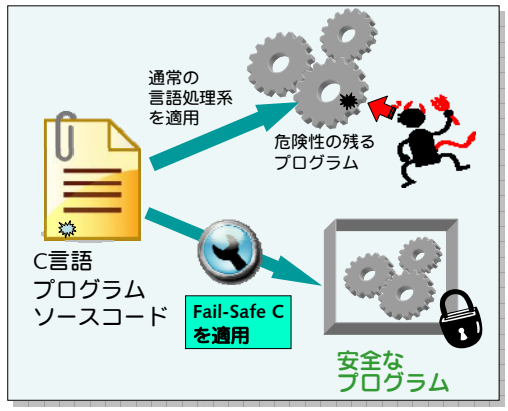
Fail-Safe C: 安全性を保証する実用C言語コンパイラの今後の展開

大岩　寛　（ソフトウェアセキュリティ研究チーム）

■ Fail-Safe C の目標: 既存のC言語ソフトウェアを可能な限り低いコストで安全にする



- ANSI-C の規格に準拠したプログラムを全てサポート
- メモリ安全性を完全に実現
  - 不正コードのメモリ上プログラムへの侵入を確実に防止
  - Java, C# と同等の安全性をC言語でも実現

サーバプログラム等に事前に適用しておけば、
仮に未知の脆弱性が発見され攻撃されても
被害を最小限に抑えることができる

■ 現在の実装の概要

- **Version 2.0**: 2010年5月に公開 (予定)
- 様々なアーキテクチャをサポート
  - **x86-64 (amd64) に新規対応**
  - その他, i386, ARM, MIPS 等の Linux で動作
- C90 の仕様をほぼフルサポート
- 数々の実用アプリがコンパイル可能
  - ほぼ再コンパイルだけで、プログラムが安全に動作
- オープンソースソフトウェアとして公開
  - ソース、Debian 非公式パッケージ等を配布中
  - ライセンス条件を緩めました (QPL → Apache)

例: (設定例を公開)
- OpenSSL (暗号処理基本ツール)
- OpenSSH, GnuPG (暗号通信ソフト)
- BIND9* (DNSサーバ)
- Postfix, qmail (メールサーバ)
- 各種データ処理ライブラリ
  - libtiff, libpng, zlib, bzip2, libogg, pcre*
- その他
  - thttpd, bzip2, bash, sed

* 複数のソフトのバグを発見

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

国際論文発表: ACM PLDI 2009 (2009年6月)

Yutaka OIWA, "Implementation of the Memory-safe Full ANSI-C Compiler", *Proc. PLDI 2009*, ACM SIGPLAN Notices, June 2009.

■Fail-Safe C の今後の展開について

1. 更なる利便性・可用性の向上

画像ファイルの処理など
特にバグや攻撃の多い箇所を
集中的に強化したい

- 公開後、実用的に使う上で出てきたいくつかの要求
  - 既存プログラムの一部に適用して部分的に安全性の向上に使いたい (native C → Fail-Safe C)
  - バイナリ配布の既存ライブラリなどを簡単に使えるようにしたい (Fail-Safe C → native C)

→ **双方向利用可能なライブラリ互換システムの設計と実装**
  - 2010年度 A-STEP フィージビリティスタディプロジェクトとして共同研究

2. 対応アーキテクチャ・環境・言語機能のさらなる拡充

- 1. の延長として、標準ライブラリの移植性の向上を通じて対応環境の拡大
  - BSD, Darwin (MacOSX) 等 POSIX 環境への対応など
- 対応言語機能の拡張: (例) C99, C++ 等
- その他機能の拡張: (例) 柔軟なエラー処理等