediante el uso de Metaexploit se procede a la búsqueda de alguna vulnerabilidad a este servicio que sea explotable siendo positiva, consiguiendo acceso al sistema mediante un exploit que consigue cargar un payload que me permite abrir una shell interactiva tipo *“meterpreter”:*

5. El usuario que permite interactuar en la shell tiene privilegios básicos (www-data y douglas) por lo que, para poder conseguir la persistencia, es necesario aumentar estos privilegios, habiendo usado una herramienta de ayuda para conseguir escalada de privilegios, llamada *linPEAS (Linux Privilege Escalation Awesome Scripts)*.

Una vez conseguido el archivo, se procede a subir el archivo a través de la meterpreter abierta en el sistema atacado mediante el comando *“upload”* junto al script obtenido *“limpeas.sh”* y el destino en la maquina objetivo*: /tmp/,* debido a los permisos restringidos de los usuarios.

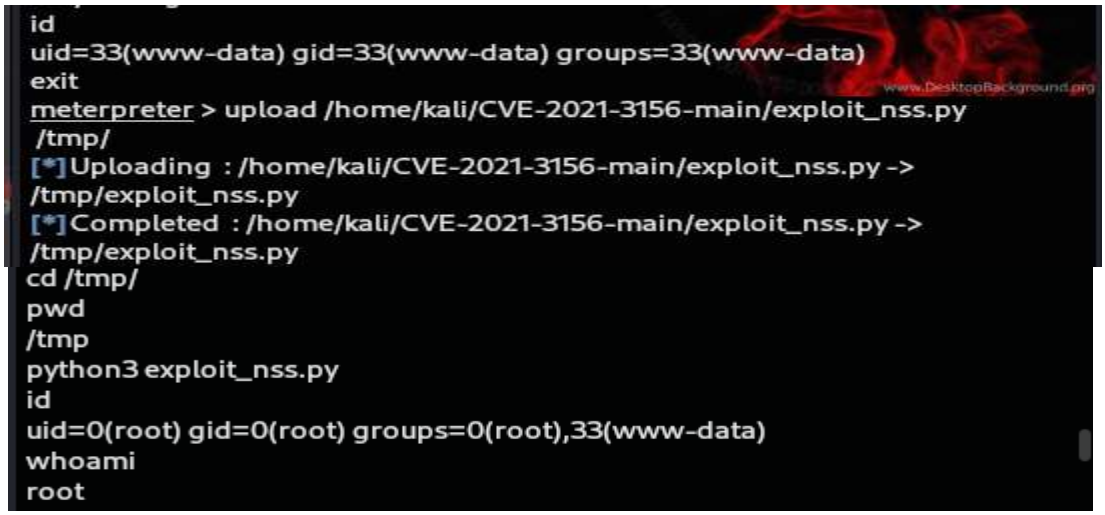
6. Ahora ejecutamos el script dentro de la máquina atacada, para que empiece el análisis, aportando numerosa información, destacando:

7. En el punto anterior podemos observar el sistema operativo completo del sistema, así como la versión sudo instalada en el mismo (1.8.27), aportando dos versiones de una vulnerabilidad que es aplicable a las versiones sudo anteriores a la 1.9.5p2, por lo que se procede a intentar explotar esta vulnerabilidad:

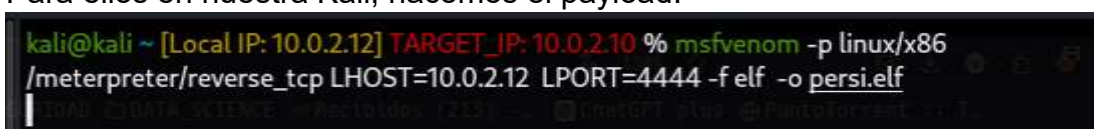


Una vez descargado y descomprimido en nuestra Kali, se puede observar en su interior varios exploit de python según la versión del sistema donde se quiere ejecutar, usando en este caso el genérico: *exploit_nss.py.*

Se procede a subir el archivo a través de meterpreter el sistema objetivo y a su ejecución, consiguiendo privilegios root:



