様式 C-19

科学研究費補助金研究成果報告書

平成 23 年 6 月 10 日現在

機関番号：57403
研究種目：若手研究(B)
研究期間：2008～2010
課題番号：20700034
研究課題名（和文）
実行時間の差分を利用した不正な動的解析の防止
研究課題名（英文）
Exploiting Time-sensitive Code for Protecting Software against Dynamic Attacks
研究代表者
神崎　雄一郎（KANZAKI YUICHIRO）
熊本高等専門学校・人間情報システム工学科・准教授
研究者番号：90435488

研究成果の概要（和文）：
本研究では，不正な解析（不正なリバース・エンジニアリング）を行う攻撃者からソフトウェアを保護するための一方法を提案した．提案方法は，ソフトウェアの任意の部分を，実行時間依存コード，すなわち，ソフトウェアの一部の実行時間に応じて動的に内容が変化するコードに変形する．提案方法を用いれば，攻撃者が動的解析によってソフトウェア内の秘密情報を取得するためのコストを効率的に増大させることができる．保護アルゴリズムの開発，試作システムの実装，および提案方法の有効性の考察・評価を行った．

研究成果の概要（英文）：
In this study, a systematic method for protecting software against malicious reverse engineering attacks is proposed. The method transforms an arbitrary part of the program into a time-sensitive code, that is, a code which is modified during execution according to the time taken to execute a part of the program. The method helps to efficiently increase the cost of obtaining secret information via dynamic attacks. The achievements include the development of the algorithm, the implementation of the prototype system, and the evaluation of the effectiveness of the proposed method.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2008 年度	800,000	240,000	1,040,000
2009 年度	700,000	210,000	910,000
2010 年度	700,000	210,000	910,000
年度			
年度			
総　計	2,200,000	660,000	2,860,000

研究分野：ソフトウェア保護
科研費の分科・細目：情報学・ソフトウエア
キーワード：ソフトウェア保護，著作権保護，難読化，動的解析，耐タンパ

１．研究開始当初の背景

秘密情報を含むソフトウェアの増加にともない，エンドユーザによるソフトウェアの解析行為（不正なリバース・エンジニアリング）を防ぐための技術への要求が高まっていた．例えば，デジタルコンテンツの著作権管理を行うソフトウェアは，内部に含まれる復号鍵やライセンスチェックを行う命令文の漏えいを防止することが求められていた．

ソフトウェアの解析を困難にする方法は，従来多数提案されていたが，既存手法の多くは静的解析（逆アセンブラなどを用いてプロ

グラムを実行させずに行う解析）を防止することを目的にしたものであった．しかし，高機能なデバッガなどをエンドユーザが容易に入手できる昨今では，静的解析と動的解析（デバッガなどを用いてプログラムを実行させながら行う解析）を組み合わせた攻撃が主流であるため，従来の静的解析を防止する方法に加えて，動的解析を困難にするための方法が強く求められていた．

２．研究の目的

本研究では，不正な動的解析を困難にするためのアルゴリズムを考案すること，および，そのアルゴリズムに基づく試作システムを実装し，有効性を検証することを目指した．

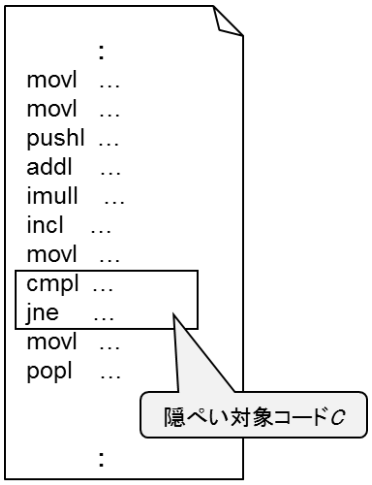
提案方法では，攻撃者が実行の一時停止を伴う動的解析（例えば，デバッガのブレイクポイントを用いた解析）を行った場合に，プログラムの実行時間が通常よりも長くなる点に注目し，通常実行時の実行時間と動的解析時の実行時間の差異を動的解析の検出・防止に用いることを考えた．具体的には，プログラムの一部の実行時間が通常実行時と同程度であるときのみ元来の内容に書き換わる「実行時間依存コード」(Time-sensitive Code)をプログラムの多数の箇所に設けることで，動的解析を伴う攻撃を妨げることを基本アイデアとした．

