安全なソフトウェア基盤の構築

米澤明憲

情報理工学系研究科コンピュータ科学専攻

概 要

現代のコンピュータにおいて、不正アクセスによる個人情報流出、サービス拒否攻撃による業務妨害、金融機関のシステム障害、個人用計算機の不安定さといった問題は、ほぼすべてソフトウェアの欠陥が原因である。我々はそのような欠陥を防止しやすい、安全なプログラミング言語について研究を推進している。

1 はじめに

一般に現代のコンピュータは、大きく分けてCPU、メモリ、ハードディスク等の**ハードウェア**と、OS、WWWブラウザ、メーラ、ワープロ等の**ソフトウェア**という二つの要素から構成される。電子機械技術の進歩により、現代のハードウェアは性能のみならず、信頼性・安定性の点においても向上が著しい。ところが、その上で動作すべきソフトウェアに関しては、設計開発技術の進歩が大規模化や多機能化に追従できず、その信頼性・安定性はむしろ低下の傾向が顕著である。たとえば、現在の一般的な家庭用ないし企業用の計算機において、アプリケーションが突如としてエラーを起こしデータが失われる、あるいは金融機関のシステムが誤動作して、誤った振り込みや引き落としが起こる、といった問題のほとんどは、ハードウェアではなくソフトウェアの欠陥が原因である。

こういったソフトウェアの欠陥は、不特定多数のコンピュータがインターネットのような広域分散ネットワークで結合され、それらを媒体として様々なデータやプログラムが交換・移動される今日の環境では、さらに重大となる。というのは、そのようなソフトウェアの欠陥を悪用することで、企業や個人の秘密情報を盗み出したり、場合によってはさらに他人のコンピュータを完全に乗っ取ってしまうことも可能なためである。実際にそのような被害が多発していることは、各種機関によって報告されている。

以上のような問題にたいして、一般に現在のコンピュータ産業界では、できるだけ欠陥が発生しないように人間が注意・確認し、それでも欠陥が発見されたらそのたびに修正するという、あまり根本的な解決策になっていない場当たり的なアプローチがとられることが多い。

上述のような状況を改善するために、我々はソフトウェアの開発においてもっとも重要な要素であるプログラミング言語 (コンピュータへの命令を記述するための言語) の様々な問題について、場当たり的な対症療法ではなく、堅固な理論にもとづく系統的な解決策を与えることを目指している。特に、ML, Scheme といった現代的な言語を様々な分野に適用するだけでなく、C, Java, Perl といった産業界で定着している古典的な言語についても、最新の理論を応用するための研究・開発を行っている。

以下に、本年度の代表的な研究について要約する。

2 安全なC言語処理系 [3, 12, 13]

C 言語は、1970 年代に提案された古典的なプログラミング言語の一つで、現在でも広範に使用されている。UNIX という代表的なオペレーティングシステムを実装するために開発されたことも

あって、もっとも原始的なプログラミング言語であるアセンブリ言語に近い、低水準の操作を詳細に記述することが可能である。

ところが、このようなC言語の特徴は一方でプログラマのミスを誘発しやすく、プログラムの複雑化ともあいまって、ソフトウェアの欠陥が多発する主要な原因の一つとなっている。アカデミックな分野ではMLやSchemeといった、より安全な言語も利用されているが、プログラマを教育するコストなどの社会的要因により、産業界で普及するにはいたっていない。

そこで本研究では、たとえプログラムに欠陥があっても「コンピュータを乗っ取られる」「データが壊される」といった致命的影響がない (= Fail-Safeな) C言語の実装方式を提案・実験した。具体的な方法は以下のとおりである。C言語の仕様には不十分な部分があり、多くの誤った操作の結果がエラーではなく「未定義」とされている (バッファ溢れや誤った型変換など)。前述のような致命的影響は、そのような未定義の操作の、実際の結果である場合がほとんどである。そこで、結果が未定義とされているような危険な操作を実行時検査により**すべて**検出し、致命的影響をおよぼす以前にプログラムを中断する、という方法である。

