マルウェアの動的解析ログにおける API コール列の文法圧縮

奥 村 嵩 宏[†]　大 山 恵 弘[†]

1. はじめに

近年、毎日大量のマルウェアが検出されている。例えば Kaspersky Lab では、2014 年末時点で毎日約 32 万 5 千個ものマルウェアが検出されている[1)]。そのような大量のマルウェアに対して自動化された動的解析を使用すると、大量の解析ログが出力されることになる。そこで、マルウェアの動的解析ログの圧縮が必要となる。

本研究では動的解析ログの中でも特に情報量が膨大となる API コール列の圧縮に着目した。API コール列とはマルウェア実行時の Windows API の呼び出し履歴のことである。そして、その API 名部分に繰り返しパターンが多いことから、API コール列の圧縮には文法圧縮が有効であると考えた。文法圧縮とは図 1 のように入力文字列を文脈自由文法に変換する圧縮法のことであり、繰り返しを多く含む文字列に対して有効である。

そこで、本研究ではマルウェアの動的解析ログにおける API コール列に対して文法圧縮が実際に有効であるかどうかを評価する。ただし、API コール列をそのまま入力文字列として文法圧縮した場合、繰り返しパターンを効率的に圧縮することができない。よって、本研究では API コール列を文法圧縮に適した入力文字列へと変換する手法の提示も行う。なお、評価方法は API コール列に対する文法圧縮と他手法の圧縮率の比較とする。

2. 実験手法

実験では、FFRI Dataset 2014[2)] の各動的解析ログから抽出した API 名のみの API コール列を、入力文字列に変換してから文法圧縮や他の圧縮手法によって圧縮した。FFRI Dataset とは FFRI 社が収集した Windows 向けマルウェアの動的解析ログのデータセットのことである。なお、実験では API コール列を持たない動的解析ログは除外した。また、文法圧縮には SEQUITUR[3)] と Re-Pair[4)] を、その比較対象とする他の圧縮手法には gzip 1.6 (GNU) と xz (XZ Utils) 5.1.0alpha を使用した。

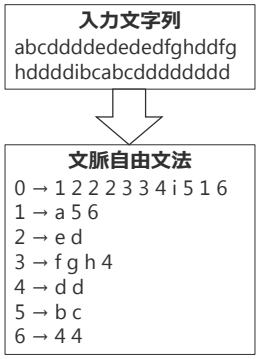

図 1　文脈自由文法の例

図 2　API コール列から入力文字列への変換例

実験にて API コール列を入力文字列に変換した手法は以下の通りである。まず、抽出した各 API コール列の間に区切り文字を挿入して連結した。そして、連結した API コール列に対して図 2 のように、各要素を 1 バイト文字に置き換えて入力文字列に変換した。このとき、API コール列の各要素とその置換文字の対応表を復元用に作成した。このようにして API コール列の各要素を 1 文字として扱うことによって、API コール列における繰り返しパターンを文法圧縮で効率的に圧縮できるようになる。

† 電気通信大学
The University of Electro-Communications

表 1 実験結果

入力文字列	gzip	xz	SEQUITUR	Re-Pair
34.99 MB	2.01 MB	0.55 MB	1.08 MB	0.62 MB
(100%)	(5.76%)	(1.58%)	(3.10%)	(1.77%)

3. 実験結果

Ubuntu 14.04 LTS (64 bit) 上で測定した実験結果を表 1 にまとめた。表 1 では、FFRI Dataset 2014 から抽出した API コール列を変換した入力文字列のサイズと、その入力文字列を各手法で圧縮した結果のサイズを示した。また、それぞれのサイズの下には入力文字列のサイズに対する割合を圧縮率として記載した。この実験結果では、SEQUITUR, Re-Pair による圧縮率は gzip による圧縮率より高かった。また、Re-Pair による圧縮率は xz による圧縮率より低かったものの、その差はたったの 0.19 %であった。

4. 関連研究

Walkinshaw らはプログラムのトレース結果における繰り返しパターンの識別用に SEQUITUR を使用した[5]。彼らの SEQUITUR の使用目的が生成した文脈自由文法の可視化であるのに対し、本研究の使用目的は圧縮である。なお、本研究の実験では、彼らの SEQUITUR の使用方法を参考にして API コール列を入力文字列へと変換している。

Larus はプログラムの制御フロー全体の新しい表現として Whole program paths を提案し、その生成処理の過程で SEQUITUR を使用した[6]。圧縮に SEQUITUR を使用しているものの、圧縮対象が制御フローである点や生成した文脈自由文法をグラフで表現している点などが本研究とは異なる。

5. まとめ

本研究では、マルウェアの動的解析ログにおける API コール列を、適切な入力文字列に変換してから文法圧縮する手法を提示した。さらに、その手法を用いた実験結果より、文法圧縮で API コール列の圧縮率が高くなる場合があることがわかった。

今後の課題としては主に次の 2 つが考えられる。1 つ目は API の引数情報を付加することである。今回の実験では API 名しか使用していない。2 つ目は、圧縮パターン照合の利用である。圧縮テキスト上でのパターン照合に特化した文法圧縮が API コール列に対して利用可能であると示すことができれば、圧縮したまま API コールを検索できるようになると考えられる。

参考文献

1) Kaspersky Lab, Kaspersky Lab is Detecting 325,000 New Malicious Files Every Day — Kaspersky Lab. http://www.kaspersky.com/about/news/virus/2014/Kaspersky-Lab-is-Detecting-325000-New-Malicious-Files-Every-Day
2) 秋山 満昭, 他. マルウェア対策のための研究用データセット〜MWS Datasets 2014〜. 研究報告コンピュータセキュリティ（CSEC）, Vol. 66, No. 19, pp. 1-7, 2014.
3) Craig G. Nevill-Manning and Ian H. Witten, Sequitur. http://www.sequitur.info/
4) re-pair. https://code.google.com/p/re-pair/
5) Neil Walkinshaw, Sheeva Afshan, and Phil McMinn. Using Compression Algorithms to Support the Comprehension of Program Traces. *Proceedings of the Eighth International Workshop on Dynamic Analysis*, pp. 8-13, ACM, 2010.
6) James R. Larus, Whole Program Paths. *Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 259-269, ACM, 1999.