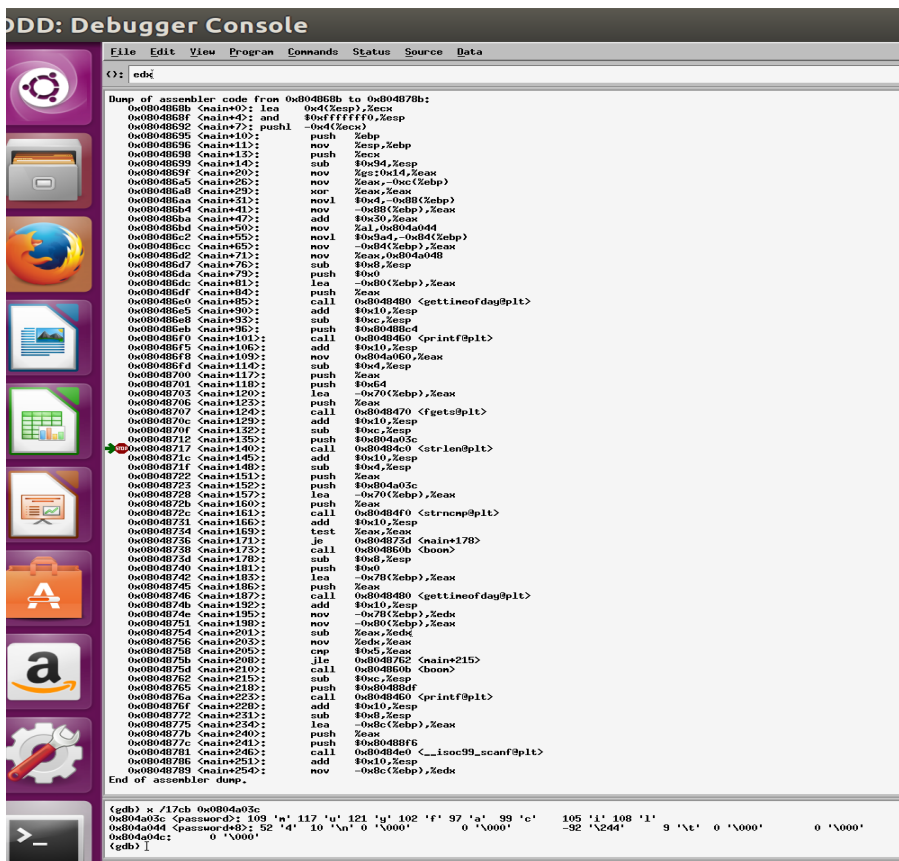


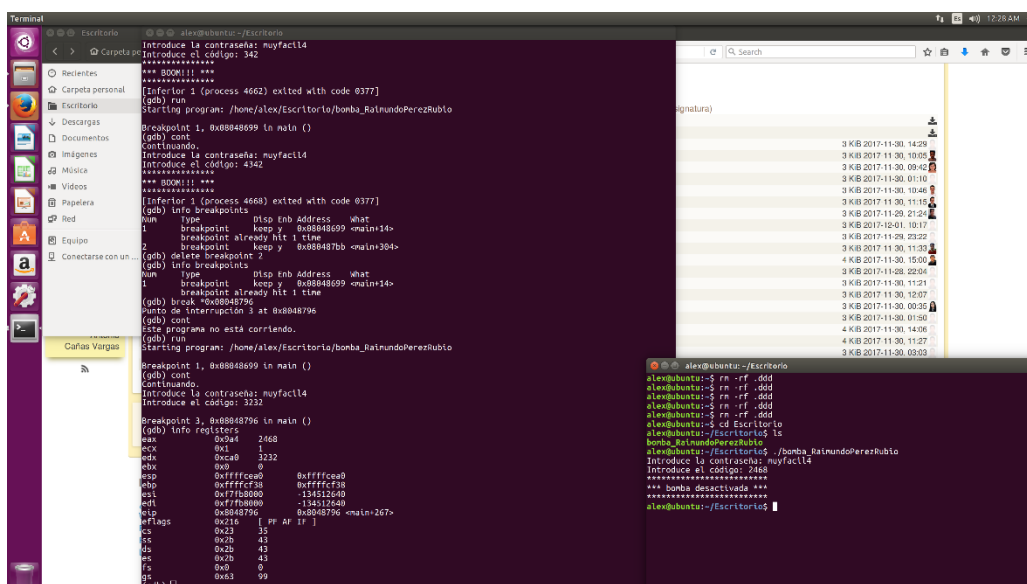
Práctica 4 Alejandro Poyatos López Segunda Parte

