Parte 1 – Bomba

Procederemos a explicar el desarrollo de las pautas necesarias para poder desactivar mi bomba.

Primeramente mostraré todo el código maquina referente al **main** y al metodo **encriptar**.

***** MAIN *****

```
080486d0 <main>:
 80486d0:       8d 4c 24 04             lea    0x4(%esp),%ecx
 80486d4:       83 e4 f0                and    $0xfffffff0,%esp
 80486d7:       ff 71 fc                pushl  -0x4(%ecx)
 80486da:       55              push   %ebp
 80486db:       89 e5                   mov    %esp,%ebp
 80486dd:       51              push   %ecx
 80486de:       81 ec 84 00 00 00       sub    $0x84,%esp
 80486e4:       65 a1 14 00 00 00       mov    %gs:0x14,%eax
 80486ea:       89 45 f4                mov    %eax,-0xc(%ebp)
 80486ed:       31 c0                   xor    %eax,%eax
 80486ef:       83 ec 08                sub    $0x8,%esp
 80486f2:       6a 00                   push   $0x0
 80486f4:       8d 45 80                lea    -0x80(%ebp),%eax
 80486f7:       50              push   %eax
 80486f8:       e8 83 fd ff ff          call   8048480 <gettimeofday@plt>
 80486fd:       83 c4 10                add    $0x10,%esp
 8048700:       83 ec 0c                sub    $0xc,%esp
 8048703:       68 e4 88 04 08          push   $0x80488e4
 8048708:       e8 53 fd ff ff          call   8048460 <printf@plt>
 804870d:       83 c4 10                add    $0x10,%esp
 8048710:       a1 60 a0 04 08          mov    0x804a060,%eax
 8048715:       83 ec 04                sub    $0x4,%esp
 8048718:       50              push   %eax
 8048719:       6a 64                   push   $0x64
 804871b:       8d 45 90                lea    -0x70(%ebp),%eax
 804871e:       50              push   %eax
 804871f:       e8 4c fd ff ff          call   8048470 <fgets@plt>
 8048724:       83 c4 10                add    $0x10,%esp
 8048727:       83 ec 0c                sub    $0xc,%esp
 804872a:       8d 45 90                lea    -0x70(%ebp),%eax
 804872d:       50              push   %eax
 804872e:       e8 d8 fe ff ff          call   804860b <encriptar>
 8048733:       83 c4 10                add    $0x10,%esp
 8048736:       83 ec 0c                sub    $0xc,%esp
 8048739:       68 3c a0 04 08          push   $0x804a03c
 804873e:       e8 7d fd ff ff          call   80484c0 <strlen@plt>
 8048743:       83 c4 10                add    $0x10,%esp
 8048746:       83 ec 04                sub    $0x4,%esp
 8048749:       50              push   %eax
 804874a:       68 3c a0 04 08          push   $0x804a03c
 804874f:       8d 45 90                lea    -0x70(%ebp),%eax
```

```
8048752:    50              push   %eax
8048753:    e8 98 fd ff ff          call   80484f0 <strncmp@plt>
8048758:    83 c4 10                add    $0x10,%esp
804875b:    85 c0                   test   %eax,%eax
804875d:    74 05                   je     8048764 <main+0x94>
804875f:    e8 ec fe ff ff          call   8048650 <boom>
