Prob 1. The order is:

NOP sled
shellcode (i.e., the exploit)
malicious return address

Prob 2. Choose two from:

**The non-executable stack.** This approach forbids the operating system from executing instructions that are on the stack. Basically, with this approach, the eip register would never be permitted to hold an address that is in the stack's address range. Machine language instructions would not ordinarily be found on the stack (the machine language instructions would be in the text section), so there is no reason for the eip to ever point to the stack's region in memory.

**The canary.** This approach entails placing a specific known value on the stack just prior to the return address. This known value is termed a canary (since a canary was used in coal mines to provide an advance indication of danger). An attempt to overwrite the return address will necessarily overwrite the canary. Before a function returns, the canary is checked to see if it has been altered. If the canary has been altered, the program is halted.

**Address Space Layout Randomization (ASLR).** In this technique, the stack and the heap are placed in random memory locations, preventing the hacker from easily predicting the location of the return address. Of course the locations of the stack and the heap are not completely random, but are usually arranged according to a fixed number of possible options.

Prob 3. The buffer overflow exploit discussed in this chapter results in the user opening a root-user shell. The person running this exploit has escalated his privileges—becoming the root user—and can now do all that the root user could do (e.g., read anyone's files, delete files, install malware, etc.)