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

現代の計算機システムにおいて、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等の技術について研究を推進している。本年度は、安全な C 言語処理系、ユーザプログラムをカーネルモードで実行可能な Linux、移動可能なファイアウォール、移動コードを実現するための言語およびミドルウェア、動的型付き言語のメモリ管理を効率化する手法、リファレンスモニタの実装技術に関して研究成果を得た。

1 はじめに

現代の計算機システムにおいて、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等の技術について研究を推進している。特に、ML のような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や、移動コードの利用を強く意識した先進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。それと並行して、オペレーティングシステムのカーネルやミドルウェア等のシステムソフトウェアを安全かつ効率的に実行する方式や、ネットワークを通じた攻撃から計算機を守るための方式の研究・開発も行っている。本年度の主要な成果を以下に挙げる。

2 本年度の成果

2.1 安全な C 言語処理系のためのインタフェース記述言語

バッファ溢れ等のセキュリティホールを悪用する攻撃を防止するための、安全な C 言語処理系の研究・開発を進めた。安全な C 言語処理系はディペンダブルなシステムを作る用途に大変有用である。対外的には、安全な C 言語処理系のためのインタフェース記述言語についての研究に関する論文を国際会議 [6] および国内の会議 [8] で発表した。その論文では、外部のソフトウェアと安全な C 言語処理系との相互運用性を向上するためのインターフェース記述言語 (IDL) の設計・開発について述べられている。プログラマは IDL によってオペレーティングシステムのシステムコールや標準ライブラリ関数との相互運用に関する仕様を記述する。IDL の処理系はその仕様にしたがい、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。この IDL はメールサーバ (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安全なC言語処理系で処理して実行するための基盤となる仕組みである。上記の論文中ではIDLの処理系の性能評価を行った結果も報告されている。

2.2 我々の安全なC言語処理系による攻撃防御の実証実験

安全なC言語処理系が攻撃を効果的に防御することを示す実証実験を行った。実験では広く普及しているメールサーバソフトウェアであるsendmailを用いた。まず、sendmail 8.11.0からsendmail 8.11.5までに実在した脆弱性を利用してsendmailを乗っ取る攻撃コードを記述した。その攻撃コードはメモリアクセスの誤りを巧妙に悪用するものであり、スタックフレームに印をつけるだけのような単純なバッファ溢れ攻撃対策では防御できない。次に、脆弱性を有する部分のコードを切り出し、既存のC言語処理系と安全なC言語処理系でコンパイルして二つのバージョンのプログラムを作成した。既存のC言語処理系によって作られたプログラムでは、上記の攻撃コードによる乗っ取りが成功し、攻撃者が危険な操作を実行することができた。一方、我々の安全なC言語処理系によって作られたプログラムは攻撃コードによる乗っ取りを許さなかった。

2.3 ユーザプログラムをカーネルモードで実行可能なLinux

型検査等の実行前検証により安全性が事前に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）の構築方式等についての研究成果を国際会議で発表した[2]。この方式では、一般にハードウェアによる実行時検査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、システムコール自体にかかるオーバーヘッドは、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、大規模なプログラムを用いて性能評価やシステムの最適化を行うことに注力した。その結果、多くの有用な実験データを収集することができ、その一部は上記の論文に反映されている。大規模な実験を行うためには標準ライブラリの実装が必要であったが、その問題は既存の標準ライブラリを少ない手間で我々のLinux用に移植するための方式を発見・考案することにより解決した。本研究はディペンダブルなオペレーティングシステムを構築するための基盤となるものである。

2.4 移動可能なファイアウォール

DDoS攻撃からサーバを防御するMoving Firewallシステムについての研究を行い、そのシステムの設計･実装･評価および複数ISP間の導入方式についての論文を国際会議で発表した[1]。サーバがDDoS攻撃を受けた際には、Moving Firewallができるだけ攻撃元の近くまで遡って攻撃トラフィックを局所的に閉じ込める。被害サーバ側で攻撃パケットを遮断する従来の「点」での防御方法に比べて、インターネットを流れる攻撃パケットの帯域を全体の「面」で制限することで攻撃被害を局所化する利点をもつ。今年度は、実際の日常業務に利用されている計算機サーバの通信トラフィックを収集し、DDoS攻撃などの不正通信トラフィックを複数回にわたり発見し、それの分析を行った。実世界のサーバに対する実際の攻撃の挙動データを収集したことは、今後ネットワーク攻撃に対するディペンダブルな防御システムの研究をさらに発展させていく上で大きな価値がある。

2.5 移動コードを実現するための言語およびミドルウェアの研究

移動コードを実現するための言語 MobileScope およびランタイムシステム Comet の研究を行い、国内外の会議で発表した [4, 5, 11]。また、国際会議に論文 [3] が採択され、発表予定である。MobileScope で書かれたプログラムでは、計算を行う主体をコンポーネントとして表現する。コンポーネントは計算機間を移動することができる。この移動の機能により、MobileScope を用いるとモバイルアプリケーションの記述が極めて容易になる。MobileScope および Comet の特徴は、弱い移動と強い移動の両方をサポートしている点、移動のための処理をコンポーネントを実装するコードの中に記述する必要がない（コンポーネントの外からコンポーネントの移動を制御できる）点、資源と通信チャネルを実行時に明示的に結び付けられる点である。MobileScope によって移動可能 web サーバ、移動可能チャットシステム、移動可能ビデオストリーミングシステムなどを実装し、MobileScope の有効性を検証した。その結果、MobileScope によってディペンダブルなシステムを少ない開発コストで構築できることを我々は確認した。