8048764:    83 ec 08                sub    $0x8,%esp
8048767:    6a 00                   push   $0x0
8048769:    8d 45 88                lea    -0x78(%ebp),%eax
804876c:    50              push   %eax
804876d:    e8 0e fd ff ff          call   8048480 <gettimeofday@plt>
8048772:    83 c4 10                add    $0x10,%esp
8048775:    8b 55 88                mov    -0x78(%ebp),%edx
8048778:    8b 45 80                mov    -0x80(%ebp),%eax
804877b:    29 c2                   sub    %eax,%edx
804877d:    89 d0                   mov    %edx,%eax
804877f:    83 f8 05                cmp    $0x5,%eax
8048782:    7e 05                   jle    8048789 <main+0xb9>
8048784:    e8 c7 fe ff ff          call   8048650 <boom>
8048789:    83 ec 0c                sub    $0xc,%esp
804878c:    68 ff 88 04 08          push   $0x80488ff
8048791:    e8 ca fc ff ff          call   8048460 <printf@plt>
8048796:    83 c4 10                add    $0x10,%esp
8048799:    83 ec 08                sub    $0x8,%esp
804879c:    8d 85 7c ff ff ff       lea    -0x84(%ebp),%eax
80487a2:    50              push   %eax
80487a3:    68 16 89 04 08          push   $0x8048916
80487a8:    e8 33 fd ff ff          call   80484e0 <__isoc99_scanf@plt>
80487ad:    83 c4 10                add    $0x10,%esp
80487b0:    8b 95 7c ff ff ff       mov    -0x84(%ebp),%edx
80487b6:    a1 48 a0 04 08          mov    0x804a048,%eax
80487bb:    39 c2                   cmp    %eax,%edx
80487bd:    74 05                   je     80487c4 <main+0xf4>
80487bf:    e8 8c fe ff ff          call   8048650 <boom>
80487c4:    83 ec 08                sub    $0x8,%esp
80487c7:    6a 00                   push   $0x0
80487c9:    8d 45 80                lea    -0x80(%ebp),%eax
80487cc:    50              push   %eax
80487cd:    e8 ae fc ff ff          call   8048480 <gettimeofday@plt>
80487d2:    83 c4 10                add    $0x10,%esp
80487d5:    8b 55 80                mov    -0x80(%ebp),%edx
80487d8:    8b 45 88                mov    -0x78(%ebp),%eax
80487db:    29 c2                   sub    %eax,%edx
80487dd:    89 d0                   mov    %edx,%eax
80487df:    83 f8 05                cmp    $0x5,%eax
80487e2:    7e 05                   jle    80487e9 <main+0x119>
80487e4:    e8 67 fe ff ff          call   8048650 <boom>
80487e9:    e8 a2 fe ff ff          call   8048690 <defused>
80487ee:    b8 00 00 00 00          mov    $0x0,%eax
80487f3:    8b 4d f4                mov    -0xc(%ebp),%ecx
```

```
 80487f6:       65 33 0d 14 00 00 00 xor    %gs:0x14,%ecx
 80487fd:       74 05                je     8048804 <main+0x134>
 80487ff:       e8 8c fc ff ff       call   8048490 <__stack_chk_fail@plt>
 8048804:       8b 4d fc             mov    -0x4(%ebp),%ecx
 8048807:       c9             leave
 8048808:       8d 61 fc             lea    -0x4(%ecx),%esp
 804880b:       c3             ret
 804880c:       66 90                xchg   %ax,%ax
 804880e:       66 90                xchg   %ax,%ax
```