我々の実験によれば、上述のような実行時検査による速度低下は、単純な実現方式でも約1倍〜十数倍以下であった。また、我々とは独立な研究者らの実験によれば、事前にプログラムを解析して無駄な実行時検査を削減することにより、平均で数十%以下、最悪の場合でも2〜3倍程度に速度低下がおさえられたと報告されている。

本年度は、基本的な方式の提案とプロトタイプによる実験の結果を論文誌 [3] ならびに国際会議 [12] で発表するとともに、メールサーバ (sendmail ないし qmail) や Web サーバ (apache) といった大規模なアプリケーションプログラムを実行するための、本格的な実装に取り組んだ。また、オペレーティングシステムのシステムコールのような、外部のソフトウェアと我々のC言語処理系との相互運用性を向上するために、データ表現の変換や事前条件の検査を行うコード (stub) を半自動的に生成するインターフェース記述言語 (IDL) を設計・開発した [13]。

3 ユーザプログラムをカーネルモードで実行可能なLinux [1]

型検査等により安全性が保証されているユーザプログラムを、カーネルモードで実行できるLinux (オープンソースのUNIXの一種) を開発・公開 [1] した。この方式では、一般にハードウェア (CPUのメモリ管理ユニット) による実行時検査 (およびそれに伴うコンテキスト切り替え) の大半が不要となり、システムコールのオーバーヘッドが大幅に軽減する。基礎的な実験によれば、システムコールのオーバーヘッドが約1ミリ秒から、30ナノ秒程度に減少した。これにより、データベースやネットワークサーバといった、入出力の頻繁なアプリケーションを高速化できる。本年度は、より様々なプログラムで大規模な実験を行うために、標準ライブラリ (libc) をサポートする作業を行った。

4 文字列処理のための正規表現型 [6, 14]

Webサーバにおける CGI プログラムなどにおいては、文字列処理が重要な位置を占めることが多い。しかし、この処理は誤りが起きやすいにも関わらず、その正しさを確かめたり、バグを見つけるためのシステムはあまり考えられておらず、サイト間スクリプティング脆弱性によりパスワードやクレジットカード番号が漏洩するといった、重大なセキュリティホールの原因となっている。そこで、本研究では正規表現を文字列の型とみなして型検査や型推論を行うことにより、文字列処理の検証・解析を実現する。本年度の論文 [6, 14] では、型システムの基礎となる型付け規則と、部分的な型推論（パターン変数のみ）の方式を提案し、それらの正当性を証明した。さらに、制約解消と文脈自由文法の正規表現近似にもとづき、パ

ターン変数以外の変数の型も推論する方式について研究した。これにより、本研究を応用した現実的な文字列処理言語を開発することも可能になると期待される。

5 安全なプログラミング言語の応用

MLやSchemeといった安全なプログラミング言語や、それらの言語における研究の成果を、Java [4, 7] やC++ [5] といった従来の言語に適用したり、メールシステム [11], 暗号プロトコル [15], 移動コードによるパケットフィルタリング [9], ゲノム解析 [8, 16], 対戦プログラム [2] といった様々な領域の問題に応用し、そのような言語ないし研究が（従来の技術と比較しても）有用であることを実証した。

6 今後の展望

来年度は、上述の方向性にしたがって各研究をさらに発展・推進し、特に

- 安全なC言語処理系の完成
- より現実的・大規模なアプリケーションによるカーネルモードLinuxの実験
- 正規表現型を実装した文字列処理言語の設計・開発

といった、本年度の成果を応用した成果の実現・公開を目指す。

参考文献

[1] Kernel Mode Linux. Toshiyuki Maeda. http://www.yl.is.s.u-tokyo.ac.jp/~tosh/kml/.

[2] 第5回ICFPプログラミングコンテスト優勝. 大岩 寛, 住井 英二郎, 関口 龍郎. 2002年10月. http://icfpcontest.cse.ogi.edu/.

