PrivEsc Aragog

Manera alternativa a la escalada, con solo dos enlaces simbólicos.

El momento de entrar como usuario florian vemos que se ejecuta un proceso cron cada cinco minutos, para saber el momento exacto solo hay que hacer un ls para ver la hora de creación de los archivos

$ ls -l dev_wiki

```
florian@aragog:/var/www/html$ ls -al
drwxrwxrwx 5 cliff  cliff   4096 Feb 19 12:05 dev_wiki
drw-r--r-- 5   cliff  cliff   4096 Dec 20 16:17 zz_backup
```

El proceso exacto es desconocido, pero si se ve en la lista de procesos que se van ejecutando alguno muy obvio como :

> cp -R /var/www/html/zz_backup/ /var/www/html/dev_wiki/

Además con la mera observación se ve que la tarea primero borra la carpeta dev_wiki, luego copia todo de zz_backup a dev_wiki (se ve en el comando anterior) y después cambia los permisos, posiblemente con un chmod -R

Objetivo, vamos a crear dentro de dev_wiki un enlace simbólico que apunte a zz_backup, ese enlace se creará justo despues de que la tarea programa borre dev_wiki y empiece a copiar, pero antes de que cambie los permisos, la manera mas sencilla de hacerlo es borrando primero dev_wiki y ejecutando este comando varias veces, se trata de **copiarlo y pegarlo repetidas veces justo en el segundo que se ejecute el cron**:

```
$ ln -s /var/www/html/zz_backup /var/www/html/dev_wiki/el_mal
```

Si la carpeta dev_wiki no existe este comando devolverá un error de lectura, eso significa que se ha hecho demasiado pronto, no hay problema, para eso espameamos el comando varias veces.
Si no devuelve nada, es el momento correcto, el enlace se ha creado.
Si devuelve “File exist” hemos espameado demasiado el comando, pero no pasa nada, porque eso significa que existe, no nos deja modificarlo, al no usar el switch -f.

Después de crear ese enlace llegará chmod y cambia los permisos de manera recursiva con lo que ya podemos escribir en *zz_backup*, y aqui creamos el segundo enlace simbolico a la carpeta */root*:

```
$ ln -s /root/ /var/www/html/zz_backup/the_evil
```

La siguiente vez que cron se ejecute copiará todo el contenido de */root* en *dev_wiki*, y le pondrá permisos de lectura para todos los usuarios.

¿Porque hace falta dos enlaces y no uno solo? ¿Porque no se hace el primer enlace directo a /root?

Si no recuerdo mal, en la máquina al intentar hacerlo directo no funcionaba, la tarea de copia se ejecutaba como root pero la del cambio de permisos no, con lo que no dejaba cambiar los permisos de root directamente.
Seria interesante ver el fichero */root/restore.sh*