26. Write a program that contains a buffer overflow that can be meaningfully exploited without modifying the execution of the program. [It should be like the gamble.c example in class.]

Include:

• The source code for the program,
• The executable for the program,
• An explanation of the buffer overflow condition, and
• A demonstration of how the buffer overflow results in a program compromise.

The overflow condition should not be immediately obvious to someone reading the source code.

27. What are the limits for the common integer types on our system?

28. Give an example of a program that contains an integer overflow error, and explain the problem. The program’s flaw should not be immediately apparent.

Correct the flaw by implementing appropriate range-checking. Do not change the types (int, unsigned int, etc.) of any of the variables.

29. Repeat the previous problem, but with a sign error.

30. Repeat the previous problem, but with a truncation error.

31. Write a program that contains an integer overflow that can be meaningfully exploited in some way.

Include:

• The source code for the program,
• The executable for the program,
• An explanation of the buffer overflow condition, and
• A demonstration of how to exploit the program.

The overflow condition should not be immediately obvious to someone reading the source code.