３．研究の方法

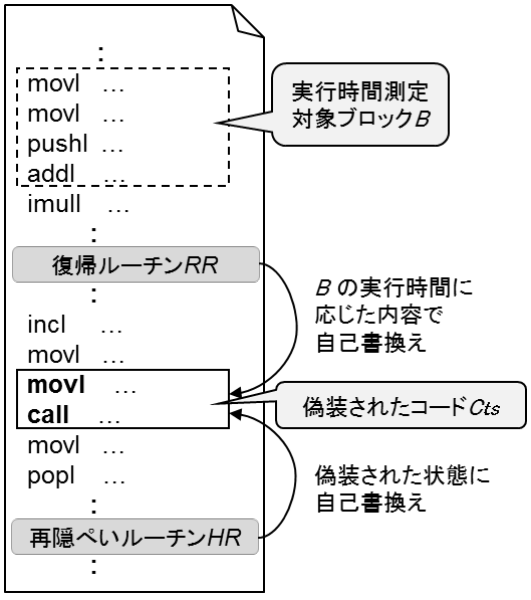
まず，提案方法の概要について説明する．図１(a)および(b)は，元来のプログラム *P* および保護された状態のプログラム *Pp* の概念図をそれぞれアセンブリレベルで示している．ここでは説明のための例として Intel x86 系 CPU を想定し，AT&T 文法によって表している．

元来のプログラムの一部（攻撃者に知られたくない部分）は，隠ぺい対象（保護対象）のコードとして選択され，別のコードで偽装（上書き）される．図１の例では，元来のプログラムに含まれる cmpl および jne という命令が，保護されたプログラムでは movl および call という命令に偽装されている．偽装されたコード *Cts* は，復帰ルーチン *RR* および再隠ぺいルーチン *HR* により自己書換えされる，すなわち，実行時に書き換えられる．*RR* は，保護されたプログラムに含まれる実行時間測定対象ブロック *B* の実行時間が一定の範囲内である場合は，元来のコードで書き換え，範囲外である場合は，元来とは異なるコードで書き換える．一方 *HR* は，*RR* によって書き換えられたコードを再び偽装されたコードに書き換える．*Cts* は，*RR* が実行されてから *HR* が実行される間の期間のみ，*C* となる．ただし，*B* の実行時間が一定時間より長い場合あるいは短い場合は，*Cts* は *C* とは異なるコードに書き換わる（*C* のコードはメモリ上に現われない）．すなわち，*Cts* は *B* の実行時間によってコードの内容が変化する実行時間依存コードとなる．

攻撃者が動的解析によって *C* のコードを知るには，*B* を一時停止させずに通常実行時と同程度の速度で通過し，*RR* と *HR* に挟まれたコード内に実行の制御を移す必要がある．このような仕組みを連鎖的に何重にも適用することにより，攻撃者が *C* を知るためのコストを高くする．



(a) 元来のプログラム *P*



(b) 保護されたプログラム *Pp*

図１　提案方法の概念図

以上のようなアイデアに基づき，本研究は，(1)保護アルゴリズムの詳細設計，(2)試作システムの実装，(3)提案方法の有効性に関する考察・評価，という３つの段階に分けて進められた．

４．研究成果

(1) 保護アルゴリズムの詳細設計

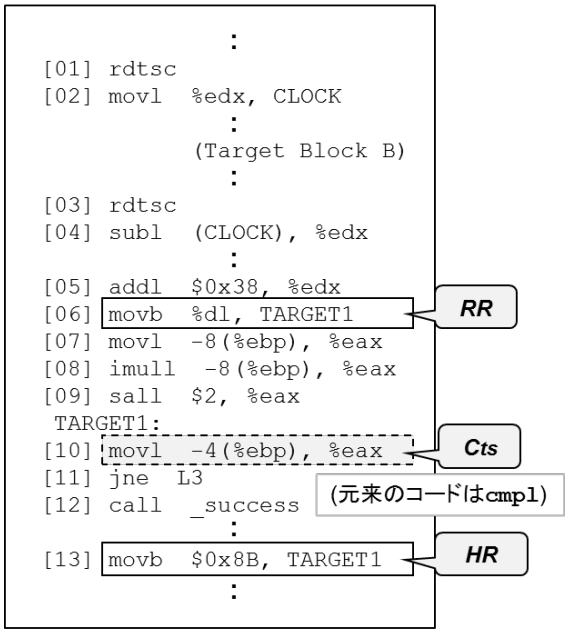
３．で述べたアイデアに基づき，プログラム内の任意の秘密情報を動的解析から保護する，すなわち，動的解析による秘密情報の取得に要するコストを効率良く増大させるための系統的なアルゴリズムを開発し，その手順をまとめて発表した（発表文献は〔学会発表〕①，④など）．また，隠ぺい対象の指定や偽装内容の決定をユーザ（提案方法の使用者）が柔軟かつ簡単に行えるように，それらの作業を（アセンブリレベルではなく）高級言語レベルで行うための方法についても検討して発表した（発表文献は〔雑誌論文〕①，〔学会発表〕⑥など）．