***** ENCRIPTAR *****

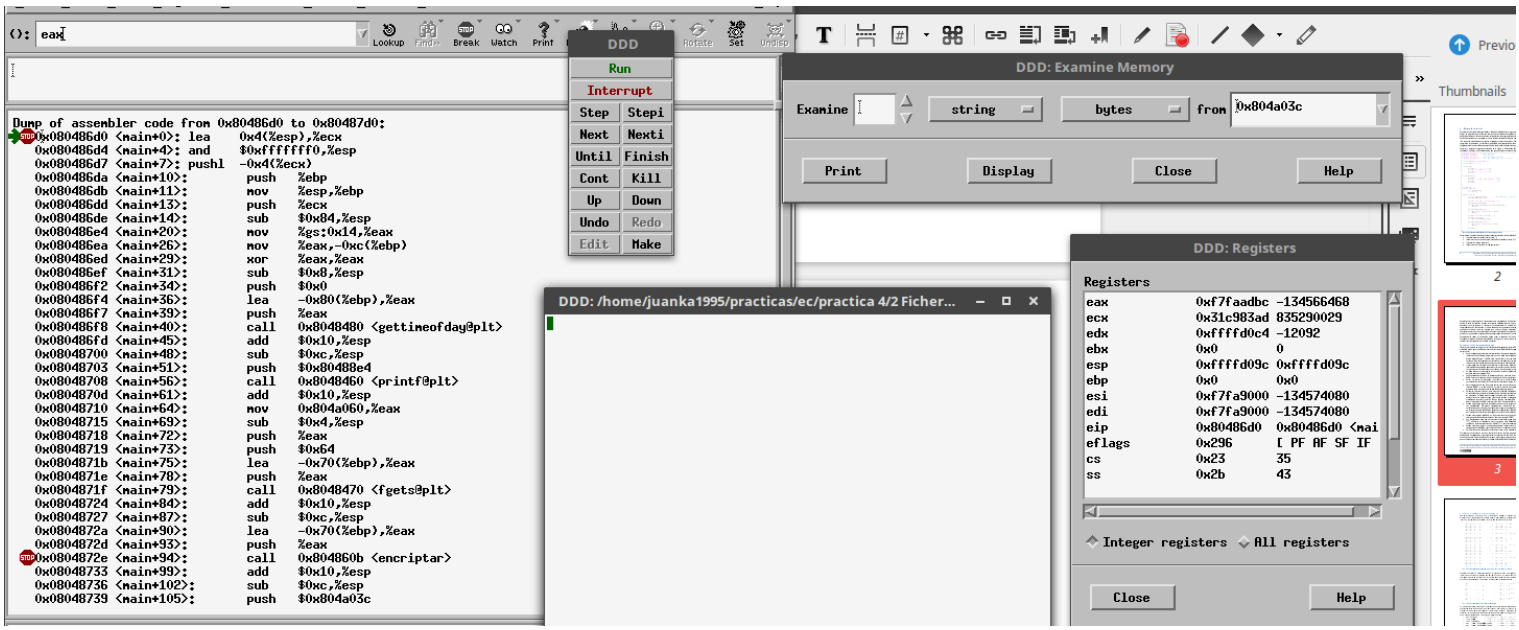
```
0804860b <encriptar>:
 804860b:       55             push   %ebp
 804860c:       89 e5                mov    %esp,%ebp
 804860e:       83 ec 18             sub    $0x18,%esp
 8048611:       c7 45 f4 00 00 00 00 movl   $0x0,-0xc(%ebp)
 8048618:       eb 1c                jmp    8048636 <encriptar+0x2b>
 804861a:       8b 55 f4             mov    -0xc(%ebp),%edx
 804861d:       8b 45 08             mov    0x8(%ebp),%eax
 8048620:       01 d0                add    %edx,%eax
 8048622:       8b 4d f4             mov    -0xc(%ebp),%ecx
 8048625:       8b 55 08             mov    0x8(%ebp),%edx
 8048628:       01 ca                add    %ecx,%edx
 804862a:       0f b6 12             movzbl (%edx),%edx
 804862d:       83 c2 04             add    $0x4,%edx
 8048630:       88 10                mov    %dl,(%eax)
 8048632:       83 45 f4 01          addl   $0x1,-0xc(%ebp)
 8048636:       83 ec 0c             sub    $0xc,%esp
 8048639:       ff 75 08             pushl  0x8(%ebp)
 804863c:       e8 7f fe ff ff       call   80484c0 <strlen@plt>
 8048641:       83 c4 10             add    $0x10,%esp
 8048644:       89 c2                mov    %eax,%edx
 8048646:       8b 45 f4             mov    -0xc(%ebp),%eax
 8048649:       39 c2                cmp    %eax,%edx
 804864b:       77 cd                ja     804861a <encriptar+0xf>
 804864d:       90             nop
 804864e:       c9             leave
 804864f:       c3             ret
```

****

Pasos para sacar la password:

Lo primero que deberiamos hacer es ejecutar nuestro fichero ejecutable **bomba** con **ddd** para depurarlo paso a paso y poder descubrir cual es la contraseña para desactivar nuestra bomba.

