**Estructura de Computadores (2016-2017)**
Subgrupo C3
Grado en Ingeniería Informática
Universidad de Granada

# Práctica 4: Bomba Digital - desensambladores

Mario Rodríguez Ruiz

6 de diciembre de 2016

# Índice



# Índice de figuras

# 1. Programar la bomba digital

## 1.1. Fichero fuente

```
/************************************
*    Mario Rodríguez Ruiz      *
*    Bomba digital          *
*    Estructura de Computadores    *
*    Diciembre 2016         *
*                                   *
*      gcc -m32 bomba.c -o bomba     *
************************************/

#include <stdio.h>  // para printf()
#include <stdlib.h> // para exit()
#include <string.h> // para strncmp()/strlen()
#include <sys/time.h> // para gettimeofday(), struct timeval

char ebx[]="%edx";
int  pc    = 1892;

void boom(){
  printf("***************\n");
  printf("*** BOOM!!! ***\n");
  printf("***************\n");
  exit(-1);
}

void defused(){
  printf("**************************\n");
  printf("*** bomba desactivada ***\n");
  printf("**************************\n");
  exit(0);
}

int main(){
#define SIZE 100
  char pass[SIZE];
  int  pasv;
    int pdn = pc*strlen(ebx) ;
  ebx[2] = ebx[2] - 2 ;
#define TLIM 5
  struct timeval tv1,tv2; // gettimeofday() secs-usecs

```

```
41    gettimeofday(&tv1,NULL);
42    printf("Introduce la contraseña: ");
43    fgets(pass,SIZE,stdin);
44    if (strncmp(pass,ebx,strlen(ebx)))
45        boom();
46
47    gettimeofday(&tv2,NULL);
48    if (tv2.tv_sec - tv1.tv_sec > TLIM)
49        boom();
50
51    printf("Introduce el código: ");
52    scanf("%i",&pasv);
53    if (pasv*strlen(ebx)!=pdn) boom();
54
55    gettimeofday(&tv1,NULL);
56    if (tv1.tv_sec - tv2.tv_sec > TLIM)
57        boom();
58
59    defused();
60 }
```

../fuentes/bomba_RodriguezRuizMario.c

Este código fuente hace dos modificaciones distintas para la contraseña y el código de la bomba.

La contraseña inicial es **%ebx** (el nombre se ha elegido para despistar un poco) y se modificará más adelante y el código es **1892**(que no va a ser modificado) como puede verse en las lineas 15 y 16.

En el main es donde se realizan las modificaciones para dificultar un poco la búsqueda de ambos valores:
Para la contraseña lo que se hace es modificar el segundo caracter de la cadena original (%edx), restando en dos el valor ASCII de la tercera posición de ésta, **convirtiéndose la 'd' en una 'b'**. Linea 37.
A continuación, lo siguiente que se modifica es el código, que se convierte en un valor distinto a través de una multiplicación que se realiza mediante él mismo. Se **multiplica** su valor por el tamaño de la cadena de la contraseña, en este caso **tres**. Linea 36.

Por último se realiza la misma acción anterior para el código que se introduzca por pantalla y así poder compararlos con el nuevo valor modificado.

# 2. Desactivación de la bomba

## 2.1. Contraseña

Recorriendo el código desde el principio con la herramienta **DDD**, poniendo un punto de ruptura en el comienzo del main se puede obtener el primer indicio para averiguar la contraseña. Éste es el que se tiene cuando se carga por primera vez el registro **%eax** con $0x804a030$ en el que si se mira su valor a través de **Data** → **Memory** con los valores que aparecen en la Figura 2.1

