1. **An explanation of what happens in the stack when a function is called, including a description of purposes of the rbp,rsp, and rip registers as used in the Intel x86 architecture.**

Whenever a function is called, the stack evaluates and pushes parameters onto the stack, it then pushes return addresses onto the stack, and then it branches into the destination function. Rap is the frame pointer for x86. it keeps local variables and function parameters from the rsp register so that whenever rsp is changed, the offset rbp still contains those values.

1. **A short description of what (in general) happens during a buffer overflow attack.**

**An explanation of why the code used to take input in bof.c is dangerous, and**

**what procedure should be used instead.**

1. **An explanation of why the code used to take input in bof.c is dangerous, and**

**what procedure should be used instead.**

1. **A list of commands you used in gdb in order to examine the stack, as well as their**

**output at each step.**

1. **A copy of the input you provided to the program in order to perform the exploit.**

