Arquitectura de Computadoras
TP1 - Análisis de Binarios

Ejercicios

Ejercicio 1 – Strings

El programa **strings** de Linux, busca en los archivos strings, y hace una lista de éstos.

1. Corra el programa **fortune**
2. Ejecute **strings** utilizando como argumento el programa **fortune**. Identifique todas las fortunas que dicho programa estaría indicando.
3. ¿Cómo hace dicho programa para encontrar los strings? Implemente su propia versión.

Ejercicio 2 – Bless

Abra con el **Bless Hex Editor** y el programa **fortune**. Busque los Strings que aparecen en el programa y cámbielos.

1. Corrija los horrores de ortografía.
2. Haga más corto el mensaje de bienvenida.

Ejercicio 3 - Objdump

La herramienta **objdump** de Linux permite hacer un análisis de archivos objeto revelando información importante de cómo está compuesto dicho archivo.

1. Haga un **disassembly** del código objeto y deduzca cómo es que se elige la fortuna a mostrarle al usuario. Si agrega el argumento “-M intel” podrá ver dicho código en formato Intel.
2. Investigue cuales son las secciones que tiene dicho archivo. ¿En qué sección se encuentran los Strings?
3. Identifique todas las etiquetas del archivo. ¿Cuales reconoce?

Ejercicio 4 – PW

Con los conocimientos adquiridos en los puntos anteriores,

1. Obtenga la contraseña del programa **password_easy**.
2. Cámbiela por “1234”.

Ejercicio 5 – PW ofuscated

Con los conocimientos adquiridos en los puntos anteriores,

1. Obtenga la contraseña del programa **password_ofuscated1**
2. Cámbiela por “1234”

Ejercicio 6 – Matríz

Considere el siguiente programa:

```
#include <stdio.h>

#define DIMENSION 1024

int matriz[DIMENSION][DIMENSION];

int main(void) {
      printf("Hello World!\n");
      return 0;
}
```

Compile y analice el tamaño del binario final. Piense, ¿Cuánto espacio ocupa una matriz de 1024x1024 de ints? ¿Dónde está toda esa información en el binario? ¿Qué tamaño esperaba?

¿Qué ocurriría si dicha matriz estuviera inicializada con 0’s? ¿Y con otro valor? Pruebe cómo varía el tamaño de dicho binario con las distintas alternativas.

```
int matriz[DIMENSION][DIMENSION] = {0};
```

Ejercicio 7 – EDB

Utilice el programa **Evan’s Debugger** con el programa **fortune** de los puntos anteriores. Corra cualquier programa e identifique los siguientes elementos de cada programa:

- Zona de Código
- Zona de Datos
- Stack

1. Conteste las siguientes preguntas:
   - ¿En qué lugar físico de la PC está la información que está visualizando?
   - ¿Algunas secciones están solapadas?
   - ¿Por qué en cada pantalla la información visualizada es distinta?
   - ¿Qué diferencia hay con los ejemplos anteriores?

- ¿En qué parte de la PC está la información que está analizando?

2. Confirme su suposición de cómo se está calculando la fortuna del usuario.
3. Abra el Programa **password_ofuscated1** y observe paso a paso cómo cambia la sección de datos a medida que se va generando la contraseña.
4. Modifique y guarde dicho programa de tal forma que sin importar la contraseña, siempre de cómo correcta.

Ejercicio 8 – PW hard

Utilice el **debugger** para deducir cómo son las contraseñas que son válidas para el programa **password_hard**.