The example code is
(This code has been copied from the net and the only modification is in the test function)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

void bof(char * input)
{
   int random=50;
   char buf[20];
   strcpy(buf, input);// overflow

   printf("Hello %s\n", buf);
}

void test()
{
   printf("**** Passing the address of this function ****\n");
   system("sudo /bin/bash");
}

int main(int argc, char ** argv)
{
   int input;
   if (argc != 2)
   {
      printf("Run prog with one argument...\n ");
      return -1;
   }
   printf("[*] Enter 0\n");
   scanf("%d", &input);
   if(input == admin_pin){
      admin();
   } else foo(argv[1]);

   return 0;
}
```

1. Compile this gcc -fno-stack-protector -g <name.c> -o <output>
2. Once compiled execute the program (./name `echo -ne “AAAABBBBCCCCDDDDEEEEFFFFGGGGHHHHIIIIJJJJKKKKLLLLMMMM”`)
3. In the other terminal gdb -p <pid>
4. Info reg in GDB will give you at which character the eip is. Refer to this (https://www.asciitable.com)
5. If you then do a disas of the test function the ebp value is the address of the function.
6. Then we run (./name `echo -ne “AAAABBBBCCCCDDDDEEEEFFFFGGGGHHHHIIIIJJJJKKKKLLLLMMMM”`) and precede the ASCII character from step 4 with the address from step 5 in reverse order.

7. This is a way of executing a function by overflowing the EIP with the address of your choice.