Homework #4

Due: March 17, 2014 (*no late assignments accepted*) **Points:** 100

Questions

1. (*25 points*) Revisit the example for $x := y + z$ in Section 16.1.1. Assume that x does not exist in state s. Confirm that information flows from y and z to x by computing $H(y_s|x_t)$, $H(y_s)$, $H(z_s|x_t)$, and $H(z_s)$ and showing that $H(y_s|x_t) < H(y_s)$ and $H(z_s|x_t) < H(z_s)$ (*text*, problem 16.1)

2. (*25 points*) Let $L = (S_L, \leq_L)$ be a lattice. Define:

   (a) $S_{IL} = \{[a, b] \mid a, b \in S_L \wedge a \leq_L b\}$

   (b) $\leq_{IL} = \{([a_1, b_1], [a_2, b_2]) \mid a_1 \leq_L a_2 \wedge b_1 \leq_L b_2\}$

   (c) $lub_{IL}([a_1, b_1], [a_2, b_2]) = (lub_L(a_1, a_2), lub_L(b_1, b_2))$

   (d) $glb_{IL}([a_1, b_1], [a_2, b_2]) = (glb_L(a_1, a_2), glb_L(b_1, b_2))$

   Prove that the structure $IL = (S_{IL}, \leq_{IL})$ is a lattice. (*text*, problem 16.2, modified)

3. (*25 points*) In the Janus system, when the framework disallows a system call, the error code **EINTR** (interrupted system call) is returned.

   (a) When some programs have read or write system calls terminated with this error, they retry the calls. What problems might this create?

   (b) Why do you think the developers of Janus did not devise a new error code (say, **EJAN**) to indicate an unauthorized system call? Justify your answer.

   (*text*, problem 17.5, modified)

4. (*25 points*) Consider the rule of transitive confinement. Suppose a process needs to execute a subprocess in such a way that the child can access exactly two files, one only for reading and one only for writing.

   (a) Could capabilities be used to implement this? If so, how?

   (b) Could access control lists be used to implement this? If so, how?

   (*text*, problem 17.3)

Extra Credit

1. (*20 points*) A company develops a new security product using the agile programming[1] software development methodology. Programmers code, then test, then add more code, then test, and continue this iteration. Every day, they test the code base as a whole. The programmers work in pairs when writing code to ensure that at least two people review the code. The company does not adduce any additional evidence of assurance. How would you explain to the management of this company why their software is in fact not "high assurance" software? (*text*, problem 18.7, modified)

[1] In the book, this is called "extreme programming"