発表概要

静的解析によるCプログラムのバッファオーバフロー検出

中村豪一[†] 村瀬一郎[†]

ネットワーク制御等高度なセキュリティが要求されるソフトウエアの開発言語としてCは依然として重要であるが，Cプログラムの最も深刻な脆弱性としてバッファオーバフローが問題になることが多い．バッファオーバフローとして重要なのはスタック上のリターンアドレス書き換えである．動的に（プログラム実行時に）このバッファオーバフローを検出するツールも開発されているが，プログラムに潜むバッファオーバフローを系統立って検出するものではないし，プログラム実行スピードの低下も招く．本研究では，Cプログラムを GCC によってコンパイルする過程で現れるレジスタ転送言語ソースを静的に解析してバッファオーバフローを検出するアルゴリズムを，通常のリブアナリシスで用いられる手法を基にしてまとめた．このアルゴリズムは，スタック上のリターンアドレス書き換えが起きる条件を関数引数等を使って表現したものを算出する．またアルゴリズムを実現したツールを開発し，バッファオーバフロー脆弱性を持つCプログラムに対するそのアルゴリズムの有効性を評価した．

Buffer-overflow Detection in C Program by Static Analysis

Goichi Nakamura[†] and Ichiro Murase[†]

C language is still important as a programming language of softwares such as network control software that needs high security. But the buffer-overflow problem is frequently seen in C programs, it is one of the most serious vulnerabilities about C programs. Among the buffer-overflow vulnerabilities, rewriting of return address on the stack is most important. There are several tools to detect this buffer-overflow vulnerability in C program dynamically (when C program is running). But these tools can not pick over this buffer-overflow vulnerability, and using these tools slow down C programs running. In this research, an algorithm is developed to detect the buffer-overflow vulnerability (rewriting of return address on the stack) in C program statically, that is, by static analysis (such as live-analysis) of register translate language source which is made in C program compilation by GCC compiler. As this algorithm output, the conditions on which rewriting of return address on the stack occurs is expressed in function arguments and so on. And the tool which carry out this algorithm is developed. Then, the effectiveness of this algorithm is evaluated by adapting the tool to C programs to detect the buffer-overflow vulnerability.

（平成 14 年 6 月 17 日発表）

† 株式会社三菱総合研究所
Mitsubishi Research Institute, Inc