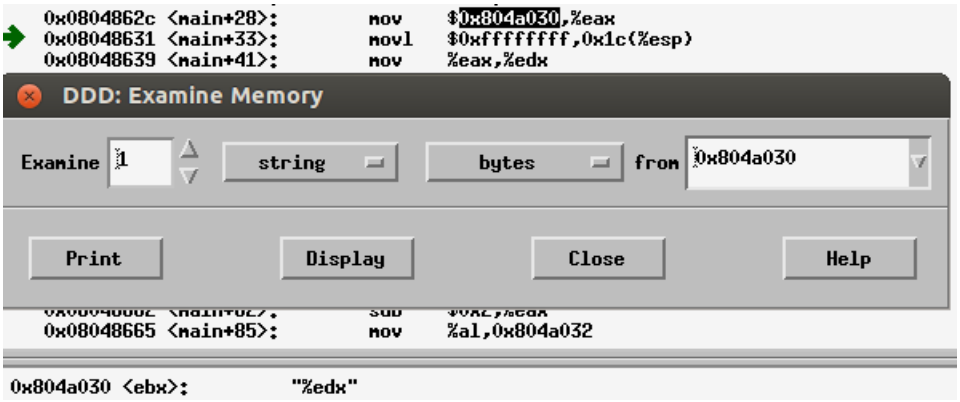


Figura 2.1: Primer indicio para averiguar la contraseña

Como se aprecia, lo que se imprime es **%edx** que es la contraseña inicial aún sin modificar. Esta cadena puede provocar cierto despiste por su nombre.

Continuando la ejecución mediante **Nexti** de **Command Tool** se llega a un punto en el que la cadena "sospechosa" anterior se va a modificar. En la Figura 2.2 aparecen tres lineas importantes. Basta con ver la primera para cerciorarse que la cadena se va a modificar. Mediante la herramienta anterior (memory) se comprueba la dirección $0x804a032$ y se ve que es **dx**. La siguiente instrucción lo qe hace es **restarle dos al valor 0x64** que se corresponde con la letra **d**.

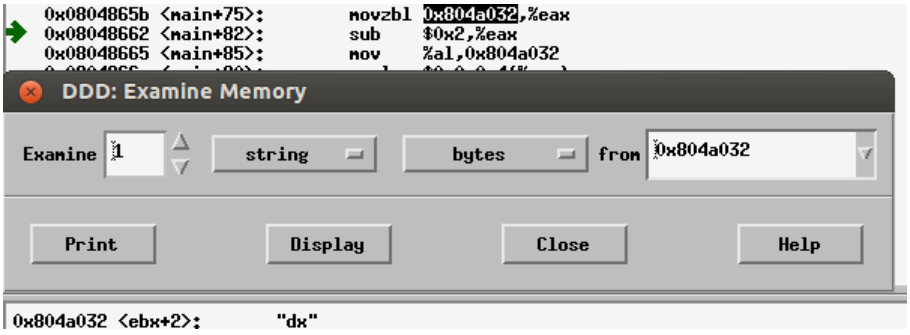


Figura 2.2: Segundo indicio para averiguar la contraseña

Después de esas tres instrucciones se revelará un dato importante. Si se comprueba el valor de la misma dirección anterior ($0x804a032$) ahora tendrá un nuevo valor: **bx**, tal y como muestra la Figura 2.3

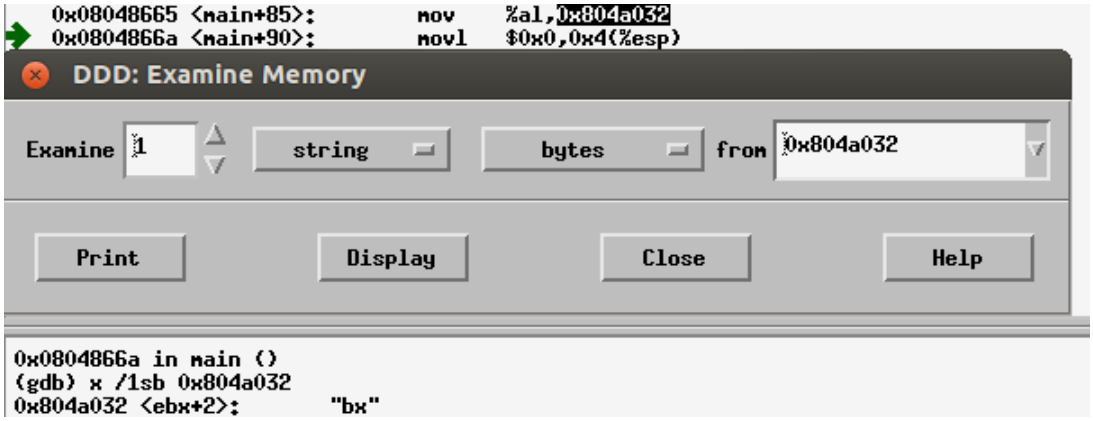


Figura 2.3: Tercer indicio para averiguar la contraseña

Por último, comprobando el valor final que se queda en %eax justo después de la llamada a **fgets** se obtiene la **contraseña**. En la Figura 2.4 puede verse la impresión del valor del registro y el obsequio de contraseña: **%ebx**.


