Operating Systems, Spring 2020, Exercise 6

1. (Review question 15.6-8 [Sta18], modified)
    a. What are the two broad categories of defences against buffer overflows?
    b. List and briefly describe some of the defences against buffer overflows that can be used when compiling new programs?
    c. List and briefly describe some of the defences against buffer overflows that can be used when running existing, vulnerable programs?
    d. Why do you need run time defences against vulnerable programs and not just recompile all code?

2. (Problem 15.3 [Sta18])
   Rewrite the function shown in Fig 15.1a (slide 24 Ch 15) so it is no longer vulnerable to stack buffer overflows. You might want to use C-functions (operators) *fgets()* and s*izeof()*.

3. Assume that UNIX passwords are limited to the use of the 95 printable ASCII characters and that all passwords are 10 characters in length.
    a. Assume a password cracker with a 4-core system and an encryption rate of 6 million encryptions per second in each core. How long will it take to test exhaustively all possible passwords?
    b. What if the intruder has a 1000 node cluster with 8 cores (same speed as above) in each cluster? How long now?
    c. What if each cluster node would have a 2000 core (same speed as above) graphics card to crack your password? How long now?
    d. What if the attacker knows that the user is lazy and uses only (upper and lower case) letters in his password? How long now (with cluster and graphics cards)?

4. Unix security. Assume that Tiina is collecting lots of personal information concerning her friends and saving that data in file People. Tiina has written program Snoop, which can be used to look at some of the data in People. File People should not be accessible to anyone, except Tiina's colleagues (in group Staff) when they use Snoop to access it.
   What kind of access controls in Unix system should Tiina set up for files People and Snoop?
   How do you prevent Tiina's colleagues from modifying file People, when accessing it?

5. Windows 7 security. Pekka and Ville (both in group Student) are together writing a report, in file Study. Pekka wants all students, but not Liisa, to be able to read his report. All staff (in group Staff) must be also able to read Study, but instructor Matti (staff) can also write his comments in the Study.
    i. Describe the access tokens for the processes/threads created by Pekka, Ville, Matti, and Liisa?
    ii. Describe the security descriptor for file Study. ACE header specifies whether this entry allows or denies access. ACE general structure is {Allow/Deny, read/write/exec, user/group}

6. (Problem 15.3 [Sta18])
   In the traditional UNIX file access model, UNIX systems provide a default setting for newly created files and directories, which the owner may later change. The default is typically full access for the owner combined with one of the following: no access for group and other, read/execute access for group and none for other, or read/execute access for both group and other. Briefly discuss the advantages and disadvantages of each of these cases, including an example of a type of organization where each would be appropriate.

7. Course feedback. Please fill in the course feedback form in WebOodi. If you used the streamed or stored lectures during course, please give some feedback on them in the comment field at the end of the form. How did you use the online lectures? Did you find them useful? What could be done

better?