2.6 動的型付き言語のメモリ管理を効率化する手法

動的型付き言語 Scheme にリージョンベースのメモリ管理を導入する方式の研究を行い、論文発表を行った [10]。Scheme はメモリ操作の安全性が保証されており、それが保証されていない C 言語に比べると開発したソフトウェアのディペンダビリティ（特に安全性）は多くの場合に高まる。これまで、リージョンベースのメモリ管理は主に静的型付き言語を対象に研究されており、動的型付き言語に適用するための技術は十分に研究されてこなかった。本研究では soft typing と呼ばれる型付け技術を用いることにより Scheme にリージョンベースのメモリ管理を導入することを可能にしている。リージョンベースのメモリ管理を使用する処理系では、ガベージコレクションのみを使用する従来の処理系に比べて、メモリ使用効率が高まり実行速度が上がることが期待されている。本研究では方式を提案するだけでなく、実際にその方式に沿ったメモリ管理を行う Scheme 処理系を実装し、性能評価も行った。本研究は、より広い視点からとらえると、リージョン推論を例にとって、soft typing をベースとすることで、先進的な静的型システムを動的型付き言語に適用する枠組みを示したとみなせる。同様に linear type や uncaught exception analysis、resource usage analysis なども、動的型付き言語に適用可能であると考えられる。

2.7 リファレンスモニタの実装技術の深化

リファレンスモニタおよびサンドボックスと呼ばれる安全なソフトウェア実行システムの実装技術を深化させる研究を行った [7, 9]。具体的な成果としては、自己修復機能を持つリファレンスモニタ SeRene の実装方式の確立がある。SeRene は安全性と頑健性を高めるための新しい仕組みを有するリファレンスモニタである。SeRene では、自身を独立したモジュールの集合体として表現することにより、実装ミスがリファレンスモニタ全体に障害をもたらすことを防ぐ。さらに、SeRene の内部には、自身が正常に動作しているかどうかを監視するための機構を組み込まれており、障害発生時には自身を再起動することによって自律的に正常状態を回復する。SeRene は Pentium プロセッサ上の FreeBSD 上に実装され、オーバヘッドなどの性能情報がすでに収集されている。

3　今後の展望

来年度は各研究をさらに発展・推進する。まず、安全な C 言語処理系に関して、処理系の実装、インタフェース記述言語の実装、実証実験を進めていく。ユーザプログラムをカーネルモードで実行可能な Linux を用いてのさらなる実験も行っていく。また、本年度に開発された実装技術をもとに実用的なリファレンスモニタを構築する作業も並行して行う。

参考文献

[1] Eric Y. Chen, Hitoshi Fuji, and Akinori Yonezawa. Federation of Network Service Providers and Its Applications. In *Proceedings of The Eighth IEEE Symposium on Computers and Communications (ISCC'2003)*, July 2003.

[2] Toshiyuki Maeda and Akinori Yonezawa. Kernel Mode Linux: Toward an operating system protected by a type theory. In *Proceedings of the 8th Asian Computing Science Conference (ASIAN'03)*, volume 2896 of *Lecture Notes in Computer Science*, pages 3–17, 2003.

[3] Takashi Masuyama, Yoshihiro Oyama, Frédéric Peschanski, and Akinori Yonezawa. Mobilescope: A programming language with objective mobility. In *Proceedings of Second International Workshop on Mobile Distributed Computing*, 2004.

[4] Frédéric Peschanski, Jean-Pierre Briot, and Akinori Yonezawa. Fine-grained dynamic adaptation of distributed components. In *International Middleware Conference 2003*, volume LNCS 2672, pages 123–142. Springer-verlag, June 2003.

[5] Frédéric Peschanski and David Julien. When concurrent control meets functional requirements, or z + petri-nets. In *International Conference of Z and B Users ZB2003*, volume LNCS 2651, pages 79–97. Springer-verlag, June 2003.

[6] Kohei Suenaga, Yutaka Oiwa, Eijiro Sumii, and Akinori Yonezawa. The Interface Definition Language for Fail-Safe C. In *Proceedings of International Symposium on Software Security*, 2003.

[7] 大山 恵弘. ネイティブコードのためのサンドボックスの技術. コンピュータソフトウェア, 20(4):55–72, 2003.

[8] 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 (PPL 2003) 論文集, 2003.

[9] 吉野 寿宏, 大山 恵弘, 米澤 明憲. 自己修復型リファレンスモニタの設計と実装. 投稿中.

[10] 永田 章人, 小林 直樹, 米澤 明憲. 動的型付き言語のためのリージョン推論に基づくメモリ管理. 日本ソフトウェア科学会第 20 回記念大会論文集, pages 31–35, 2003.

[11] 増山 隆, フレデリック ペシャンスキ, 大山 恵弘, 米澤 明憲. MobileScope: 透明な移動機能を備えた分散コンポーネント言語. 日本ソフトウェア科学会第 20 回記念大会論文集, pages 196–200, 2003.