類似マルウェアとの差分の抽出による正確な挙動の解析手法についての検討

羽田 大樹[†]　　後藤 厚宏[†]

情報セキュリティ大学院大学[†]

1. はじめに

政府や企業におけるマルウェア感染の脅威がますます増大している. 近年のサイバー攻撃の傾向としては, APT (Advanced Persistent Threat)や標的型攻撃に代表されるように攻撃が高度化し, 被害の大規模化や深刻化が問題となっている. これらに対して絶対に侵入されないよう確実な対策を行うことは現実的でなく, 侵入されることを前提として事後の対応における調査を迅速かつ確実に行うことも考えておかなければならない. 軍事機関や政府, 大企業など重要インフラにおいて被害を受けた場合, 単純にシステムを復旧すれば良いということはなく, 侵入経路や挙動, 影響範囲を特定するためフォレンジックなどによりマルウェアの挙動を正確に把握する必要がある.

マルウェアの挙動を正確に解析するためには, 実行ファイルを逆アセンブルして実行コード列を読み解く静的解析という作業が必要となる. これは熟練した技術者が時間をかけて行う必要がある. ここで, すでに類似するマルウェアの挙動を正確に解析した結果を保有している場合, この結果を用いて解析作業を簡略化することが考えられる.

2. マルウェア解析の手法と課題

マルウェア解析の一般的な手法と課題について述べる.

動的解析: 解析専用環境などを利用してマルウェアを実際に動作させ, ファイルシステムやレジストリへの変更やネットワーク通信を観測することでマルウェア挙動を解析する. 解析されていることを検知して動作を停止するマルウェアや, ボットのような攻撃者からの指令無しには動作しないマルウェア, ある特定の時刻にしか動作しないマルウェアに関して, その挙動を網羅的に抽出することは難しい.

静的解析: 動的解析では調査できない挙動については実行コードを直接読み解くことが必要となる. 圧縮や暗号化などを利用して実行コードを隠蔽(パッキング)している場合, まずパックされたコードからオリジナルコードを復元し, 機械語を逆アセンブルした後に実行コード列から挙動を解析する. 詳細な挙動を解析できる反面, 技術者のスキルと多大な時間を要するという問題がある.

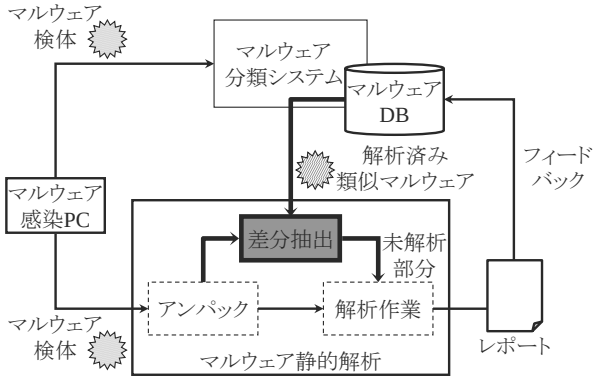

図 1 マルウェア静的解析の提案フロー

静的解析におけるこの問題を解決するため, 我々はマルウェア解析において図 1 に示すフローを提案し, 類似する 2 つのマルウェアから差分を抽出する方式を議論する. 静的解析の際に解析対象マルウェアと類似する解析済みのマルウェアを特定し, 解析済みのマルウェアとの差分を抽出することで解析対象マルウェアの未解析部分のみを解析対象として解析作業にかかる時間を削減することを考える.

3. 差分抽出の実現方式

本研究では, 解析対象とするマルウェアと, 静的解析が完了した類似マルウェアを入力として, 解析対象マルウェアの未解析部分のみを出力するシステムの構築を検討する. このシステムは以下の要件を満たすべきと考えられる.

① マルウェア分類システムが存在し, 類似する解析済みマルウェアをデータベースから選択して解析データとともに出力する.
② 入力とする 2 つのマルウェアのアンパックと逆アセンブルが完了している, すなわちアセンブリ言語で記述された実行コード列が取得できている.
③ 類似するマルウェアの差分について, 単純な実行コード列の差分ではなく, 機能的に本質的に異なる箇所を抽出できる. 例えば同じソースコードであってもコンパイルオプションの違いによって異なる実行コード列を出力するが, 本質的には同じであると見なすべきである.

