Desactivación Bomba de María Camarero

Estructura de Computadores - Grupo C3

Mario Rodríguez Ruiz

1. Contraseña

Para averiguar la contraseña se ha utilizado el depurador **DDD** realizando los siguientes pasos:

En primer lugar se ha puesto un punto de ruptura en la llamada a **fgets**, que es cuando se le pide al usuario por pantalla que ingrese una contraseña.

Una vez funcionando el programa y detenido ahí se ha metido una contraseña al azar para avanzar.

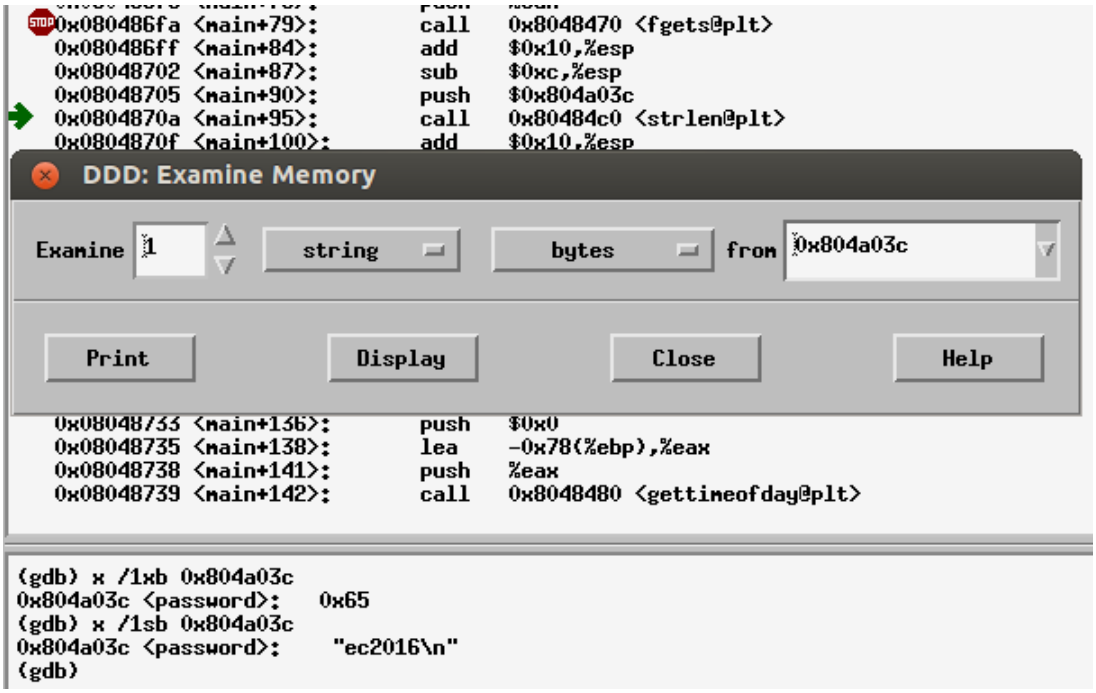

Figura 1.1: Examine Memory desde DDD

Justo después del guardado de la contraseña introducida, se hace un **push** "sospechoso". Ha sido el momento de examinar los datos en memoria mediante la opción **Data** → **Memory**: Poniendo Examine a 1, tipo **string** en **bytes** desde la dirección **0x804a03c** y haciendole un **print** ha mostrado la contraseña tal y como se ve en la Figura 1.1.

Contraseña: **ec2016**

2. Código

Para averiguar la contraseña se ha utilizado el depurador **DDD** realizando los siguientes pasos:

Llegando a la sección del programa que se encarga de procesar y validar el código, se ha puesto un punto de ruptura en esta parte para futuros intentos.

Como se ve en la Figura 2.1 el programa hace una llamada a una función con nombre **desactivar**, por lo que mediante **stepi** la ejecución seguirá su curso adentrándose en ésta.

