

CS 6V81-002: Quiz 4 Solutions

January 30, 2008

1. CFI (without SMAC or other extensions) enforces which security policy?
 - (a) Every memory-write operation must correspond to some node in the CFG.
 - (b) Every memory-read operation must correspond to some node in the CFG.
 - (c) Every branch operation must correspond to some edge in the CFG.**
 - (d) Every pair of call and return operations must correspond to some node-edge pair in the CFG.
2. In CFI, an *ID* is...
 - (a) a set of high-order bits common to all valid jump destination addresses
 - (b) a unique integer assigned to each basic block
 - (c) a series of bytes inserted immediately before each valid jump destination**
 - (d) an integer passed to a callee to uniquely identify the caller
3. Which of the following are assumptions upon which CFI relies (circle all that apply)?
 - (a) The code segment is non-writable.**
 - (b) The data segment is non-executable.**
 - (c) One thread cannot change another thread's register contents.**
 - (d) Program behavior does not rely on status flags being preserved across jumps.**
4. A shadow call stack is useful for (circle all that apply)...
 - (a) prohibiting untrusted code from invoking a policy-violating system call
 - (b) preventing callees from returning to an address other than the callsite**
 - (c) protecting against buffer overrun attacks that modify the return address**
 - (d) securely computing unique ID's even if attackers can modify the data segment
5. The authors' CFI implementation in-lines code that computes ID 0x12345678 at run-time by first loading the constant 0x12345677 into a register and then incrementing it by 1. Why not just load 0x12345678 into the register directly, saving an instruction?

The byte-level encoding of an instruction that loads the constant 0x12345678 into a register contains the literal bytes 0x12345678. This would therefore introduce that ID into the code in a place other than a valid branch target, violating the UNQ assumption.