Tema4.4.5.4.EnriqueGomezTagle

sábado, 17 de febrero de 2024 11:59

Enrique Ulises Báez Gómez Tagle – 0241823 - 0241823@up.edu.mx
UP. 2024. Ene-Jun. Hackeo Ético y Recuperación Ante Desastres – Yoel Ledo Mezquita

d. Emitan los siguientes dos comandos en la **VM CyberOps Workstation**:

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls -l
```

¿Cuál es el significado de la salida? ¿Dónde se almacenan físicamente los archivos de la lista?

Ir al directorio raíz
Después listar los archivos en la raíz /dev/sda1

¿Por qué no se muestra **/dev/sdb1** en la salida de arriba?

No está montado

d. Ahora que se ha montado **/dev/sdb1** en **/home/analyst/second_drive**, utilicen **ls -l** para volver a generar una lista con el contenido del directorio.

```
[analyst@secOps ~]$ ls -l second_drive/
total 20
drwx------ 2 root root 16384 Mar 3 10:59 lost+found
-rw-r--r-- 1 root root 183 Mar 3 15:42 myFile.txt
```

¿Por qué ya no está vacío el directorio? ¿Dónde se almacenan físicamente los archivos de la lista?

/home/analyst/second_drive se convierte en el punto de entrada al sistema. Físicamente en dev/sdb1

b. Utilicen el comando **ls -l** para mostrar los permisos de archivo.

```
[analyst@secOps scripts]$ ls -l
total 60
-rwxr-xr-x 1 analyst analyst 190 Jun 13 09:45 configure_as_dhcp.sh
-rwxr-xr-x 1 analyst analyst 192 Jun 13 09:45 configure_as_static.sh
-rwxr-xr-x 1 analyst analyst 3459 Jul 18 10:09 cyberops_extended_topo_no_fw.py
-rwxr-xr-x 1 analyst analyst 4062 Jul 18 10:09 cyberops_extended_topo.py
-rwxr-xr-x 1 analyst analyst 3669 Jul 18 10:10 cyberops_topo.py
-rw-r--r-- 1 analyst analyst 2871 Apr 28 11:27 cyops.mn
-rwxr-xr-x 1 analyst analyst 458 May 1 13:50 fw_rules
-rwxr-xr-x 1 analyst analyst 70 Apr 28 11:27 mal_server_start.sh
drwxr-xr-x 2 analyst analyst 4096 Jun 13 09:55 net_configuration_files
-rwxr-xr-x 1 analyst analyst 65 Apr 28 11:27 reg_server_start.sh
-rwxr-xr-x 1 analyst analyst 189 Dec 15 2016 start_ELK.sh
-rwxr-xr-x 1 analyst analyst 85 Dec 22 2016 start_miniedit.sh
-rwxr-xr-x 1 analyst analyst 76 Jun 22 11:38 start_pox.sh
-rwxr-xr-x 1 analyst analyst 106 Jun 27 09:47 start_snort.sh
-rwxr-xr-x 1 analyst analyst 61 May 4 11:45 start_tftpd.sh
```

Consideren el archivo **cyops.mn** a modo de ejemplo. ¿Quién es el propietario del archivo? ¿Y del grupo?

De ambos es analyst

Los permisos para **cyops.mn** son **–rw-r--r--**. ¿Qué significa esto?

Propietario: Leer y escribir sin ejecutar
Grupo: solo leer
Usuarios: leer.

c. El comando **touch** es muy simple y útil. Permite crear rápidamente un archivo de texto vacío. Utilicen el siguiente comando para crear un archivo vacío en el directorio **/mnt**:

```
[analyst@secOps scripts]$ touch /mnt/myNewFile.txt
touch: cannot touch '/mnt/myNewFile.txt': Permission denied
```

¿Por qué no se creó el archivo? Genere una lista con los permisos, la propiedad y el contenido del directorio **/mnt** y explique qué sucedió. Cuando se agrega la opción **-d**, muestra el permiso del directorio principal. Registren las respuestas en las siguientes líneas.