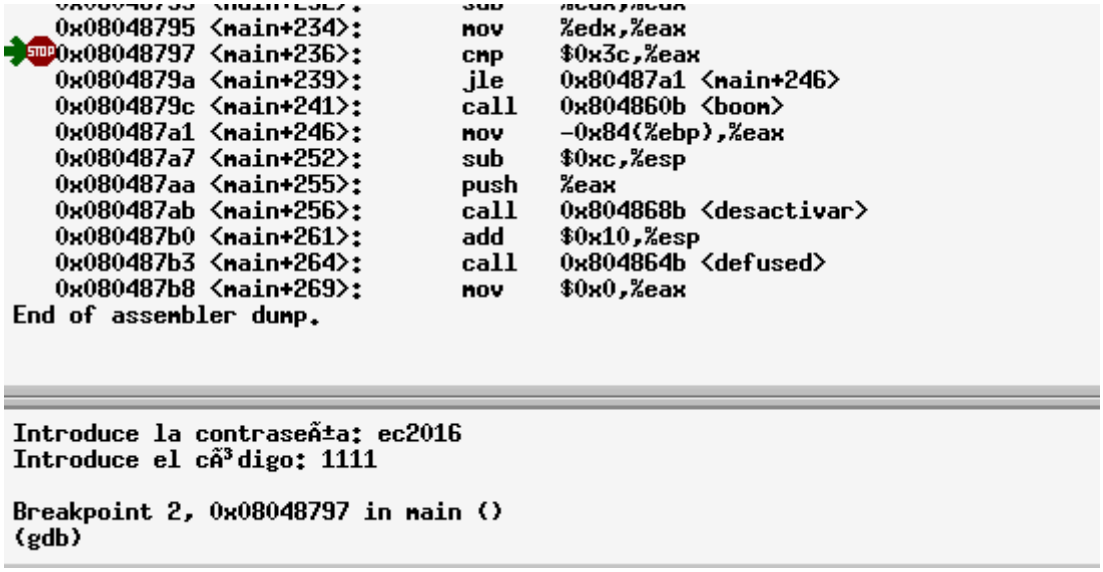

Figura 2.1: Examine Memory desde DDD para el código

Una vez dentro de dicha función, tal y como se muestra en la Figura 2.2, se pone un punto de ruptura para futuros intentos y a continuación se examina.

Figura 2.2: Examine Memory desde DDD para el código

Mediante la ayuda del estado de los registros (**Status→Registers**) y avanzando una a una cada instrucción se puede ver cómo van modificando sus valores.

El código de prueba introducido ha sido **1111**, sin embargo "extrañamente" se ha visto cómo se ha modificado su valor convirtiéndose ahora en **-905**, como puede verse en la Figura 2.3.
Es ahora cuando se hace una comparación de este nuevo valor con el valor que contiene **%eax** que es cero.

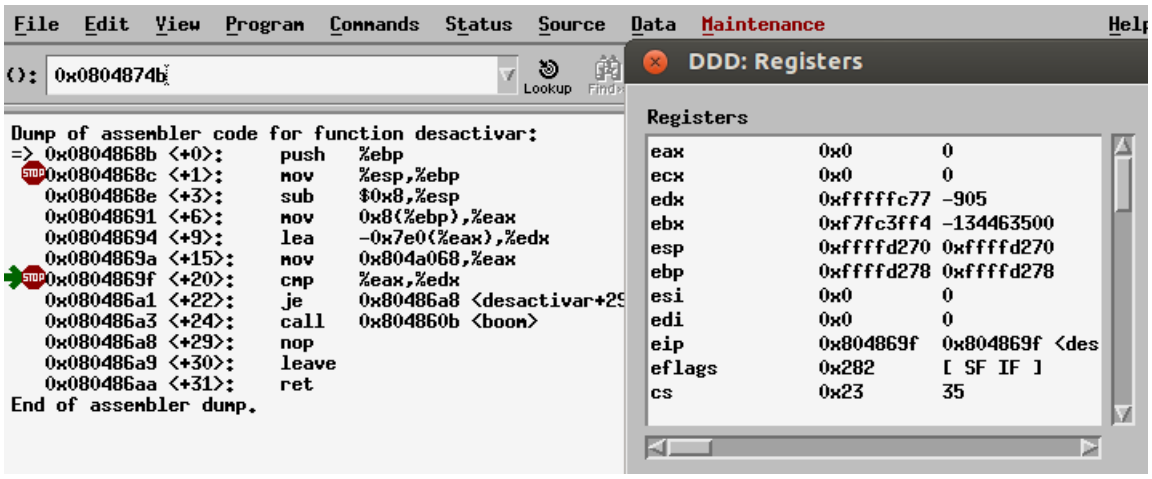

Figura 2.3: Examine Memory desde DDD para el código

Aquí es cuando se aprecia que se ha realizado una resta sobre el valor introducido y que su resultado tiene que ser cero. Por lo que sumando el valor introducido como prueba (1111) y el resultado después de la modificación (905) aparece la solución al problema: 1111+905=2016

Por tanto, el código es **2016**.