# CARLOS MORALES AGUILERA
# PRÁCTICA 4. - BOMBA DIGITAL, DESENSAMBLADORES

## 4.1 PROGRAMAR LA BOMBA DIGITAL

```
// Carlos Morales Aguilera - 2ºB - B3
// Orden de compilacion usada:	gcc -m32 -O2 bomba.c -o bomba

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/time.h>

//char password[] = "percebe\n";	// Contraseña de la bomba -> ASCII = 112 101 114 99 101 98 101
char password[] = {115,104,117,102,104,101,104,'\n'+3};  // ASCII + 3

int  clave_numerica = 2708;		// Clave de la bomba

// Explotar Bomba
void Explode(){
	printf(  "##########################################\n"
		"#                                        #\n"
	"#             ¡BOOOOOOOOM!               #\n"
		"#                                        #\n"
	"##########################################\n\n");
	exit(-1);
}

// Desactivar Bomba
void Defuse(){
	printf(  "##########################################################\n"
	"# ¡Enhorabuena! Has conseguido desactivar mi bomba.  #\n"
	"##########################################################\n\n");
	exit(0);
}

// Comprobacion de passwor
void Comprobar_password(char *password_proporcionada){

	char password_cifrada[strlen(password)];
	int i=0;

	if (strlen(password_proporcionada) != strlen(password))
		Explode();

	// Rellenamos un vector de caracteres auxiliar con los caracteres de la cadena introducida por teclado
	// a los que le aplicamos el mismo incremento de valor que a la cadena clave.
	while(i < strlen(password)){
		password_cifrada[i] = password_proporcionada[i]+3;
```

```
                i++;
        }


        if (strncmp(password_cifrada,password,strlen(password)))
                Explode();

}

// Comprobacion de clave
void Comprobar_clave(int clave_proporcionada){
        if (clave_proporcionada != clave_numerica)
                Explode();
}

#define SIZE 100
#define TAM_LIM 60

int main(){
        // Declaracion de datos
        char password_proporcionada[SIZE];
        int  clave_proporcionada;
        struct timeval tv1,tv2;

        gettimeofday(&tv1,NULL);

        // Obtencion de datos
        printf("Introduce la password: ");
        fgets(password_proporcionada,SIZE,stdin);

        // Comprobacion de password
        Comprobar_password(password_proporcionada);

        gettimeofday(&tv2,NULL);
        if (tv2.tv_sec - tv1.tv_sec > TAM_LIM)
           Explode();

        // Obtencion de datos
        printf("Introduce el codigo: ");
        scanf("%i",&clave_proporcionada);

        // Comprobacion de clave
        Comprobar_clave(clave_proporcionada);

        gettimeofday(&tv1,NULL);
        if (tv1.tv_sec - tv2.tv_sec > TAM_LIM)
           Explode();

        // Desactivacion de la bomba
        Defuse();
}
```

## 4.2 MÉTODO PARA DESACTIVAR LA BOMBA: GDB DEPURADOR

**- CONTRASEÑA**

Puedo verlo también, utilizando el depurador, (gdb), los pasos que sigo son los siguientes:

1. **gdb ./bomba** ( entro en gdb con el ejecutable )

2. **break main ----> run -----> disass**

3. **break *** <direccion anterior a encriptar con Cifrar_Password>

4. **print (char *)(($eax))** //Así imprimo mi cadena, es decir, la que va a entrar a la cadena.

5. Yo he introducido abcd, así que gdb me muestra: **$1 = 0xffffd5b8 "aaaaa"**

6. Hago **disass**, y busco la línea en la que se hace la comparación, en mi caso:
0x0804876b <+139>: call 0x8048500 <strncmp@plt>

7. Pongo un **break** en la instrucción inmediatamente anterior a esa operación, y vuelvo a imprimir la cadena **print (char *)(($eax))** y me muestra: **$2 = 0xffffd5b8 "ddddd"**

La cadena obtenida, es el resultado de sumarle tres a la que yo introducí por pantalla.

Busco con ghex mi cadena cifrada y la encuentro “shufheh”, que realizando la transformación inversa obtengo “percebe”.

**- CÓDIGO NUMÉRICO**

Continúo con el código numérico. Se pasa el parámetro a %eax, (es decir, la cifra numérica).

Me ayudo de gdb. Para ello, sigo los siguientes pasos.

1. **gdb ./bomba** ( entro en gdb con el ejecutable )

2. **break main ----> run -----> disass**

3. **break *** <direccion anterior a encriptar con Cifrar_Passcode>

4. **print (($eax))** //Así imprimo mi numero introducido.

5. Yo he introducido 123456, así que gdb me muestra: **$1 = 123456**

6. Hago **disass**, y busco la línea en la que se hace la comparación.

**7. Pongo un break en la instrucción inmediatamente posterior a esa operación, en mi caso:**
0x0804870e <+14>: je 0x8048715 <_Cifrar_Passcode+21>
**y vuelvo a imprimir print (($eax)) y me muestra: $2 = 2708 , que es mi código secreto.**