

CS 6V81-002: Quiz 20 Solutions

April 14, 2008

1. The authors prove which of the following results? (circle all that apply)
 - (a) if the key set is finite, no obfuscator can simulate strong-typing
 - (b) if the key set is finite, no strong type-system can simulate obfuscation
 - (c) if the key set is infinite, no obfuscator can simulate strong-typing
 - (d) if the key set is infinite, no strong type-system can simulate obfuscation**

The major result of the paper is that although strong-typing can precisely simulate obfuscation with a finite key set, it cannot precisely simulate obfuscation with an infinite key-set.

2. An attack is defined as *resistible* if, when the attack is attempted... (circle one)
 - (a) a typing-error results at compile-time
 - (b) a typing-error results at runtime
 - (c) at least one diverse replicant is not compromised
 - (d) at least one diverse replicant has different observable behavior**

3. *Probabilistic type-checking* involves... (circle one)
 - (a) randomly type-checking some (but not all) diverse replicants
 - (b) making a value's type be the probability distribution of its possible values
 - (c) randomly allowing some passed runtime checks to terminate the program
 - (d) randomly allowing some failed runtime checks to continue execution**

4. Which of the following are true of statically-typed Toy-C? (circle all that apply)
 - (a) It satisfies memory-safety.
 - (b) It satisfies control-flow safety.
 - (c) The programmer indicates which variables are observable to the attacker by labeling them "observable".**
 - (d) The set of observable variables depends on the attacker model, not the program text.

Statically-typed Toy-C is meant to be pretty much like statically-typed C in that it is not memory or control-flow safe. However, to distinguish attacker-observable variables from non-observable variables in the model, the authors introduce type qualifier labels.

5. Type system T^{strg} is a strictly conservative approximation of obfuscator τ^{addr} because... (circle one)
 - (a) T^{strg} disallows some memory-reads permitted by τ^{addr}**

- (b) T^{strg} disallows some memory-writes permitted by τ^{addr}
- (c) T^{strg} prohibits all pointer arithmetic
- (d) T^{strg} prohibits pointer arithmetic that yields a pointer to an unallocated memory location

The problem with T^{strg} is that it disallows out-of-bounds memory-reads even if the resulting value never flows into an attacker-observable variable or output. This behavior differs from that of an obfuscator, meaning that T^{strg} is a strictly conservative approximation of obfuscator τ^{addr} .