科学研究費助成事業　　研究成果報告書

平成　２６　年　　６　月　１６　日現在

機関番号：　５７４０３
研究種目：　若手研究(B)
研究期間：　2012～2013
課題番号：　２４７０００３９

研究課題名（和文）ソフトウェア保護機構の「発見の困難さ」の評価

研究課題名（英文）Evaluating the Stealthiness of Software Protection Code

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

交付決定額（研究期間全体）：（直接経費）　　1,700,000円、（間接経費）　　510,000円

研究成果の概要（和文）：本研究では，ソフトウェア保護機構の発見の困難さを評価する一方法を提案した．保護機構が「不自然」なコード(命令列)を含んでいる場合，保護機構が攻撃者に発見されやすくなるという仮定のもと，保護機構の発見の困難さを，N-gramモデルなどによって数値化したコードの不自然さに基づいて評価した．ケーススタディを通して，元来のコードの意味を破壊する変形を伴う保護や，意味のないコードを無作為に挿入する保護は，不自然な命令列を生じやすく，保護機構の発見を容易にする傾向が強いことがわかった．

研究成果の概要（英文）：This research proposes a method for evaluating the stealthiness of software protection code. Artificial code fragments, which means unusual code fragments caused by obfuscation, are easy to distinguish from unprotected code fragments. Based on the fact, the degree of stealthiness is estimated according to the artificiality of the protected code, by means of probabilistic language model (N-gram model). The results of the case study show that the obfuscation techniques that corrupt the original semantics of the program or that just insert junk code fragments to the program, tend to decrease the stealthiness of the code.

研究分野：　総合領域

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　セキュリティ　プログラムの難読化　耐タンパソフトウェア　確率的言語モデル

1. 研究開始当初の背景

商用のソフトウェアには，商業的価値の高いアルゴリズム，ライセンスチェックのルーチンなど，ユーザに知られたくない秘密情報を含む場合がある．そのような秘密情報を取得しようとする悪意のあるユーザ (以下，攻撃者と呼ぶ) の解析行為を防ぐために，プログラムの難読化・暗号化などのソフトウェア保護方法が多数提案されている．保護方法の利用者や開発者など，ソフトウェアを防御する側にとっては，保護方法がどれだけ「強い」のか，すなわち，攻撃者が目的を達成するのに必要な労力 (時間) を保護方法によってどれだけ増大させることができるか，という点を把握することは大変重要である．そのため，ソフトウェア保護方法の強さを検証するための有用な指標や方法が強く求められている．

保護の強さを正確に検証するには，現実的な攻撃方法を踏まえて複数の視点から攻撃に要するコストを評価する必要がある．Collbergらが述べているように，ソフトウェアに対する攻撃が「保護機構の発見」,「保護機構の理解と改ざん」,「動作テスト」という 3 つのステップを繰り返すことであると考えると，ソフトウェア保護方法の強さを検証するには，保護機構の理解・改ざんの困難さに加えて，保護機構の発見の困難さを評価する必要があるといえる．しかし，発見の困難さを評価するための具体的な方法は，研究開始当初ほとんど論じられていなかった．そこで本研究では,「保護機構の発見の困難さ」の評価方法の開発に取り組むことにした．

2. 研究の目的

本研究の目的は，保護機構の発見の困難さを評価する一方法を提案することである．具体的には，コードのステルシネスを定量的に測定する方法を提案する．コードのステルシネスとは，直観的にはコード (命令列) の「目立ちにくさ」あるいは「保護されていないコードとの区別のつきにくさ」を表す指標である．あるコードのステルシネスが高い場合，コードに保護機構 (保護のために追加・変形された命令列) が含まれていることが，保護機構特有の特徴のある命令列を手がかりに保護機構を見つけようとする攻撃者に気づかれにくいと判断する．提案方法は，保護を適用したコードのステルシネスを測定できる，すなわち，コードに保護機構が含まれているように見える度合を数値化できるため，保護機構を発見する困難さの判断の一助となると考える．

3. 研究の方法

まず，コードのステルシネスを評価するにあたって，保護機構のコード，すなわち，保護のために追加・変形されたコードが「不自然」であればあるほど，保護されていないコードと区別が付きやすく，ステルシネスが低い (攻撃者に発見されやすい) コードになると仮定した．ここでコードが不自然であるとは，コンパイラによって出力されるアセンブリ (機械語) コードとしてもっともらしくないことを指す．たとえば，暗号化されたコードが逆アセンブルされた場合，一般的に出現頻度の低い命令が意味なく連続して出現する場合が多い．また，実行時に自身のコード領域を書き換える動的難読化が適用されたコードは，フラグに影響を与えない命令と条件付ジャンプ命令が組み合わされるなどの意味的なつながりが破壊されたコードとなる場合がある．このような一般的なアセンブリコード (コンパイラによって出力されるコード) としてもっともらしくない不自然なコードは，攻撃者にとって保護されていないコードと区別が付きやすいため，ステルシネスが低いと考える．

