

CS 6V81-002: Quiz 12 Solutions

March 5, 2008

1. Which of the following are examples of *covert channels*? (circle all that apply)
 - (a) The power consumption of a smart card fluctuates randomly as a pin number is entered.
 - (b) A login prompt takes 5 microseconds to reject incorrect passwords but only 2 microseconds to accept the correct password.
 - (c) A database program crashes when over 1000 records match a search query regardless of whether the user has permission to view the records.**
 - (d) An online store logs private credit card info to a public webpage.

Answer (a) is not a covert channel because if the power fluctuates randomly, the attacker learns nothing. Answer (b) is incorrect because the correctness of a password attempt is always revealed explicitly by the password-checker; it is not secret information. Answer (c) is a covert channel because the attacker learns information about the number of records matching the query. Answer (d) is a non-covert information flow.

2. Let h be a high-security variable and l be a low-security variable. Which of the following program fragments have implicit flows from high to low? (circle all that apply)
 - (a) if $h = 1$ then $l := l + 1$**
 - (b) if $l = 1$ then $h := h + 1$
 - (c) while $(h > l)$ $l := l + 1$**
 - (d) while $(h = l)$ skip

An attacker can learn something about h from observing l in both (a) and (c), making them the correct answers. Answer (d) leaks information through a termination channel but is not an implicit flow.

3. In the typing judgment $[pc] \vdash C$, the pc is... (circle one)
 - (a) a security label for detecting implicit flows**
 - (b) the address of the next instruction to be executed
 - (c) a mapping from variables to types
 - (d) a mapping from variables to security labels

4. State the property of *non-interference* informally in words:

Informally, a program satisfies non-interference if high-confidentiality data does not affect (i.e., does not “interfere” with) low-confidentiality data.

5. Leino and Joshi define a program C to be secure iff the property

$$\forall s \in S. \llbracket HH; C; HH \rrbracket_s \approx \llbracket C; HH \rrbracket_s$$

holds, where operation HH is (informally) defined as, “set h to an arbitrary value”. This definition is useful for detecting... (circle all that apply)

- (a) **explicit flows**
- (b) **implicit flows**
- (c) timing channels
- (d) **probabilistic channels**

The definition says that programs are secure if they satisfy non-interference across all possible runs. Thus, it relates not only to explicit and implicit flows but also to probabilistic channels.