Resolución de las bombas de los alumnos de la clase

He resuelto la bomba del compañero Raimundo Perez Rubio, siguiendo los siguientes pasos, para comenzar he observado que no existe codificación externa a la del main, por lo que lo he abierto con ddd y he colocado un breakpoint justo después de la comparación de strlen como se observa en la siguiente foto:



Y observando los datos de tipo char de la dirección que hace push inmediatamente antes sacamos la password directamente: muyfacil4



Por otro lado para para la cifra numérica he usado gdb y he colocado un breakpoint en el salto antes de la segunda llamada a boom para ver el valor del registro eax:

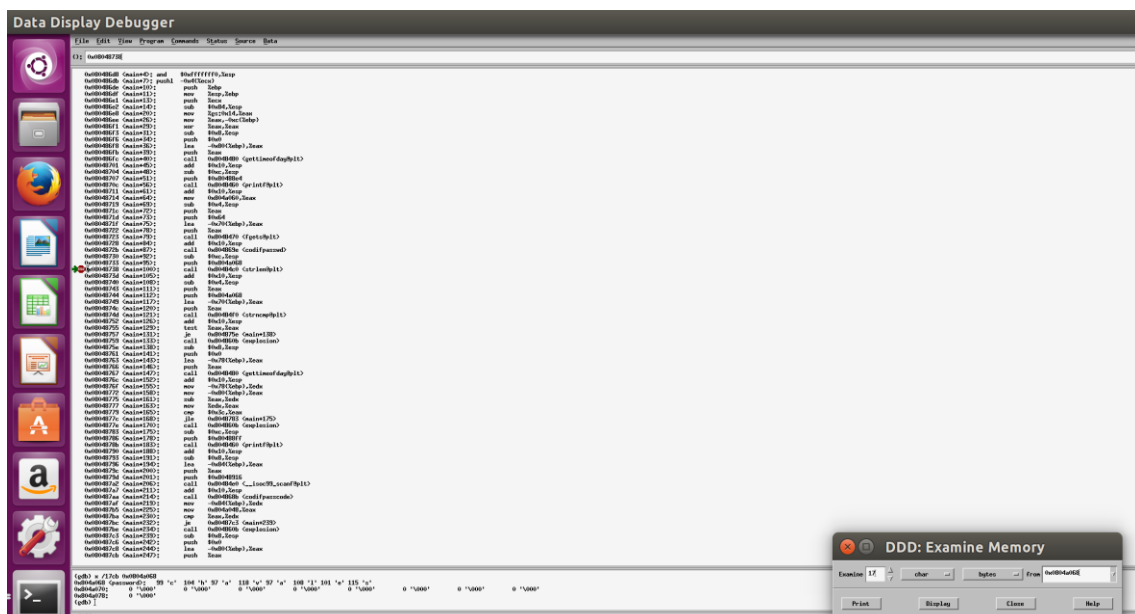
Como podemos observar el código esta a simple vista 2468, por lo que ya esta resuelta la bomba.

La segunda bomba es la del compañero David Carrasco Chicharro, que tiene dos funciones externas al main para codificar la password y el passcode:

```
0804868b <codifpasscode>:
804868b: 55                push    %ebp
804868c: 89 e5            mov     %esp,%ebp
804868e: a1 48 a0 04 08   mov     0x804a048,%eax
8048693: c1 f8 02        sar     $0x2,%eax
8048696: a3 48 a0 04 08   mov     %eax,0x804a048
804869b: 90              nop
804869c: 5d              pop     %ebp
804869d: c3              ret

0804869e <codifpasswd>:
804869e: 55                push    %ebp
804869f: 89 e5            mov     %esp,%ebp
80486a1: 83 ec 10        sub     $0x10,%esp
80486a4: c7 45 fc 00 00 00 00 movl    $0x0,-0x4(%ebp)
80486ab: eb 1e          jmp     80486cb <codifpasswd+0x2d>
80486ad: 8b 45 fc        mov     -0x4(%ebp),%eax
80486b0: 05 3c a0 04 08   add     $0x804a03c,%eax
80486b5: 0f b6 00        movzbl (%eax),%eax
80486b8: 83 e8 02        sub     $0x2,%eax
80486bb: 89 c2          mov     %eax,%edx
80486bd: 8b 45 fc        mov     -0x4(%ebp),%eax
80486c0: 05 68 a0 04 08   add     $0x804a068,%eax
80486c5: 88 10          mov     %dl,(%eax)
80486c7: 83 45 fc 01     addl    $0x1,-0x4(%ebp)
80486cb: 83 7d fc 07     cmpl    $0x7,-0x4(%ebp)
80486cf: 7e dc          jle     80486ad <codifpasswd+0xf>
80486d1: 90              nop
80486d2: c9              leave   %ebp
80486d3: c3              ret
```

