SEGURIDAD EN SISTEMAS OPERATIVOS

4º Grado en Informática – Complementos de Ing. del Software

Curso 2017-18

Práctica [2]. Ingeniería inversa en Linux.

Sesión [2]. Explotaciones y protecciones del formato ELF

Autor[1]: Rubén Calvo Villazán

Ejercicio 1.

Para ver la arquitectura o la información completa del sistema, podemos hacer

```
$> uname -a
```

Vemos que nos aparece

```
Linux kali 4.13.0-kali1-amd64 #1 SMP Debian 4.13.10-1kali2 (2017-11-08) x86_64 GNU/Linux
```

Al ser un sistema operativo basado en Debian, éste usará las mismas protecciones que usa Debian.

Las podemos consultar en https://wiki.debian.org/Hardening

Entre otras, tenemos:

- Address Space Layout Randomization
- runtime memory allocation validation
- -fstack-protector
- non-exec memory segmentation (ExecShield)
- heap protection
- libc pointer encryption
- Stack Protector

---

1 Como autor declaro que los contenidos del presente documento son originales y elaborados por mi. De no cumplir con este compromiso, soy consciente de que, de acuerdo con la "Normativa de evaluación y de calificaciones de los estudiantes de la Universidad de Granada" esto "conllevará la calificación numérica de cero … independientemente del resto de calificaciones que el estudiante hubiera obtenido ..."

El compilador usado es

gcc (Debian 7.2.0-12) 7.2.1 20171025

Ejercicio 2.

a)

El cleaner o limpiador del virus lo que hace es copiar el archivo del virus sin el virus en un archivo nuevo limpio, de forma que se queda el archivo con el contenido original eliminando el virus de él.

Se usa para ello la orden

dd if=$INFECTED.vx of=$INFECTED count=$ORIG_SIZE skip=$VIRUS_SIZE bs=1

Que podemos mejorar cambiando el tamaño a copiar 'bs' en función del tamaño de nuestro archivo.

b)

Añadiendo la comprobación de la arquitectura en tiempo de compilación

```
// Comprobación de la arquitectura


#ifdef __amd64__

if (ehdr.e_machine != EM_X86_64) return 1; // AMD64

#elif __arm__

if (ehdr.e_machine != EM_ARM) return 1; // ARM

#else

   if (ehdr.e_machine != EM_386) return 1;

#endif
```