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

Written exam

TDDC90 Software Security

2009-04-17

Permissible aids

Dictionary (printed, NOT electronic)

Teacher on duty

Shanai Ardi, 013-282602, 0762-105806

Instructions

The exam is divided into two parts with a total of ten questions. You should answer all questions in all parts. In order to get the highest grade you will need sufficient points in the second part.

You may answer in Swedish or English.

Grading

Your grade will depend on the total points you score on the exam. The following grading scale is preliminary and might be adjusted during grading.

Grade	3	4	5
Points required	18	24	30

Important
In order to get the highest grade you must have scored at least six points in part 2.

Part one

Question 1: Developing secure software (2 points)

When developing secure software, is it sufficient to ensure that there are no flaws in the implementation, or is something else required? If so, what? Motivate your answer.

Question 2: Fuzz testing (2 points)

Which is better: fuzz testing or static analysis? Elaborate and motivate your answer.

Question 3: The Common Criteria (2 points)

Explain what a security target is and how it is used.

Question 4: Static analysis (2 points)

Explain what soundness means in the context of static analysis.

Question 5: Static analysis (4 points)

Static analysis is usually applied to source code, but can also be applied to executables. Give at least two examples of why it may be appropriate to analyze the executable rather than the source code, and explain each one in detail.

Question 6: Privilege separation (4 points)

Privilege separation is a proven and highly successful design that has prevented exploits of vulnerabilities in e.g. OpenSSH. Explain what privilege separation is, what the goals of privilege separation are, and how it prevents exploits. Show, using an example, how privilege separation works in a typical application (such as OpenSSH).

Question 7: Architectural risk analysis (4 points)

What are the critical steps in the architectural risk analysis (based on the "Touchpoint Process")? What kind of flaws can be discovered by each step?

Question 8: Security requirements (4 points)

What are the steps in the CLASP methodology to derive the security requirements, explain each step? How is the SMART+ requirements used in this methodology?

Part two

In order to score well on these questions you will need to show that you understand not only the technical issue or concept at hand, but also its context and its interactions with its context (e.g. processes, methods, techniques, technology, people, risks, threats, and so on). We *think* that good answers to these questions will require at least one or two handwritten pages (more or less may be required depending on how you write).

Question 9: Run-time exploit prevention (6 points)

PaX is a security add-on to Linux that incorporates a number of useful techniques. It implements something similar to W⊕X protection (which, among other things, results in a non-executable stack) and address space layout randomization (ASLR).

(a) On a platform with 32 bit addressing, the effectiveness of PaX is limited. Why? How can an attacker exploit this? Do the same limitations apply on platforms with 64 bit addressing? Motivate your answer.

(b) If there is an format string vulnerability that allows the attacker to examine the stack of the protected program, then PaX is quite easy to bypass. Explain how.

Question 10: Processes and capability levels (6 points)

The Secure Development Lifecycle (SDL) is one approach to secure software development. Explain how it works. What SSE-CMM capability level do you think this process can belong to? Motivate your answer.