マルウェアの差分を抽出するためのアイディアとしては大きく分けて 2 つの手法が存在する. 岩村らは実行コード列の最長共通部分列(LCS)の長さを用いて類似度を定義してマルウェアの分類と差分を抽出する手法を提案している[1]. この手法は大量のマルウェアを効率的に分類することを目的としているため, 非常に高速に動作する.

一方で, 実行コード列ではなくプログラムの制御構造に着目し, これをグラフ化することでプログラムの構造の違いを把握する手法がある. グラフを利用することで, 例えばプログラム中の関数の順番が入れ替わった場合でも同一と見なすことができる. Flake らは, プログラムのパッチ解析などのリバースエンジニアリングを目的としてアセンブリコードをコールグラフと制御フローグラフという 2 つのグラフの入れ子構造で表現して比較する手法を提案している[2][3]. コールグラフとは, アセンブリコードを関数単位で分割してノードとし, 呼び出し関係を有向エッジで表したものである. コールグラフの各ノードはさらに jmp, jnz などの条件分岐命令, loop などのループ命令, call 命令, ret 命令などのような制御命令でベーシックブロック単位に分割され, 制御フローグラフとして表される. さらにコールグラフの各ノードに対して, 特徴量をベーシックブロックの数, ノード内の辺の数, ノード間の辺の数を 3 次元座標として特徴量とし, 2 つのプログラム間において各ノードのユークリッド距離が小さくなる写像を構成することで構造上の差分を抽出する. Flake らはこの提案方式を BinDiff というソフトウェアとして実装している[4].

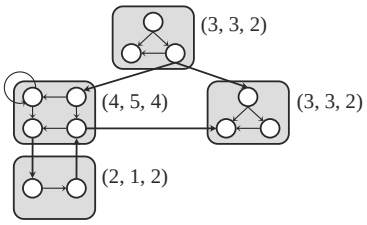

図 2 コールグラフ/制御フローグラフと特徴量

4. 検討結果と課題

グラフを利用した実行コード列の比較を利用することで, 2 つのマルウェアにおける関数の追加や削除, 変更点を抽出することができ, 一定の有効性を示すことができた.

一方で, フォレンジックの場面においてマルウェア解析を効率的に実施する上で, さらに以下のような課題が挙げられる.

① マッチング精度の向上

BinDiff はプログラムのパッチ解析を主な目的としたソフトウェアであり, マルウェア解析に特化したマッチングを考えることができる.

② メタモーフィックマルウェアへの対応

メタモーフィックマルウェアは感染するたびに自分自身の挙動を変えずに実行コード列を変化させる. この変更を差分として抽出せずに同一とみなすことでさらに解析にかかる時間を削減することができる.

③ 機能の比較

実行コード列の差分が大きいことが, すなわち挙動が大きく異なることを直接には意味しない. 例えば API コールやファイル I/O など特定の処理の追加がフォレンジックにおいて大きな意味を持つ場合がある. 優先的に着目すべき箇所を示すことで解析作業を効率化できる.

5. まとめと今後の予定

大量に作成されるマルウェアの亜種を広く網羅的に把握できるよう, 近年のマルウェア解析の研究テーマとしては, いかに作業を自動化して高速かつ大量に処理できるかを考えることが主流である. 一方で我々は, 重要インフラが狙われる今日ではフォレンジックにおいて特定のマルウェアに焦点を当てた厳密な解析がこれまで以上に重要になってくると考えているため, 解析作業の完全な自動化を目指すのではなく, 熟練した技術者が行う解析作業を効率化する方法を検討した.

ただし, 我々が提案する静的解析フロー(図 1)において差分を抽出するためには, マルウェアの分類, アンパック, 逆アセンブルが高い精度で実現可能であることを前提としており, 自動解析と手動解析の両面から研究を進めていく必要があると考える.

今後は差分抽出の具体的な実現方式と実装について研究を行う予定である.

参考文献

[1] 岩村誠, 伊藤光恭, 村岡洋一. 機械語命令列の類似性に基づく自動マルウェア分類システム. 情報処理学会論文誌 Vol. 51 No. 9 1- 11, 2010.
[2] H. Flake. Structural Comparison of Executable Objects. DIMVA, 2004.
[3] T. Dullien, R. Rolles. Graph-based comparison of Executable Objects. SSTIC, 2005.
[4] Zynamics BinDiff http://www.zynamics.com/bindiff.html

A Study of Extracting the Difference Between the Same Kind of the Malwares for Forensic.
Hiroki Hada†, Atsuhiro Goto†.
† Institute of Information Security.