

SEGURIDAD EN SISTEMAS OPERATIVOS (2017-2018)  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Práctica 2 Sesión 2

---



**UNIVERSIDAD  
DE GRANADA**

Iván Rodríguez Millán  
ivanrodmil@gmail.com

## Índice

- |   |  |   |
|---|--|---|
| 1 | Ejercicio 1: Para el sistema que utilizas, indica la arquitectura, distribución y compilador que utilizas e indica las protecciones que se utilizan de cara a proteger un binario ELF.   | 3 |
| 2 | Ejercicio 2: Podemos mejorar el siguiente virus: Escribiendo un mejor scanner, limpiador para él. Añadir ifdef para comprobar la arquitectura de forma que cambie de forma automática el valor en tiempo de compilación para adaptarlo al sistema donde estemos. | 5 |

## Índice de figuras

1.1.	.....	3
1.2.	.....	3

## Índice de tablas

**1. Ejercicio 1: Para el sistema que utilizas, indica la arquitectura, distribución y compilador que utilizas e indica las protecciones que se utilizan de cara a proteger un binario ELF.**

Arquitectura y versión del compilador:

Tenemos como versión del Kernel: Linux-version 4.13.16.

A terminal window titled 'ivancito@ivancito:/home/ivancito' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The command 'cat /proc/version' is executed, showing the output: 'Linux version 4.13.16-200.fc26.x86\_64 (mockbuild@bkernel01.phx2.fedoraproject.org) (gcc version 7.2.1 20170915 (Red Hat 7.2.1-2) (GCC)) #1 SMP Mon Nov 27 18:19:47 UTC 2017'.

```
ivancito@ivancito:/home/ivancito
Archivo Editar Ver Buscar Terminal Ayuda
[root@ivancito ivancito]# cat /proc/version
Linux version 4.13.16-200.fc26.x86_64 (mockbuild@bkernel01.phx2.fedoraproject.org) (gcc version 7.2.1 20170915 (Red Hat 7.2.1-2) (GCC)) #1 SMP Mon Nov 27 18:19:47 UTC 2017
[root@ivancito ivancito]#
```

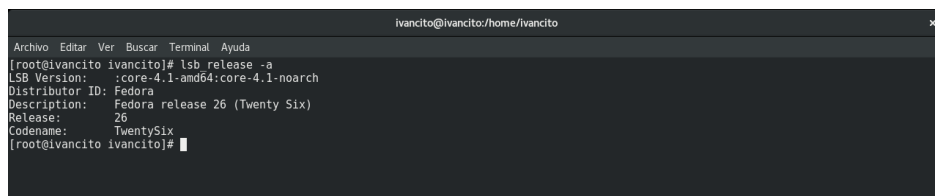
Figura 1.1: .

Como arquitectura x86\_64 (64 bits). [1]

Como versión del compilador: gcc 7.2.1.

Distribución:

En donde se aprecia perfectamente que es la distribución Fedora en la versión 26.

A terminal window titled 'ivancito@ivancito:/home/ivancito' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The command 'lsb\_release -a' is executed, showing the output: 'LSB Version: :core-4.1-amd64:core-4.1-noarch, Distributor ID: Fedora, Description: Fedora release 26 (Twenty Six), Release: 26, Codename: TwentySix'.

```
ivancito@ivancito:/home/ivancito
Archivo Editar Ver Buscar Terminal Ayuda
[root@ivancito ivancito]# lsb_release -a
LSB Version: :core-4.1-amd64:core-4.1-noarch
Distributor ID: Fedora
Description: Fedora release 26 (Twenty Six)
Release: 26
Codename: TwentySix
[root@ivancito ivancito]#
```

Figura 1.2: .