以上の考えに基づき，(1) コードの不自然さの定式化，(2) 不自然さを評価するシステムの実装，(3) 評価システムを用いたケーススタディの実施，(4) ステルシネスを評価するための他のアプローチの検討，という手順で研究を行った．

4. 研究成果

上記の研究手順に沿って，研究成果を報告する．

(1) コードの不自然さの定式化

まず，コードの不自然さを，確率・統計的自然言語処理の分野で広く用いられている N-gram モデルを用いて定式化した．N-gram モデルは，ある時点における単語の生起を (N-1) 重マルコフ仮定で近似するモデルである．N-gram モデルによって，与えられたコードを構成する命令列 $i_1^n = i_1 i_2 \dots i_n$ の生成確率 $P(i_1^n)$ は，次のように近似される．

$$P(i_1^n) \approx \prod_{k=1}^{n} P(i_k | i_{k-N+1}^{k-1})$$

すなわち，k 番目の命令 i_k の生成確率が，直前の (N-1) 命令 $i_{k-N+1} \dots i_{k-2} i_{k-1}$ にのみ依存すると考える．この生成確率 $P(i_1^n)$ の値が小さいほど，アセンブリ言語のコードとしてはもっともらしくなく，不自然であると考える．

命令列 $i_1^n = i_1 i_2 \dots i_n$ で構成されるコード C の不自然さ $A(C)$ を，上記の生成確率 $P(i_1^n)$ を用いて，次のように定義した．

$$A(C) = -log_{10} P(i_1^n)$$

$A(C)$ の値が大きいほど，アセンブリ言語としてもっともらしくなく，C は不自然であると考える．

あるアセンブリコード C の不自然さと，C に含まれる比較命令 (cmp) や分岐命令 (je および jmp) を命令のカムフラージュ法によって難読化 (カムフラージュ) した C_{camf} の不自然さを比較したものを，表 1 に示す．ここでは，2,030 個のソフトウェアのアセンブリコードから生成された N-gram モデルを用いている．$N = 2$ および $N = 3$ のどちらの場合においても，C の不自然さよりも C_{camf} の不自然さの方が大きくなっていることがわかる．このように，難読化等により変形されたコードが元来と比べてどの程度不自然になったか，すなわち，コードとしてのもっともらしさがどの程度失われたかを定量的に評価できるよう，N-gram モデルを用いてコードの不自然さの定式化を行った．

表 1　不自然さの評価例

	C	C_{camf}
コード	cmp	mov
	je	nop
	mov	mov
	jmp	nop
	mov	mov
	mov	mov
	leave	leave
	ret	ret
不自然さ ($N = 2$)	10.1	16.1
不自然さ ($N = 3$)	10.3	14.7

(2) 評価システムの実装

(1) で述べたアイデアに基づき，コードの不自然さを評価するシステムを実装した．構築したシステムの概要を図 1 に示す．まず，言語モデルを構築するためのソフトウェア群 (コーパス) を用意する．各ソフトウェアは，動作プラットフォームおよび文法が同一であるアセンブリプログラムで記述 (アセンブリプログラムに変換) されており，かつ，難読化などのソフトウェア保護方法が適用されていないものとする．コンパイラによって出力されたままのコードを多く集めたものであれば，条件分岐の表現において比較命令 (cmp 命令) の後に条件付ジャンプ命令が続くことや，サブルーチンを呼び出す call 命令の前に mov 命令や push 命令が先行することなど，アセンブリコードの持つ一般的な性質 (文脈の傾向) をコーパスは持つものと考える．各アセンブリプログラムに対して，1 命令を 1 単語 (構成要素) とみなした N-gram の分析 (カウント) を行い，N-gram モデルを生成する．生成された N-gram モデルを用いて，与えられた評価対象のコード C の生成確率を推定し，コードの不自然さ $A(C)$ の評価値を得る．

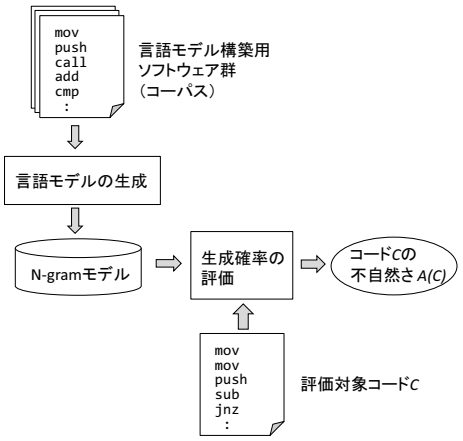

図 1　提案システム概要

(3) 評価システムを用いたケーススタディの実施

実装したシステムを用いて，既存の難読化法や最適化法が適用されたコードの不自然さを提案方法によって評価し，ステルシネスを考察するケーススタディを行った．ケーススタディで用いる N-gram モデルは，オープンソースの 2,030 個のソフトウェア (エディタ，コンパイラ，ゲーム等) のアセンブリコードをコーパスとして用いて構築した．評価対象のコードは，ソフトウェアのライセンスをチェックするルーチンや暗号化されたメディアを復号するルーチンを含む DRM のコンテンツ再生モジュールとした．