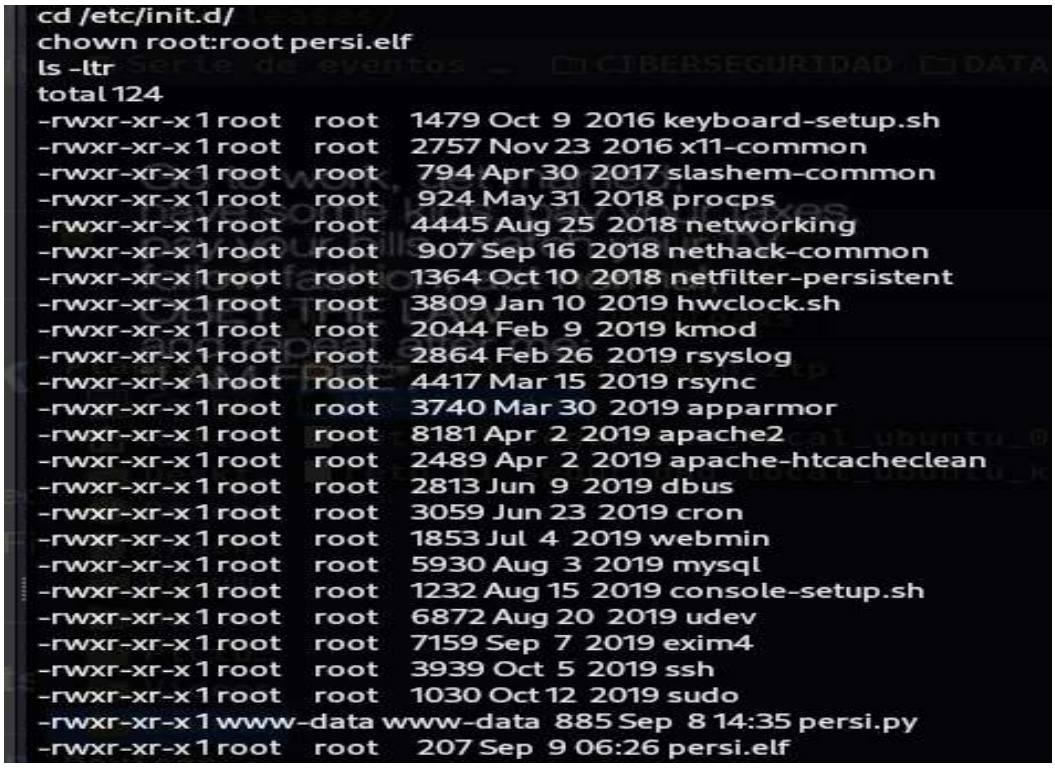
8. Una vez que obtenemos el acceso root a través de la shell básica, vamos a cargar un payload usando MSFvenom para poder ganar una meterpreter persistente en el sistema con privilegios root.
   Para ellos en nuestra Kali, hacemos el payload:



9. Se procede a subirlo a la maquina objetivo a través de meterpreter:

10. Este paso es importante para ganar la persistencia, aunque la maquina sea reiniciada, por lo que, como tenemos permisos root, buscaremos ubicar nuestro payload en una zona que se inicie con el sistema, como el directorio */etc/init.d,* el cual contiene los scripts de inicio que se ejecutan durante el arranque.

11. Debido a que el archivo tiene permisos www-data, se procede a modificarlo para que tengan permisos root:



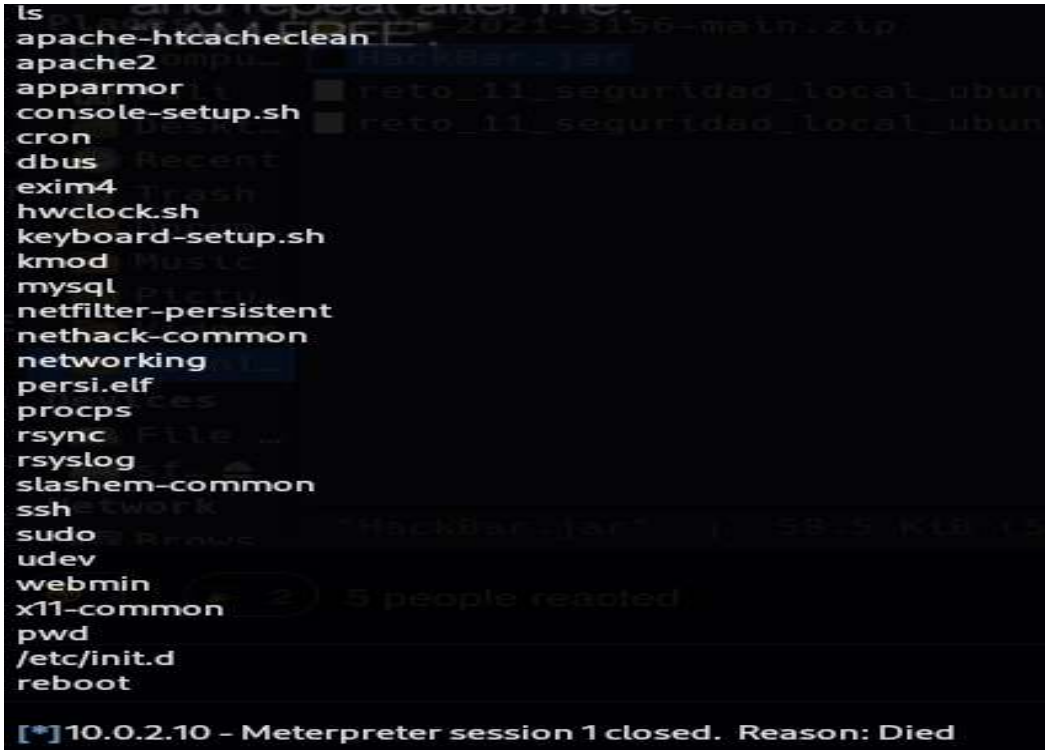
12. Una vez ubicado y con los permisos adecuados, es necesario automatizar nuestro payload en el inicio del sistema:

13. Aunque con esto ya debería bastar, vamos a agregarlo también al archivo de tareas programadas *“Crontab”,* dándole instrucciones que lo haga en cada reinicio del sistema:

```
echo "@reboot /etc/init.d/persi.elf" > /tmp/crontab_root
crontab /tmp/crontab_root
crontab -l
@reboot /etc/init.d/persi.elf
ls
```

Con el comando *"@reboot"* hará que nuestro script malicioso se ejecute cada vez que el sistema se reinicie.

14. Una vez inyectado nuestro script y obtenida la persistencia, se procede al reinicio del sistema atacado.



15. Finalmente, abrimos un handler en MetaExploit para poder conectar con nuestro payload después del reinicio, siendo la misma positiva, consiguiendo una shell interactiva avanzada con permisos root.