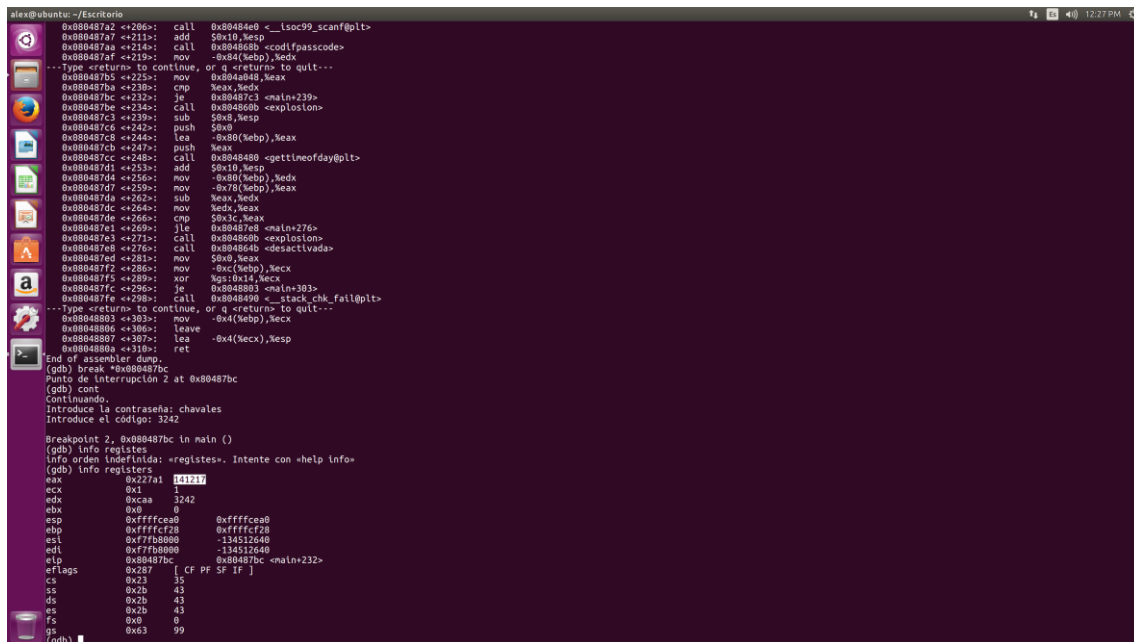
Aunque no me ha hecho falta descifrar el funcionamiento de ambos, ya que he conseguido averiguar la password de vuelta como se muestra en la siguiente imagen:



Obteniendo ya la password descodificada: chavales

La compruebo veo que funciona y continuo hacia el passcode.

