Homework 3

Please submit your solutions to the email address 7ccsmsen at gmail dot com. Please submit only one homework per email. Please also submit only ASCII text or PDFs. Every solution should be preceded by the corresponding question, like:

Qn:	**...a difficult question from me...**
A:	**...an answer from you ...**
Q$n+1$	**...another difficult question...**
A:	**...another brilliant answer from you...**

Solutions will only be accepted until 20th December!

1. How does a buffer-overflow attack work? (Hint: What happens on the stack.)
2. Why is it crucial for a buffer overflow attack that the stack grows from higher addresses to lower ones?
3. What does it mean for the stack to be executable and why is this important for a buffer overflow attack?
4. If the attacker uses a buffer overflow attack in order to inject code, why can this code not contain any zero bytes?
5. How does a stack canary help with preventing a buffer-overflow attack?
6. Why does randomising the addresses from where programs are run help defending against buffer overflow attacks?
7. What is a format string attack?
8. Assume format string attacks allow you to read out the stack. What can you do with this information? (Hint: Consider what is stored in the stack.)
9. Assume you can crash a program remotely. Why is this a problem?
10. How can the choice of a programming language help with buffer overflow attacks? (Hint: Why are C-programs prone to such attacks, but not Java programs.)
11. When filling the buffer that is attacked with a payload (starting a shell), what is the purpose of padding the string at the beginning with NOP-instructions.
12. **(Optional)** This question is for you to provide regular feedback to me, for example what were the most interesting, least interesting, or confusing parts in this lecture? Is there anything you like to have improved or explained in the handouts? Please feel free to share any other questions or concerns.