LiTH, Linköpings tekniska högskola
IDA, Institutionen för datavetenskap
Nahid Shahmehri

Written exam

TDDC90 Software Security

2008-03-28

Permissible aids

Dictionary (printed, NOT electronic)

Teacher on duty

David Byers, 013-282821, 0708-282821

Instructions

The exam is divided into two parts.

There are eight questions in part one. You should answer all of them.

There are four questions in part two. You should answer only one of them. If you fail to follow this instruction and answer more than one question in part two, we will randomly choose one of your answers to grade and disregard the others.

You can get a maximum of 32 points in part one and 6 points in part two, giving a maximum total of 38 points on the exam.

You may answer in Swedish or English.

Grading

Your grade will depend on the total number of points you accumulate on the exam.

The following grading scale is preliminary. It might be adjusted during grading.

Grade	3	4	5
Points required	20	27	33

Part one

Question 1: Common criteria (2 points)

Explain what a protection profile (in the context of the common criteria) is.

Question 2: Static analysis (2 points)

Explain what soundness means in the context of static analysis.

Question 3: System call wrappers (2 points)

System call wrappers are sometimes used to enforce security policies in operating systems. In 2007 a problem with the implementation of many system call wrappers was discovered. Explain this problem and its consequences.

Question 4: Developing secure software (2 points)

Is it possible for a program to have serious security flaws despite the implementation being absolutely flawless? Motivate your answer.

Question 5: SQUARE (6 points)

Briefly explain the Security Quality Requirements Engineering (SQARE) methodology.

Question 6: Fuzz testing (6 points)

Explain what fuzz testing is and what the main challenges are to fuzz testing. Explain why these are challenges to fuzz testing, why they are important and outline how they might be overcome.

Question 7: Preventing exploits (6 points)

ProPolice (newer versions are known as SSP) includes the following three mechanisms that protect against buffer overflows: a canary is placed between the saved frame pointer and return address on the stack; local variables are arranged so that buffers are stored at higher addresses than pointers (i.e. pointers are higher on the stack than buffers); and function arguments are copied to the current stack frame from where they were placed by the caller.

(a) For each of the three mechanisms listed above explain how the mechanism protects against buffer overflows. Your answer must include a characterization of the buffer overflows each mechanism targets.

(b) Explain one attack that ProPolice does not protect against.

(c) Describe a protection mechanism that would prevent successful exploitation using the attack you describe in (b).

Question 8: RMF (6 points)

Explain the five stages of activities in a risk analysis framework (RMF). At what time in the software lifecycle is RMF used? Elaborate your answer with an example.

Part two

Important

Only answer one question in this section. If you answer more than one then we will choose one of your answers at random to grade and disregard the others.

Question 9: Security requirements (6 points)

Are security requirements functional or non-functional requirements? Motivate your answer and give examples.

Question 10: Protection against integer-based vulnerabilities (6 points)

RICH is a technique for automatically protecting against integer-based vulnerabilities. Explain what an integer-based vulnerability is, how they can be exploited and how RICH protects against such vulnerabilities.

Question 11: Instruction set randomization (6 points)

Instruction set randomization is a technique that potentially can prevent a large class of remote exploits. Explain what instruction set randomization is and what its advantages and drawbacks are. What kinds of problems does instruction set randomization protect against? Are there any important classes of vulnerabilities that it *doesn't* protect against? If so, which ones?

Question 12: Security processes and best practices (6 points)

The Sustainable Software Security Process (S^3P), CLASP and the Secure Development Lifecycle (SDL) are examples of approaches to secure software development. Choose two of these approaches and explain how they work and how they are related to security best practices.