Figura 2.4: Obtención del valor de la contraseña

## 2.2. Código

Para obtener el código válido se va a proceder una estrategia parecida a la de la obtención de la contraseña.
Recorriendo el código desde el principio con la herramienta **DDD**, poniendo un punto de ruptura en el comienzo del **main** se puede obtener el primer indicio para averiguar el código buscado.

Se llega a un punto en el que se realizan instrucciones "sospechosas", como el de una multiplicación (instrucción señalada con la flecha en la Figura 2.5). Como se ve en la Figura 2.5 el registro **%eax**, con valor decimal 1892, se multiplica por el registro **%edx**, con valor decimal 4.

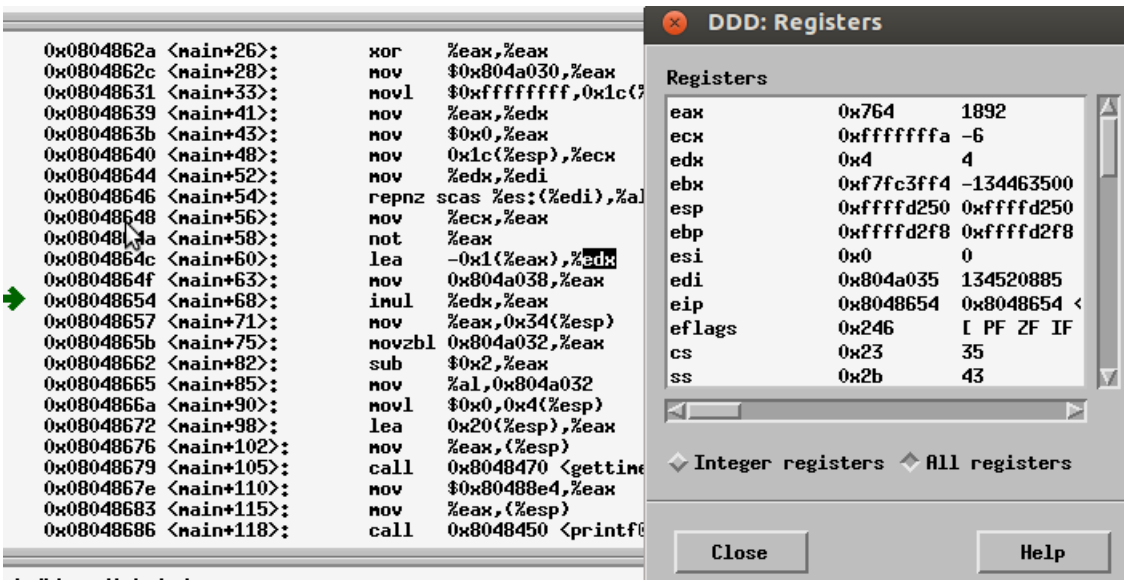


Figura 2.5: Primer indicio para averiguar la contraseña

Una vez que se requiere que se introduzca el código, en este caso se introduce el **1111** para probar, se puede observar a continuación cómo es transformado su valor por el de **4444** antes de proceder a su comparación como se ve en la Figura 2.6.

Es de aquí donde se puede afirmar que se realiza una multiplicación por 4 igual que se había visto anteriormente.