Con respecto a las protecciones que se utilizan de cara a proteger un binario ELF, tenemos: [3]

- Non-Executable Memory (NX) : Ayuda a prevenir ciertos tipos de explotaciones de desbordamiento de buffer. Los procesadores modernos soportan la característica llamada NX, que que permite a un sistema controlar la ejecución de varias porciones de memoria. [2]
- Stack Protector : gcc's -fstack-protector, protege contra los desbordamientos de pila. [4]
- Position Independent Executables (PIE) : Es comúnmente usado para librerías compartidas, de tal forma que el mismo código de librería puede ser cargado en una localización en cada espacio de direcciones de programa. [5]

- Address Space Layout Randomization (ASLR) : Es una técnica de seguridad para prevenir la explotación de vulnerabilidades de corrupción de memoria. [6] [2]

## 2. Ejercicio 2: Podemos mejorar el siguiente virus: Escribiendo un mejor scanner, limpiador para él. Añadir ifdef para comprobar la arquitectura de forma que cambie de forma automática el valor en tiempo de compilación para adaptarlo al sistema donde estemos.

Apartado A:

Apartado B: Para esto requerimos añadir a nuestro código las macros correspondientes, en nuestro caso lo haremos con GNUC [7]:

Ejemplo:

---

```
/* Test for GCC > 3.2.0 */
#if __GNUC__ > 3 || (__GNUC__ == 3 && (__GNUC_MINOR__ > 2 || (__GNUC_MINOR__ == 2
    && __GNUC_PATCHLEVEL__ > 0))
```

---

O también podemos directamente decir lo siguiente sin importarnos versiones de GNU:

---

```
#if __GNUC__
    #if __x86_64__
        #define 64ENV
    #else
        #define 32ENV
    #endif
#endif
#endif
```

---

Así en el caso de que se esté en una arquitectura de 64 bits, se puede indicar lo siguiente:

---

```
#ifdef 64ENV
    Elf64_Ehdr hdr;
#else
    Elf32_Ehdr hdr;
#endif
```

---

Ésta última información está obtenida por el siguiente enlace: [8].

Podemos encontrar mayor variedad de macros en los siguientes enlaces:

Sourceforge, macros asociadas a sistemas AMD [En línea], 5 de Diciembre del 2017, disponible en:

<https://sourceforge.net/p/predef/wiki/Architectures/>

Nadeausoftware, macros asociadas a sistemas INTEL [En línea], 5 de Diciembre del 2017, disponible en:

[http://nadeausoftware.com/articles/2012/02/c\\_c\\_tip\\_how\\_detect\\_processor\\_type\\_using\\_compiler\\_predefined\\_macros](http://nadeausoftware.com/articles/2012/02/c_c_tip_how_detect_processor_type_using_compiler_predefined_macros)

## Referencias

- [1] [https://fedoraproject.org/wiki/Architectures#Fedora\\_Architectures](https://fedoraproject.org/wiki/Architectures#Fedora_Architectures), consultado el 5 de Diciembre de 2017.
- [2] [https://fedoraproject.org/wiki/Security\\_Features\\_Matrix](https://fedoraproject.org/wiki/Security_Features_Matrix), consultado el 5 de Diciembre de 2017.
- [3] <https://www.akkadia.org/drepper/nonselsec.pdf>, consultado el 5 de Diciembre de 2017.
- [4] <https://access.redhat.com/blogs/product-security/posts/blueborne>, consultado el 5 de Diciembre de 2017.
- [5] <https://access.redhat.com/blogs/766093/posts/1975793>, consultado el 5 de Diciembre de 2017.
- [6] [https://en.wikipedia.org/wiki/Address\\_space\\_layout\\_randomization](https://en.wikipedia.org/wiki/Address_space_layout_randomization), consultado el 5 de Diciembre de 2017.
- [7] <https://gcc.gnu.org/onlinedocs/cpp/Common-Predefined-Macros.html>, consultado el 5 de Diciembre de 2017.
- [8] <https://linux.die.net/man/5/elf>, consultado el 5 de Diciembre de 2017.