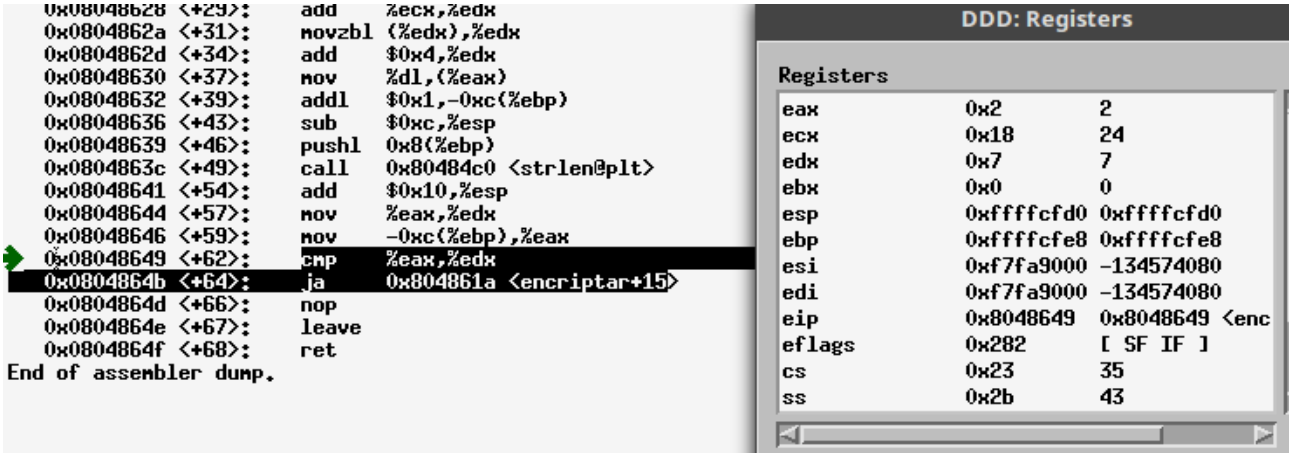
Una vez abierto tal que así:



Deberemos ir ejecutando paso a paso nuestro programa. Primeramente nos pedira la contraseña que como no la sabemos para probar yo he introducido **“juanka”**.

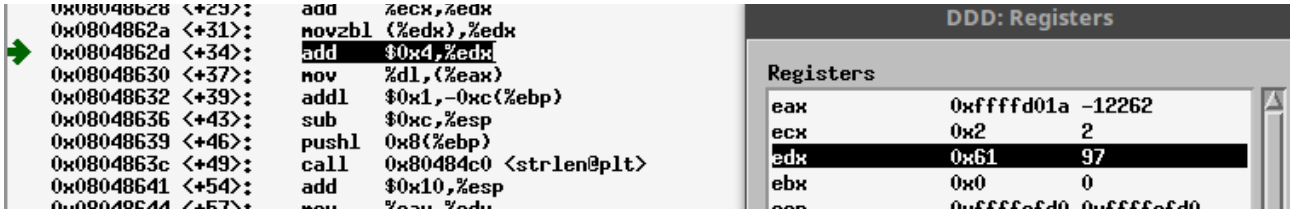
Despues de esto llegara al **2º breakpoint** que podemos ver en la imagen anterior. Este breakpoint esta justo en la llamada al metodo **encriptar** que será mediante el cual seamos capaces de sacar que algoritmo de cifrado se ha utilizado.

En la parte del metodo **encriptar** si lo ejecutamos paso a paso podremos ver que lo que se encuentra dentro de dicho metodo es un bucle ya que se realiza una comparación de **%eax** con **%edx** donde eax sería el indice.



Si analizamos de forma mas detallada este codigo veremos que cargamos en **%eax** de **-0x8(%ebp)** la clave que hemos introducido, en mi caso **“juanka”** y en **%edx** cargamos el indice, mediante **-0xc(%ebp)**. Ahora moveriamos el mismo contenido que tiene **%eax** al registro **%edx**. Este es el que recibira las modificaciones.

