Fail-Safe C: 安全なC言語コンパイラの実用的処理系の構築

大岩　寛 (情報セキュリティ研究センター ソフトウェアセキュリティ研究チーム)

C言語の成立経緯とセキュリティ

1970年代:

- Unix のためのシステム記述言語
  - 単純明快で高速な高級言語
  - ポインタを使った自由なメモリ操作
    (アセンブリ言語の置き換えとしての利用)

- 当時の計算機は今よりはるかに遅かった.
- セキュリティバグの影響は今より小さかった
  - ネットワークの黎明期 = 外部からの攻撃がない

2000年代:

- security hole の原因となっている
  - 言語レベルのメモリ安全性の欠如
  - 言語レベルの高機能データ構造サポートの欠如

  致命的な（任意コード実行）脆弱性の大半がC言語のメモリ操作バグに起因

- 多くのCプログラムは自由なメモリ操作を実際には必要としていない
- インターネットに関連したプログラムにとって安全性は極めて重要.

- どう解決する?

  解決1. C 言語をやめる. (Java や ML を利用)
  - ある種理想的、バグを減らすのにも有効
  - だが、既存プログラムが使えない!

  解決2. C 言語を安全にする
  - 実現できれば実践的
  - 既存プログラムをそのまま使える!

Fail-Safe C: C言語セキュリティの切り札

- **100% ANSI C 規格 (JIS規格) 上位互換**
  - ポインタ演算, キャスト, 共用体等
    C言語特有の操作に全て対応
  - プログラムの書き換え不要
- **100% メモリ安全**
  - Java, C#, Lisp, ML などと同等の安全性を確保
- C言語プログラム特有の多くの記述をサポート
  - 既存のプログラムとの互換性を可能な限り確保
    （厳密にANSI C 規格に準拠しないプログラムが多い）
- 可能な限り高速に動作
  - 実運用での導入が可能な速度を目指す
    - 安全性を犠牲にせずに実装の工夫などで実現
    - 「遅くても数倍程度」が当面の目標

現在のシステム:

- Linux (Intel IA32) 上で動作
- 完全なコンパイラとリンカのインタフェースを提供
  - Gcc の代わりに "**fscc**"を用いるだけで動作
  - 分割コンパイルも安全にサポート
    - リンク時のモジュール整合性検査機構を実装
- 安全なC言語標準ライブラリ
  - POSIX 規格の500以上の関数をサポート
    - 関数1つ1つに対応した安全性検査付きの実装を作成
  - 関数の誤った使用をきちんと検出・メモリ破壊を防止

- 複数の実用アプリケーションが動作
  - **OpenSSL – 暗号通信ライブラリ**
  - **OpenSSH – 暗号遠隔ログイン**
  - **BIND9 (named) – 標準ネームサーバ**
  - **thttpd – 軽量高速Webサーバ**

  がほとんどプログラム書き換え無しに動作
- 性能: 計算時間で平均 3~5倍程度
  - プログラムにより 1.01~7 倍程度
  - 最適化による更なる改善を予定

内部の実装技術

(1) 型を使ったメモリ管理

- メモリ領域をブロック単位でオブジェクト化:
  - 要素数に加え、データ型の情報をブロックに記録
  - 基本的なメモリの読み書き操作をメソッドとして付加

キャスト操作があっても安全なメモリ操作を実現

(2) Smart ポインタ と Cast フラグ

- ポインタを内部的に2ワードで表現
  - 参照先のオブジェクトを常に記録
  - ポインタ演算のためのオフセットをポインタに付加
  - さらに、キャストの有無を示すフラグを付加

キャストがない場合は高速なメモリアクセスを実現

- プログラムからは、従来同様の1ワード表現に見える
  - 既存プログラムとの高い互換性を実現

ホームページ: http://www.rcis.aist.go.jp/project/FailSafeC-ja.html or http://failsafec.jp/

- 4月11日より、処理系をオープンソースソフトウェアとして一般に公開

* 本研究の実装の一部は、経済産業省「新世代情報セキュリティ研究開発事業」の一部として行いました.
ライブラリ実装の一部は、株式会社レピダムとの共同プロジェクトとして行われました.

今後の展望

- 処理系の更なる効率化・利便性の向上
  - 静的解析による最適化の導入
  - 支援ツール類の実装 など
- 実システムへの適用を目指した取り組み
  - 1-CD 環境・組み込み環境などへの適用
  - その他 企業等への働きかけと導入サポート
    - 知見を基に、実用システムのための改善に反映
    - 興味がありましたらぜひお声をおかけください

研究協力

- Fail-Safe C to Java 変換 (東北大学 上嶋他)
- VitC (情報流解析・保護付き Fail-Safe C, 東京大学 古瀬他)

関連研究

- CCured [Necula et al. 2002]
  - キャストの有無をコンパイル時に分類し変換
  - プログラムが大きくなると効率が大きく低下
    - 型システムの設計上、「キャスト有り」のポインタが参照する全てのデータが「キャスト有り」でなければならない
    - Fail-Safe C は実行時型情報があるためこの問題が起こらない