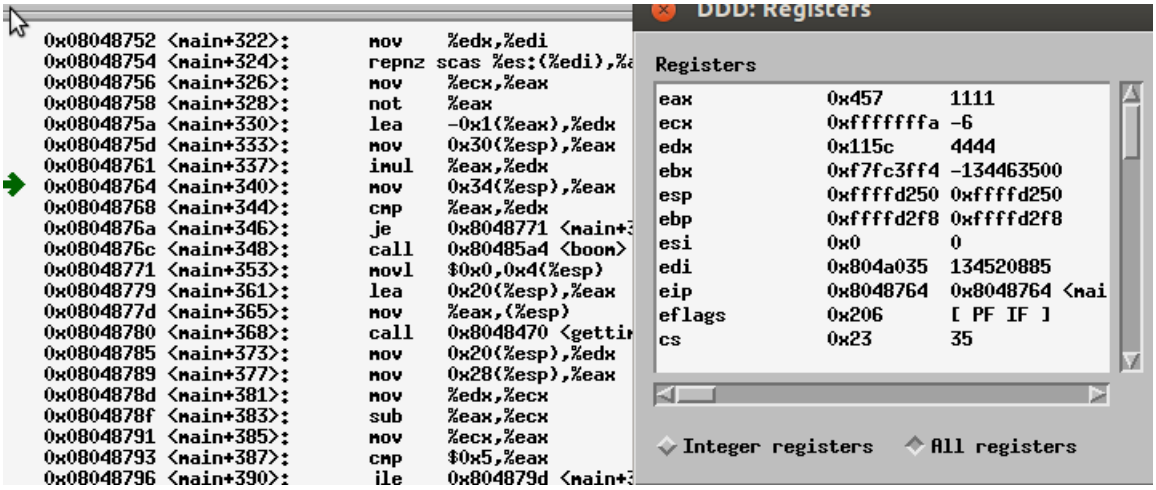


Figura 2.6: Primer indicio para averiguar la contraseña

Por último aparece la parte crítica, en la que se corroborarán los resultados obtenidos hasta ahora o no. Dicha parte es la de la comparación, en la que se ve que se comparan los valores **7568** y **4444**, véase en la Figura 2.7. El valor 7568 aparece como se vio anteriormente de la multiplicación de **1892 por 4** y 4444 es el código que se ha introducido por pantalla como prueba (1111) multiplicado también por cuatro.

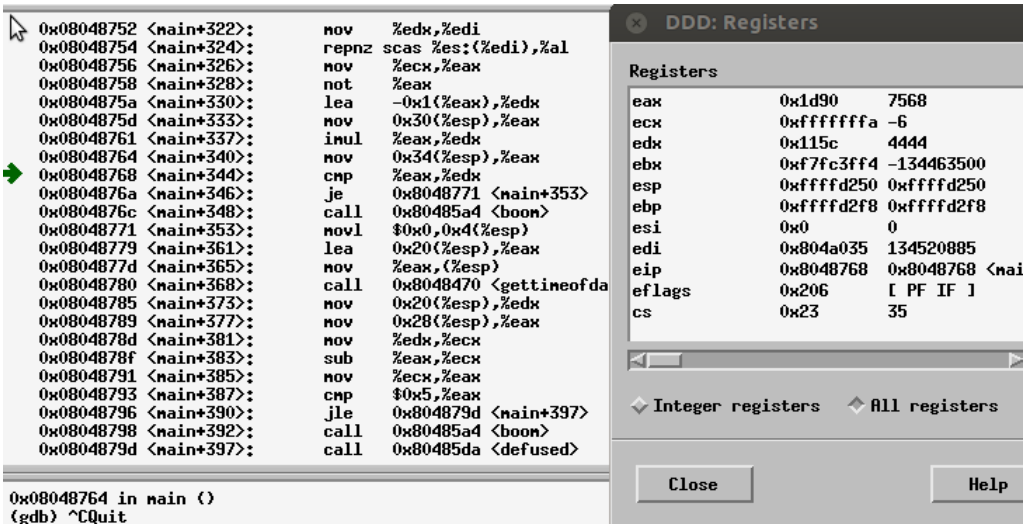


Figura 2.7: Primer indicio para averiguar la contraseña

Gracias a la herramienta **Status** → **Registers** de **DDD** ha sido posible seguir el transcurso de los registros y la modificación de sus valores en cada instrucción hasta conseguir el código final **1892**.