Secure Programming via Safety Games

William R. Harris
University of Wisconsin, Madison

ABSTRACT

Writing secure programs remains an open, challenging, and important problem. However, new operating systems allow application programs to write secure programs with a tractable amount of effort. Such systems define a notion of *privilege* and provide a set of system calls, or *primitives*, that a program can invoke to manage its privilege and the privileges of other programs with which it interacts. Unfortunately, in practice it is difficult to rewrite a program to invoke primitives so that the program satisfies high-level security and functionality requirements [3].

In automata theory, *two-player turn-based safety games* generalize traditional automata. A traditional automaton is a state machine that reads a sequence of actions from a single agent, *the environment*, and accepts the sequence if it drives the automaton to an accepting state. Many problems in analyzing programs can be reduced to reachability problems for automata: given an automaton, is there some sequence of actions that the automaton accepts? A two-player turn-based safety game is an automaton that reads a sequence of actions in alternation (i.e., a *play*) from two adversarial agents, called the *Attacker* and *Defender*. The Attacker *wins* a play if the play drives the game to an accepting state; otherwise, the Defender wins the play. Given a two-player game, one natural problem is to decide if there is some *winning strategy* that the Defender can always follow to generate only winning plays [1].

This paper references the recent work "Secure Programming via Visibly Pushdown Safety Games" [2].

BODY

To write a correct and secure program, one can find a winning Defender strategy for a two-player turn-based safety game.

REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *FOCS*, 1997.

[2] W. R. Harris, S. Jha, and T. W. Reps. Secure programming via visibly pushdown safety games. In *CAV*, 2012.

[3] R. N. M. Watson, J. Anderson, B. Laurie, and K. Kennaway. Capsicum: Practical capabilities for UNIX. In *USENIX Security*, 2010.

Volume 1 of Tiny Transactions on Computer Science