Deng, Chaoqiang; Namjoshi, Kedar S.

Securing the SSA transform. (English) [Zbl 1420.68055]


Summary: Modern optimizing compilers use the single static assignment (SSA) format for programs, as it simplifies program analysis and transformation. A source program is converted to an equivalent SSA form before it is optimized. The conversion may, however, create a less secure program if fresh SSA variables inadvertently leak sensitive values that are masked in the original program. This work defines a mechanism to restore a program to its original security level after it has been converted to SSA form and modified further by a series of optimizing transformations. The final program is converted out of SSA form by grouping variables together in a manner that blocks any new leak introduced by the initial SSA conversion. The grouping relies on taint and leakage information about the original program, which is propagated through the optimizing transformations using refinement proofs.

For the entire collection see [Zbl 1369.68036].

MSC:

68N20 Theory of compilers and interpreters

Full Text: DOI