得られた結果の一例を図 2 に示す．図 2 は，元来のコード (難読化されていないコード)，静的難読化 (自己書換えを用いない難読化) が適用されたコード 4 つ，および，動的難読化 (自己書換えを用いる難読化) が適用されたコード 5 つについて，それぞれの正規化された不自然さ (測定対象コードの不自然さを同じ命令長を持つ一般的なコードの不自然さで割ったもの) を示している．ケーススタディの結果から，コードの制御構造や命令表現の変形のみを行う保護はコードのステルシネスを高く維持できる一方，意味のないコードを無作為に挿入する保護や元来のコードの意味を破壊する変形を伴う保護は，コードのステルシネスを低くする傾向が強いことがわかった．

上記 (1)〜(3) に述べた，N-gram モデルを用いてコードのステルシネスを評価する試みについて得られた成果は，5. の発表論文リス

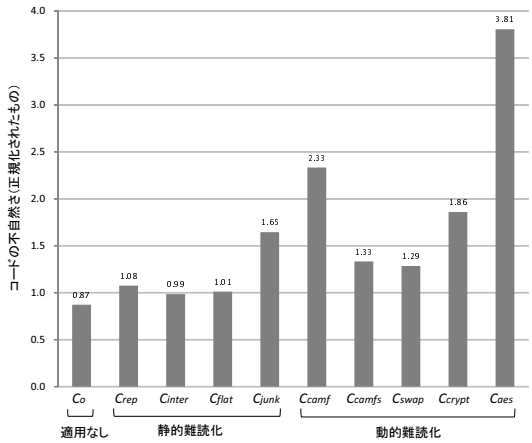

図 2　難読化されたコードの不自然さ

トの雑誌論文 [1]，および，学会発表 [1]，[2] において発表した．

(4) 他のアプローチの検討・その他の成果

ステルシネスを評価する別のアプローチとして，IDF(逆文書頻度) を用いた方法についても検討した．ある単語の IDF は，ある文書集合においてその単語が特定の文書に偏って出現する度合いを意味し，直観的には単語のめずらしさを表す指標であるといえる．IDF の定義における文書集合をプログラム集合，文書をプログラム，単語をアセンブリコード (命令列) と置き換え，IDF の大きさから，ステルシネスを考察する試みを行った．その成果は，学会発表 [6] において発表した．

また，ケーススタディ等に用いたプログラム難読化の方法に関して，学会発表 [4]，[5] などの成果を得た．加えて，雑誌論文 [2]，学会発表 [3] では，エントロピーやコルモゴロフ複雑性の概念を用いてソフトウェア解析の困難さを評価する方法を検討した．

5. 主な発表論文等

(研究代表者には下線)

〔雑誌論文〕（計 2 件）

[1] 神崎雄一郎，尾上栄浩，門田暁人．コードの「不自然さ」に基づくソフトウェア保護機構のステルシネス評価．情報処理学会論文誌，Vol.55，No.2，pp.1005-1015，2014 年 2 月．(査読有)

[2] 二村阿美，門田暁人，玉田春昭，神崎雄一郎，中村匡秀，松本健一．命令のランダム性に基づくプログラム難読化の評価．コンピュータソフトウェア，Vol.30，No.3，pp.18-24，2013 年 8 月．(査読有)

〔学会発表〕（計 6 件）

[1] 神崎雄一郎，尾上栄浩，門田暁人．N-gram モデルを用いたソフトウェア保護機構の不自然さ評価．2014 年暗号と情報セキュリティシンポジウム (SCIS2014) 予稿集 CD-ROM(講演番号 2D2-1), 2014 年 1 月 22 日，鹿児島県．

[2] 大滝隆貴，大堂哲也，玉田春昭，神崎雄一郎，門田暁人．Java バイトコード命令のオペコード、オペランドを用いた難読化手法のステルシネス評価．2014 年暗号と情報セキュリティシンポジウム (SCIS2014)，講演番号 2D2-2，2014 年 1 月 22 日，鹿児島県．

[3] 二村阿美，門田暁人，玉田春昭，神崎雄一郎，中村匡秀，松本健一．命令の乱雑さに基づくプログラム理解性の評価．ソフトウェア工学の基礎 XIV，日本ソフトウェア科学会 FOSE2012，Vol.19，pp.151-160，2012 年 12 月 14 日，大分県．(査読有)

[4] Hideshi Sakaguchi, Yuichiro Kanzaki, Akito Monden. Program Encryption Based on the Execution Time. Proc. International Symposium on Technology for Sustainability (ISTS2012), pp. 188-191, November 22, 2012, Bangkok, Thailand. (査読有)

[5] 坂口英司，神崎雄一郎，門田暁人．実行時間に依存したプログラムの暗号化．第 11 回情報科学技術フォーラム (FIT2012)，講演論文集 第 1 分冊 pp.257-260，2012 年 9 月 6 日，東京都．

[6] 尾上栄浩，神崎雄一郎，門田暁人．コードの「めずらしさ」に基づく保護機構のステルシネス考察．第 11 回情報科学技術フォーラム (FIT2012)，講演論文集 第 1 分冊 pp.261-266，2012 年 9 月 6 日，東京都．

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