[3] 大岩 寛, 住井 英二郎, 米澤 明憲: 安全性を保証するANSI-C実行系の実装手法. コンピュータソフトウェア, 岩波書店, 19巻3号 39–44頁, 2002年5月.

[4] Akihito Nagata, Eijiro Sumii, and Akinori Yonezawa: A Scheme-to-Java Translator with Soft Typing. Manuscript, May 31, 2002. 7 pages. http://www.yl.is.s.u-tokyo.ac.jp/~sumii/pub/scm2java.ps.gz.

[5] 増山 隆, 住井 英二郎, 米澤 明憲: C++テンプレートを分割コンパイルするためのアプローチ. 情報処理学会第39回プログラミング研究会, 2002年6月17–18日. 16頁.

[6] Naoshi Tabuchi, Eijiro Sumii, and Akinori Yonezawa: Regular Expression Types for Strings in a Text Processing Language. Proceedings of Workshop on Types in Programming (TIP'02), Dagstuhl, Germany, July 9, 2002 (Electronic Notes in Theoretical Computer Science, Elsevier Science, the Netherlands, to appear). 19 pages.

[7] Reynald Affeldt, Hidehiko Masuhara, Eijiro Sumii, and Akinori Yonezawa: Supporting Objects in Run-time Bytecode Specialization. Proceedings of ACM SIGPLAN ASIAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation, Aizu, Japan, September 12–14, 2002, pp. 50–60.

[8] Eijiro Sumii and Hideo Bannai: VMλ: A Functional Calculus for Scientific Discovery (Extended Abstract). Proceedings of Sixth International Symposium on Functional and Logic Programming, University of Aizu, Aizu, Japan, September 15–17, 2002 (Lecture Notes in Computer Science, Springer-Verlag, Germany, vol. 2441), pp. 290–304.

[9] Eric Y. Chen, 柏 大, 冨士 仁, 米澤 明憲: Moving Firewall における DDoS 攻撃対策システムの評価. 電子情報通信学会情報ネットワークシステム研究会予稿集, 信学技報, NS2002-121, 2002 年 9 月, pp. 73–78.

[10] Eric Y. Chen, Hitoshi Fuji, and Akinori Yonezawa: Solution Deployment on Multi-Provider Networks, Proceedings of The OPENSIG 2002 Workshop, University of Kentucky, Lexington, Kentucky, US, October 17–18, 2002.

[11] Reynald Affeldt and Naoki Kobayashi: Formalization and Verification of a Mail Server in Coq. Proceedings of International Symposium on Software Security, Tokyo, Japan, November 8–10, 2002 (Software Security – Theories and Systems, Lecture Notes in Computer Science: Hot Topics, Springer-Verlag, Germany, vol. 2609). 17 pages.

[12] Yutaka Oiwa, Tatsuro Sekiguchi, Eijiro Sumii, and Akinori Yonezawa: Fail-Safe ANSI-C Compiler: An Approach to Making C Programs Secure (Progress Report). Proceedings of International Symposium on Software Security, Tokyo, Japan, November 8–10, 2002 (Software Security – Theories and Systems, Lecture Notes in Computer Science: Hot Topics, Springer-Verlag, Germany, vol. 2609). 21 pages.

[13] 末永 幸平, 大岩 寛, 住井 英二郎, 米澤 明憲: Fail-Safe C のためのインターフェイス定義言語. 第 5 回プログラミングおよびプログラミング言語ワークショップ, 2003 年 3 月 5–7 日. 採録済. 14 頁.

[14] 田渕 直, 住井 英二郎, 米澤 明憲: テキスト処理言語における文字列のための正規表現型. 情報処理学会論文誌: プログラミング, 採録済. 12 頁.

[15] Eijiro Sumii and Benjamin C. Pierce: Logical Relations for Encryption. Journal of Computer Security, IOS Press, the Netherlands, to appear. 29 pages.

[16] Eijiro Sumii and Hideo Bannai: VMλ: A Functional Calculus for Scientific Discovery. Submitted for publication. 22 pages.