(2) 試作システムの実装

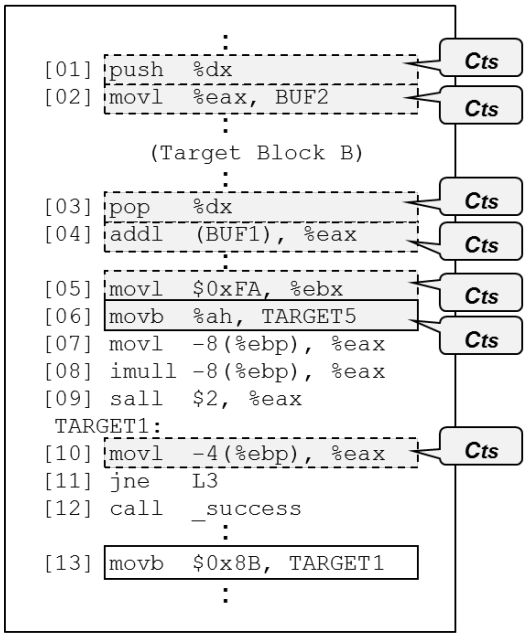
(1)で検討したアルゴリズムに従い，提案保護方法を任意のソフトウェア（アセンブリプログラム）に適用するための試作システムを実装した．

図２に，試作システムによって提案方法が適用されたプログラムの例（一部）を示す．このプログラムには，簡単なパスワードのチェックルーチンが含まれている．各行の左端に記してある番号は説明のための行番号である．図 2(a)は，提案方法により保護が１回適用されたプログラムの例である．元来のプログラムでは 10 行目の位置に存在した，パスワードのチェックに必要な比較命令である cmpl 命令が隠ぺい対象(秘密情報)として選択され，movl 命令で偽装されている（図中 *Cts*）．6 行目の復帰ルーチン *RR*（movb 命令１つで構成されている）は，1～4 行目において測定された *B*（プログラムの一部）の実行時間に応じて，10 行目の命令を書き換える．また，13 行目の再隠ぺいルーチン *HR* は，10 行目の命令が実行された後に，その命令を再び偽装命令である movl 命令に書き換える．実行時間の測定には RDTSC 命令が用いられている．*B* の直前（1 行目）と *B* の直後（3 行目）において各時点における実行クロック数を RDTSC 命令で測定しておき，その差分を求めることによって *B* に要した実行時間(実行クロック数）を取得している．movl 命令 を元来の cmpl 命令にするためには，movl 命令のオペコードの部分（1 バイト目） を cmpl のオペコードの値に書き換えればよい．5 行目の addl 命令において，書き換えるコードの値を *B* の実行時間に基づいて計算しているが，*B* の実行時間が長い場合は書き換える値が大きくなり，cmpl 命令とは異なる命令に書き換わる．また，*B* の実行クロック数が短くなるように *RR* や *B* を改ざんしても，その実行クロック数が判定基準の範囲外である限り，*Cts* は元来の命令に書き換わることはない．

図 2(b)は，図 2(a)に提案方法を繰り返し適用したプログラムの例である．復帰ルーチンや実行時間を測定するための命令なども隠ぺい対象（*Cts*）として選択され偽装されているため，図 2(a)と比べて攻撃者が秘密情報（10 行目）を見つけるのが困難となる．このように，適用を繰り返して多くの実行時間依存コードを構成することで，攻撃者が秘密情報を取得するためのコストを増大させることができる．



(a) 保護が１回適用されたプログラムの例



(b) 保護が繰り返し適用されたプログラムの例

図２　保護されたプログラムの例

(3)提案方法の有効性に関する考察・評価

以下に示すように，提案方法の有効性に関する考察および評価を行った（関連する発表文献は〔雑誌論文〕②,〔学会発表〕①など).

