Research System Controls the Effects of Software Bugs

Scientists at a Swiss research institution have created a prototype tool that keeps software bugs from causing system deadlocks.

The Dimmunix tool (http://dslab.epfl.ch/proj/dimmunix), which a team at the Ecole Polytechnique Fédérale de Lausanne's Dependable Systems Lab developed, lets individual machines or networks of computers avoid the deadlocks that many bugs can cause.

Researcher George Candea, assistant professor and director of EPFL's Dependable Systems Lab, said the tool's novelty is its ability to help programs build immunity to deadlocks without intervention from programmers or users.

Dimmunix doesn't require a central server. Instead, users install it on individual computers. The research team has built a prototype that it has verified to work on Linux machines. However, Candea said, their technique is platform-independent and could even run on mobile devices.

On an individual computer, the tool starts working after the first time a bug causes a deadlock.

On an ongoing basis, Dimmunix observes the method calls that program threads make and the order in which they occur.

The system looks for specific method-call sequences that end with the host machine deadlocking. Dimmunix then creates and stores as a signature the method calls and program-execution processes that led to the problem.

By watching a program's method-call sequences, Dimmunix recognizes when the bug is about to cause a deadlock again. It then changes the sequence in which threads run, to avoid conflicts that can deadlock a system.

In a network, Candea said, administrators can install copies of Dimmunix on each machine. The copies communicate and exchange bug signatures so that the host computers can be inoculated against deadlocks, even those machines that haven't experienced the problem yet.

So far, the researchers have shown Dimmunix to be effective on several systems, including the MySQL and SQLite databases, the JBoss application server, the Apache ActiveMQ message broker, the Limewire peer-to-peer file-sharing client, and the Java Development Kit.

Candea said his team has built Dimmunix to work with applications written in Java, C, and C++ but that the technique could work with almost any programming language.

Typically, developers have used static analysis to detect a propensity for deadlocks in code before it runs, noted Bielefeld University professor Peter Bernard Ladkin. This analysis runs concurrently with software to detect when deadlocks appear likely to happen and use various techniques to try to keep the problem from occurring, he explained.

Dimmunix's key weakness, Candea said, is that a machine somewhere in a system must experience a bug-related deadlock before the tool can establish defenses.

Also, he noted, the approach doesn't work with deterministic bugs, which always cause crashes under specific circumstances regardless of thread schedules. However, he added, these bugs are generally easy to fix.

The EPFL group plans to release Dimmunix as an open source application.

News Briefs written by ***Linda Dailey Paulson****, a freelance technology writer based in Portland, Oregon. Contact her at ldpaulson@yahoo.com.*

Editor: Lee Garber, *Computer*; l.garber@computer.org