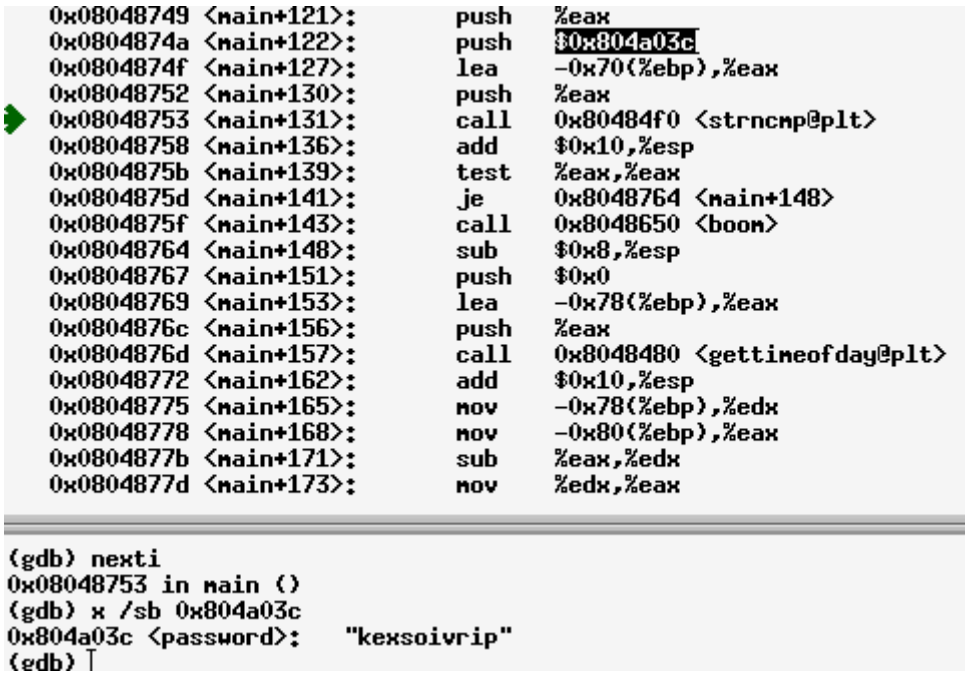
Si nos fijamos en las siguientes 2 imágenes, veremos que lo que realmente se hace es añadir **0x4** a **%edx**. En la primera imagen **%edx** tiene el valor **97** y tras la ejecucion de la instruccion **add** vemos en la segunda imagen que su valor a cambiado a **101**.



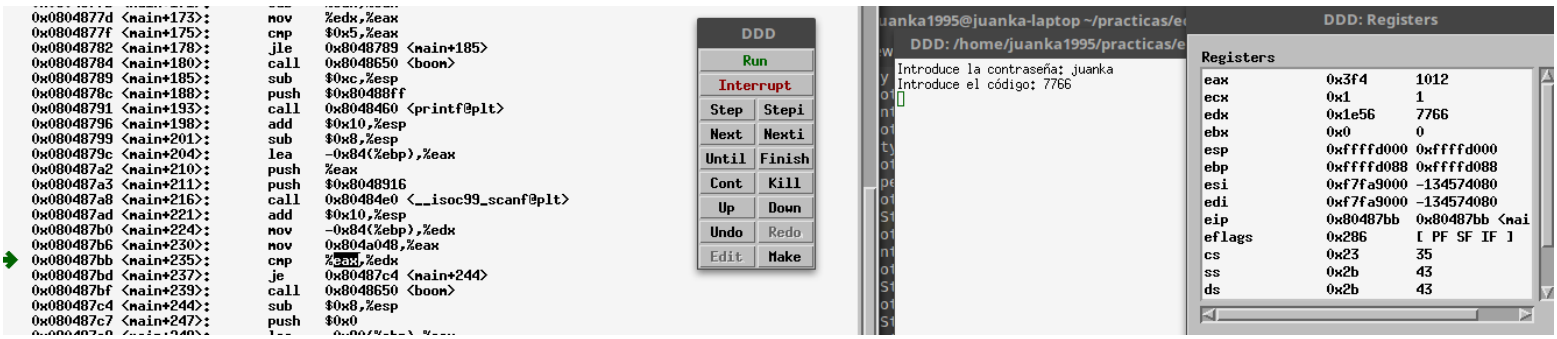
Por lo tanto el algoritmo utilizado es **sumarle 4 a cada letra** de la clave.

Ahora necesitamos saber cual es la password correcta, la cual estará cifrada, pero como ya sabemos el algoritmo de cifrado solo deberiamos aplicarselo a la inversa para obtener la contraseña correcta.

En la siguiente imagen podemos ver que en la dirección de memoria **0x804a03c** es donde se encuentra almacenada la contraseña correcta cifrada. Si aplicamos el algoritmo de cifrado a la inversa obtendriamos la clave **gatokernel**.



Por utltimo faltaría averiguar la clave de digitos cosa mas sencilla sabiendo ya la password. Para ello ejecutaremos paso a paso con el **DDD** hasta que lleguemos a la siguiente linea.



En dicha linea como podemos apreciar, se realiza la siguiente operación **cmp %eax,%edx** donde **%eax** es el valor de la clave de digitos CORRECTA y **%edx** es el valor de la clave de digitos introducida, en este caso **7766**.

Con todo esto podriamos llegar a desactivar nuestra bomba sin ningun problema.