①現実的な攻撃者のモデリングと提案方法の有効性の考察

まず，静的解析・動的解析両方を組み合わせてソフトウェアの解析を行う現実的な攻撃者のモデル（典型的な攻撃者の環境と解析手順の詳細）について調査・検討した．得られた攻撃者モデルを用いて，提案方法によって保護されたソフトウェアが「現実的な攻撃」を受けた場合の耐性について考察した．提案方法は，逆アセンブラなどを用いる静的解析に加えて，一時停止を伴う動的解析（デバッガのブレイクポイント機能を使った解析など）を妨げることができるため，現実的な攻撃者がソフトウェア内の秘密情報を取得するのに必要なコストを効率良く増大させることができる．秘密情報に該当する命令そのものに加え，秘密情報の場所を特定する手がかりとなる命令や，保護機構として追加した命令列などを対象に繰り返し保護を適用することで，より耐性の高い保護を行うことができる．

②実行時間のオーバーヘッドに関する実験

提案方法が適用されたソフトウェアについて，適用されていない場合と比較してどの程度の実行時間のオーバーヘッドが生じるかを，実験によって評価した．具体的には，実験用ソフトウェアに提案方法を適用し，適用の前後で一定の処理を行わせたときの実行時間がどの程度変化するかを測定した．実行時間のオーバーヘッドは保護を適用した範囲の広さ（命令数）にしたがって増加するという結果が得られ，提案方法を適用する度合（保護の強さ）と実行時間のオーバーヘッドの大きさはトレードオフの関係にあることがわかった．

以上のように，不正な動的解析を困難にするための系統的な方法を提案し，国内外に成果を発表した．提案した保護方法は，プログラムの難読化法や暗号化法等，既存の保護技術と組み合わせて適用することが可能であり，今後のソフトウェア保護技術の一要素として用いられることが期待される．

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計２件）

① 神崎雄一郎，門田暁人，中村匡秀，松本健一，“ソースコードレベルにおけるプログラムのカムフラージュ”，コンピュータソフトウェア（日本ソフトウェア科学会学会誌），Vol.28，No.1，pp.300-305，Feb. 2011（査読有).

② Hiroki Yamauchi, Akito Monden, Masahide Nakamura, Haruaki Tamada, Yuichiro Kanzaki, Ken-ichi Matsumoto, “A Goal-Oriented Approach to Software Obfuscation”, International Journal of Computer Science and Network Security, Vol.8, No.9, pp.59-71, Sep. 2008（査読有).

〔学会発表〕（計６件）

① Yuichiro Kanzaki, et al.，“A Software Protection Method based on Time-sensitive Code and Self-modification Mechanism”，The IASTED International Conference on Software Engineering and Applications (IASTED SEA 2010), Nov. 1, 2010 (Marina del Rey, USA).

② 坂口英司ほか，“動的解析防止を目的とした実行時間依存のコード構成法”，第 18 回電子情報通信学会九州支部学生会講演会，Sep. 24, 2010（福岡市）.

③ 尾上栄浩ほか，“C 言語で偽装内容を記述できるプログラムカムフラージュ”，第 18 回電子情報通信学会九州支部学生会講演会，Sep. 24, 2010（福岡市）.

④ 神崎雄一郎ほか，“実行時間差に着目したコードの隠ぺい方法”，第 8 回情報科学技術フォーラム(FIT2009)，Sep. 2, 2009（仙台市）.

⑤ 山口恭平ほか，“実行時間差を用いた不正な動的解析の防止”，電子情報通信学会 関西支部 第 14 回学生会研究発表講演会，Mar. 9, 2009（大阪市）.

⑥ Yuichiro Kanzaki, et al.，“Program Camouflage: A Systematic Instruction Hiding Method for Protecting Secrets”，World Congress on Science, Engineering and Technology, Sep. 24, 2008 (Heidelberg, Germany).

６．研究組織

(1)研究代表者

神崎　雄一郎（KANZAKI YUICHIRO）

熊本高等専門学校・人間情報システム工学科・准教授

研究者番号：90435488

(2)研究分担者

なし

(3)連携研究者

なし