CS 6V81-002: Quiz 13 Solutions

March 17, 2008

1. JFlow enforces which security policies (excluding those already enforced by Java)? (circle all that apply)

   (a) memory and control-flow safety
   (b) security automata
   (c) data confidentiality
   (d) data availability

   Memory and control-flow safety are already enforced by Java. JFlow's type-system enforces confidentiality through an information flow analysis. Security automata and availability policies are not enforced.

2. A decentralized label model differs from a centralized model in that... (circle one)

   (a) each principal can have a different policy for shared data
   (b) each method can be type-checked separately instead of requiring a global analysis
   (c) the lattice of security labels is always finite
   (d) the security enforcement mechanism can be distributed across multiple machines that enforce the policy end-to-end

3. Consider two JFlow labels L_1 and L_2 satisfying $L_1 \sqsubseteq L_2$. Which of the following statements are guaranteed to be true? (circle all that apply)

   (a) L_2 is at least as permissive as L_1
   (b) L_2 is at least as restrictive as L_1
   (c) $(L_1 \sqcup L_2) \sqsubseteq L_1$
   (d) $L_2 \sqsubseteq (L_1 \sqcup L_2)$

4. Which of the following accurately characterizes the power of the JFlow type-checker? (circle one)

   (a) All labels are inferred statically without programmer assistance.
   (b) All labels are inferred statically, but some inferences require programmer-supplied typing annotations.
   (c) Some labels cannot be inferred statically, requiring dynamic checks.
   (d) Some labels cannot be inferred dynamically, resulting in premature termination (but no information leak).

5. Add labels to the following JFlow method prototype to indicate that principal p allows only q to read `y`, `y`'s label is at least as restrictive as `x`'s label, and parameter `lb` holds `x`'s label at runtime.

```
void mymethod(int{*lb} x, label lb, int{p:q;x} y)
```