発表概要

コンパイラによるLinux向け メモリ保護ドメインの利用支援

池田貴広[†1]　千葉雄司[†1]　土居範久[†1]

メモリを不正に書き換えるバグは，しばしば原因究明が困難なクラッシュを引き起こす．なぜなら，メモリの不正な書き換えからクラッシュまでに長い時間が経過することがあり，結果として，クラッシュ時のメモリダンプを見ても，書き換え元がどこか分からなくなってしまいうるからである．この問題を解決する手段に，メモリ保護を使う方法がある．この方法では，書き換えるべきでないメモリを書き込み禁止に保護することで，バグが不正な書き換えを試みた瞬間に，プログラムの実行をクラッシュさせ，書き換え元の所在を明らかにする．ただ，この方法を利用するには，プログラムのどの処理の実行時点で，どのメモリを書き込み禁止にするか定めるコードを，プログラム中の適切な箇所に挿入する必要があるが，挿入の作業を手動で行うには手間がかかる．そこで我々は，この手間の軽減を目的として，挿入の作業を自動的に行うC/C++コンパイラを開発した．開発したコンパイラは，プログラムモジュールに固有なヒープの作成を支援するもので，プログラムモジュールの出入口にヒープへの読書権限を確保/放棄するコードを挿入する．挿入箇所を最適化する機能を実現，評価したところ，挿入箇所を 7.9〜79.2%削減できることが分かった．

Compiler Support of Memory Protection Domain for Linux

Takahiro Ikeda,[†1] Yuji Chiba[†1] and Norihisa Doi[†1]

A bug that invalidly overwrites memory often leads to a hard-to-debug crash, because the overwriting does not necessarily cause the crash immediately and then clues to the bug disappear as the time passes. It is often the case that we can find little clue in the memory dump at the crashed time. One solution to this problem is to protect the memory when it should not be overwritten so that the overwriting cause the crash immediately, but we cannot use this solution easily because it is troublesome to insert code to protect the memory properly along the whole program. To save the programmer the touble of inserting the code, we have implemented a C/C++ compiler that automatically inserts the code. Our compiler helps a programmer create heap that is private to a program module by inserting the code at the entry or the exit of the program module. The compiler applies an optimization to eliminate the inserted code if it is found to be redundant. Our evaluation showed that the optimization eliminates 7.9 to 79.2% of the code.

（平成 20 年 8 月 6 日発表）

†1 中央大学大学院理工学研究科情報工学専攻
Information and System Engineering Course, Graduate School of Science and Engineering, Chuo University