Project 1 – CS-6035
Magahet Mendiola

Task 1

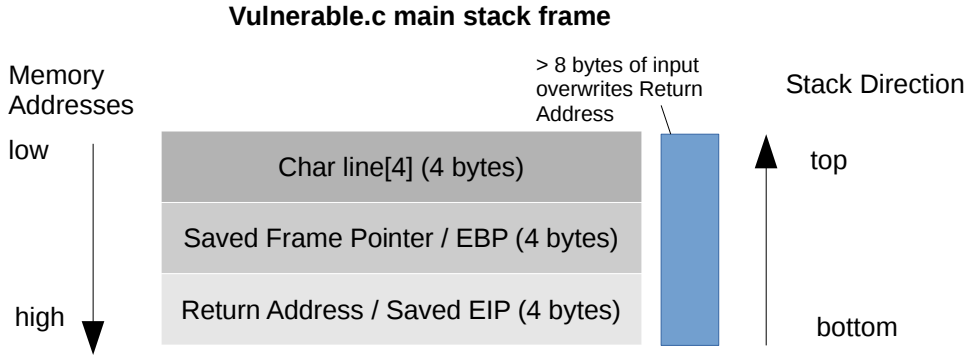
1. Vulnerable Program

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

/*
 * a vulnerable program
 */

int main()
{
  char line[4];
  printf("\nEnter your name fool: ");
  gets(line);
  printf("\nHa! I have you now, %s\n", line);
  return 0;
}
```

2. Stack Layout



3. Exploiting Explanation

The main function of vulnerable.c allocates 4 bytes to a local variable, *line*. Line is passed to the gets() function, which will read from stdin into *line*. Since no bounds checking is done, the input can exceed the allocated 4 bytes, and gets() will happily continue to write data to memory, overwriting any values lower on the stack. If this data is large enough (> 8 bytes in this case), it will overwrite the return address. This will cause the function to return control to an unexpected memory location. If we customize this location, it would be possible to cause our program to execute malicious code.