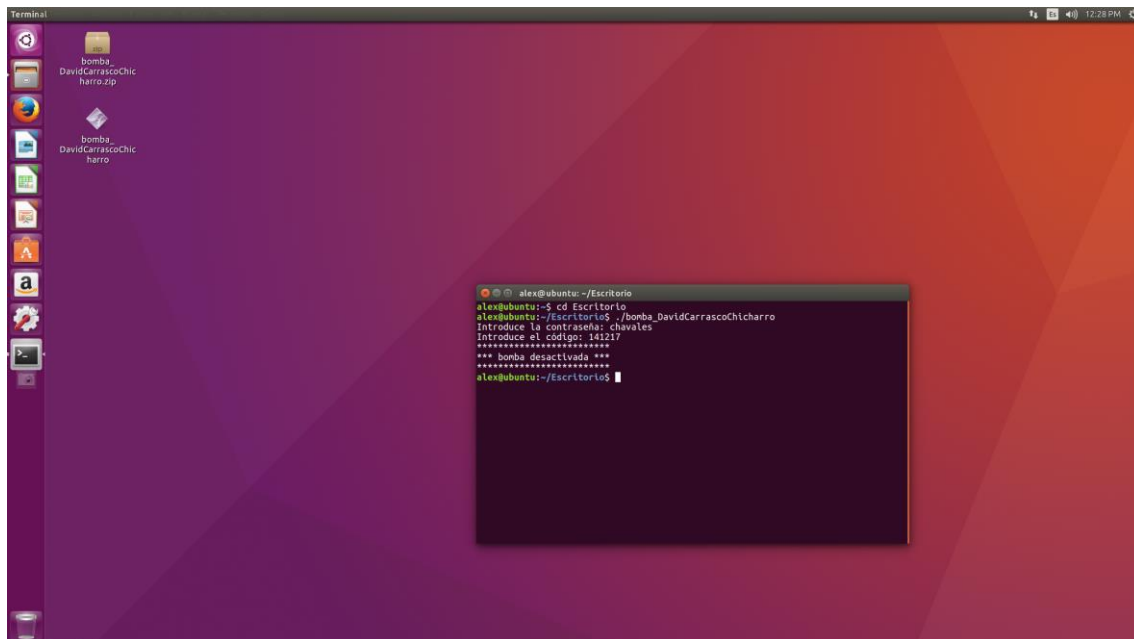
Para el passcode me situo con gdb en la última comparación antes de la llamada a la explosión, observando que se comparan dos registros el eax y el edx, siendo el edx el que yo introduje, observo que el eax contiene el passcode descifrado:



```
alex@ubuntu: ~/Escritorio
0x080487a2 <<206>: call 0x080484e0 <_Isoc99_scanf@plt>
0x080487a7 <<211>: add $0x10,%esp
0x080487aa <<214>: call 0x0804865b <codifpasscode>
0x080487af <<219>: mov -0x84(%ebp),%edx
---Type <return> to continue, or q <return> to quit---
0x080487b5 <<225>: mov 0x0804a040,%eax
0x080487b6 <<226>: cmp %eax,%edx
0x080487bc <<232>: je 0x080487c3 <main+239>
0x080487bd <<234>: call 0x08048600 <explosion>
0x080487c3 <<239>: sub $0x9,%esp
0x080487c6 <<242>: push $0x0
0x080487c8 <<244>: lea -0x80(%ebp),%eax
0x080487cb <<247>: push %eax
0x080487cc <<248>: call 0x08048480 <gettimeofday@plt>
0x080487d1 <<253>: add $0x19,%esp
0x080487d4 <<256>: mov -0x80(%ebp),%edx
0x080487d7 <<259>: mov -0x78(%ebp),%eax
0x080487d8 <<262>: sub %eax,%edx
0x080487dc <<264>: mov %edx,%eax
0x080487de <<266>: cmp $0x3c,%eax
0x080487e1 <<269>: jle 0x080487e8 <main+276>
0x080487e3 <<271>: call 0x0804860b <explosion>
0x080487e5 <<276>: call 0x0804864b <desactivada>
0x080487ed <<281>: mov $0x9,%eax
0x080487f2 <<286>: mov -0xc(%ebp),%ecx
0x080487f5 <<289>: xor %ecx,%ecx
0x080487fc <<296>: je 0x08048803 <main+303>
0x080487fe <<298>: call 0x08048490 <_stack_chk_fail@plt>
---Type <return> to continue, or q <return> to quit---
0x08048803 <<303>: mov -0x4(%ebp),%ecx
0x08048806 <<306>: leave
0x08048807 <<307>: lea -0x4(%ecx),%esp
0x0804880a <<310>: ret
End of assembler dump.
(gdb) break *0x080487bc
Punto de interrupción 2 at 0x080487bc
(gdb) cont
Continuando.
Introduce la contraseña: chavales
Introduce el código: 3242

Breakpoint 2, 0x080487bc in main ()
(gdb) info registers
Info orden indefinida: «registers». Intente con «help info»
(gdb) info registers
eax             0x227a1 141217
ecx             0x1      1
edx             0xcaa 3242
ebx             0x0      0
esp             0xffffca0 0xffffca0
ebp             0xffffcf28 0xffffcf28
esi             0xf7fb0000 -134512640
edi             0xf7fb0000 -134512640
eip             0x080487bc 0x080487bc <main+232>
eflags          0x287 [ CF PF SF IF ]
cs              0x23 35
ss              0x2b 43
ds              0x2b 43
es              0x2b 43
fs              0x0      0
gs              0x0      0
(gdb)
```

Compruebo el código y veo que ya esta descifrada:



```
alex@ubuntu: ~/Escritorio
alex@ubuntu:~$ cd Escritorio
alex@ubuntu:~/Escritorio$ ./bomba_DavidCarrascoChicharro
Introduce la contraseña: chavales
Introduce el código: 141217
*****
*** bomba desactivada ***
*****
alex@ubuntu:~/Escritorio$
```