```
[analyst@secOps ~]$ ls -ld /mnt
drwxr-xr-x 2 root root 4096 Jan 5 2018 /mnt
```

los permisos de ese directorio son del root. Solo root puede escribir

¿Qué se puede hacer para que el comando **touch** que se muestra arriba tenga éxito?

- con sudo
- cambiar permisos

d. El comando **chmod** se utiliza para cambiar los permisos de un archivo o directorio. Como antes, monten la partición **/dev/sdb1** en el directorio **/home/analyst/second_drive** que ya se creó en esta práctica de laboratorio:

```
[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/
```

e. Pasen al directorio **second_drive** y generen una lista con su contenido:

```
[analyst@secOps ~]$ cd ~/second_drive
[analyst@secOps second_drive]$ ls -l
total 20
drwx------ 2 root root 16384 Mar 3 10:59 lost+found
-rw-r--r-- 1 root root 183 Mar 3 15:42 myFile.txt
```

¿Cuáles son los permisos del archivo **myFile.txt**?aquí.

- rw- r- r

f. Utilicen el comando **chmod** para cambiar los permisos de **myFile.txt**.

```
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[analyst@secOps second_drive]$ ls -l
total 20
drwx------ 2 root root 16384 Mar 3 10:59 lost+found
-rw-rw-r-x 1 root root 183 Mar 3 15:42 myFile.txt
```

¿Cambiaron los permisos? ¿Cuáles son los permisos de **myFile.txt**?

- rw-rw-rx

sudo chmod 777 myFile.txt

¿Qué comando cambiaría los permisos de myFile.txt a rwxrwxrwx, con lo que se otorgaría acceso total al archivo a cualquier usuario del sistema?

```
[analyst@secOps second_drive]$ echo test >> myFile.txt
[sudo] contraseña para analyst:
[analyst@secOps second_drive]$ cat myFile.txt
```

¿Fue exitosa la operación? Explique.

Sí. Los permisos 666 permiten que propietario y grupo hagan cambios.

```
drwxr-xr-x 2 analyst analyst 4096 Aug 7 15:25 pcaps
drwxr-xr-x 7 analyst analyst 4096 Sep 20 2016 pox
-rw-r--r-- 1 analyst analyst 473363 Feb 16 15:32 sample.img
-rw-r--r-- 1 analyst analyst 65 Feb 16 15:45 sample.img_SHA256.sig
drwxr-xr-x 3 analyst analyst 4096 Jul 18 10:10 scripts
-rw-r--r-- 1 analyst analyst 25553 Feb 13 2017 SQL_Lab.pcap
```

Comparen los permisos del directorio **malware** con el archivo **mininet_services**. ¿Cuál es la diferencia entre la parte inicial de la línea de malware y la línea mininet_services?

-d indica un directorio y no un archivo.
El bit de ejecución.

incluidas las carpetas ocultas.

f. Cambien los nombres de los archivos originales: file1.txt y file2.txt, y observen el efecto en los archivos vinculados.

```
[analyst@secOps ~]$ mv file1.txt file1new.txt
[analyst@secOps ~]$ mv file2.txt file2new.txt

[analyst@secOps ~]$ cat file1symbolic
cat: file1symbolic: no such file or directory

[analyst@secOps ~]$ cat file2hard
hard
```

Observe que **file1symbolic** ahora es un enlace simbólico roto porque ha cambiado el nombre del archivo que apuntaba a to **file1.txt**, pero el archivo de enlace rígido **file2hard** todavía funciona correctamente porque apunta al inodo de **file2.txt** y no a su nombre, que ahora es **file2new.txt**.

¿Qué creen que sucedería con **file2hard** si abrieran un editor de texto y modificaran el texto de **file2new.txt**?

Cambiaría el contenido del otro pues apuntan al mismo nodo en el disco duro.