卒論ミーティング (10/29)

31010 佐藤秀明

概要

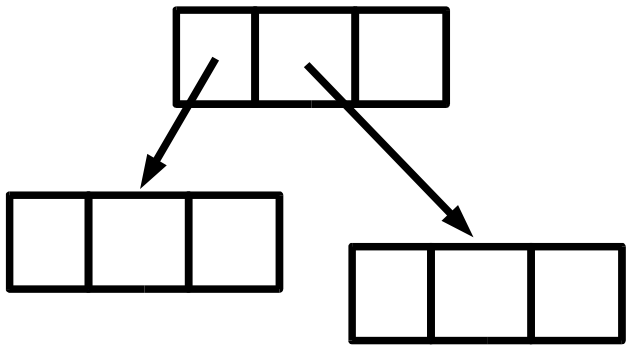
- 先週のおさらい (Tamperproofing)
- 検査コードを埋め込む方法
- その他

やりたいこと (再掲)

- 監視するコードの内容を基にデータ構造を参照
  - 動的な情報は攻撃者による解析を困難にする
- 改変されたコードを基にした操作はポインタの不正な参照 (NULL 参照など) を引き起こし、直ちに異常終了

operate_structure(structure, codes);

↓　各ノード参照



コード検査の相互依存

- あるコードＴを検査するコードＣもまた、他のコードC' によって検査されてもよいことにする [1]
  - プログラムのある一部分を改変するために、プログラム全体に拡がる検査の連鎖全体を修正しなければならなくなる

検査コードを埋め込む位置

- 検査対象となるコードＴが実行される前に、Ｔを検査するコードＣが実行されていて欲しい
- 検査コードＣが実行される前に、Ｃが操作するデータ構造は完成していなければならない

データ構造を構成するコード(Initializer)

↓

対象となるコードを検査するコード(Checker)

↓

対象となるコード(Target)

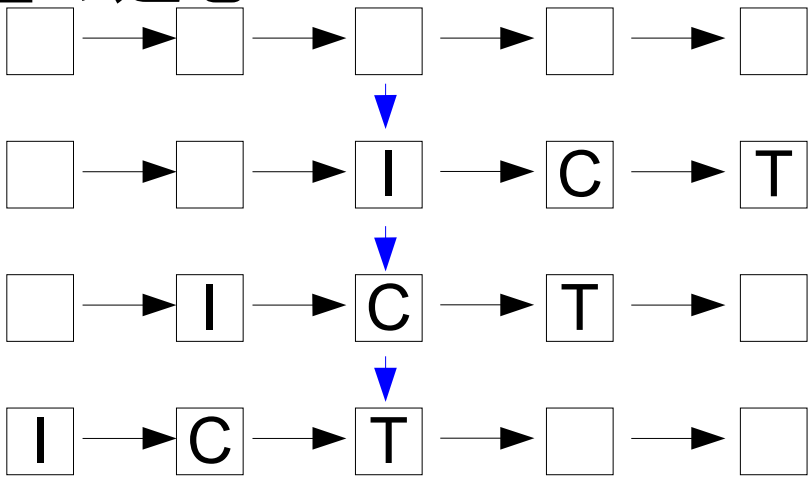
検査コードを埋め込む手順(1)

1 プログラムの制御フローを取得する

2 以下の操作を、フローグラフの葉から根の方向に繰り返す

1 検査対象となるブロックＴを支配するブロックＣに、Ｔを検査するためのコードを埋め込む

2 検査コードが埋め込まれているブロックＣを支配するブロックＩに、Ｃで操作されるデータ構造を構成するためのコードを埋め込む



検査コードを埋め込む手順 (2)

- 他のブロックに支配されないブロック (開始ブロック) はどうするか ?
  - 「検査対象となるコード T が実行される前に、T を検査するコード C が実行される」という条件を諦める
    - 開始ブロックの後に必ず実行されるブロックに、開始ブロックの検査を託す
  - 厳密な検査を諦める
    - 検査の及ばないコードを少しだけ持つ
    - コード全体は検査されるが、検査の際のヒントになる値をコード外に置く

その他

- データ構造をどうするか？
  - 元プログラム中のデータ構造を利用する
- 各ブロックの大きさをどうするか？
  - 検査コードの埋め込み効率やプログラム本体の実行性能が落ちないように、ブロックを適当にマージ・分割する

Reference

- [1] H. Chang and M. Atallah. Protecting software code by guards. In Proc. 1st ACM Workshop on Digital Rights Management (DRM 2001), pages 160-175. Springer LNCS 2320, 2002.