

[Escribir el nombre de la compañía]

Práctica 6: Nueva Llamada

Diseño de Sistemas Operativos
David Guillermo Morales Sáez

Índice

Introducción..... 2

Diseño 2

Conclusión..... 2

Anexo 3

Introducción

En esta práctica, se nos ha dado cancha libre para poder trabajar como deseemos con el kernel de linux, ya sea creando una nueva llamada del sistema o modificando una ya existente. Yo he optado por modificar una ya existente, EXEC, mostrando ciertos elementos internos de cada proceso.

Diseño

Se ha modificado la función *do_execve_common*, principal para el uso de la llamada del sistema EXEC, para que vaya mostrando en el la pantalla de depuración cada una de las llamadas que se le han hecho con el formato:

!!!: Se ha lanzado el proceso **NOMBRE** con **N** argumentos

Para ello, se ha utilizado la función *printk* y se han accedido a los campos correspondientes en la estructura *linux_binprm*:

- Nombre: campo **filename**
- Número de argumentos: campo **argc**

Por otro lado, para poder ver la salida de cada llamada, hemos de lanzar el comando *dmesg* en la terminal.

Conclusión

Al trabajar dentro del kernel de linux, se pueden llegar a hacer un gran número de cosas y acceder a una alta cantidad de información, pero hay que tener en cuenta que no se puede hacer de todo, ya que no se pueden utilizar las funciones estándar de C (como *printf* o *fopen*), ya que constituirían un fallo grave de seguridad y de estabilidad, ya que el núcleo ha de ser completo, es decir, no ha de requerir un software externo para poder funcionar, y las funciones estándar de C están fuera del núcleo (por eso podemos actualizar la versión que se utiliza).

Anexo

A continuación, se mostrará el código que se ha incluido, además de parte del código integrante base superior e inferior para poder ubicar su posición en el fichero:

Parte superior

```
bprm->exec = bprm->p;
retval = copy_strings(bprm->envc, envp, bprm);
if (retval < 0)
    goto out;

retval = copy_strings(bprm->argc, argv, bprm);
if (retval < 0)
    goto out;

retval = search_binary_handler(bprm,regs);
if (retval < 0)
    goto out;
```

Nuevo código:

```
/* Obtenemos los datos a almacenar */

printk(KERN_DEBUG "!!!: Se ha lanzado el proceso %s con %d
argumentos\n", bprm->filename, bprm->argc);
```

Parte inferior:

```
/* execve succeeded */
current->fs->in_exec = 0;
current->in_execve = 0;
acct_update_integrals(current);
free_bprm(bprm);
if (displaced)
    put_files_struct(displaced);
return retval;
```