Automated deduction system を用いた マルウェア外部観測ログ解析の自動化

安藤類央 †　　門林雄基 †　　篠田陽一 †

† 情報通信研究機構
〒 184-8795 東京都小金井市貫井北町 4-2-1

ruo@nict.go.jp

あらまし　本論文では、automated deduction system を用いたマルウェアの外部観測ログの解析の自動化を提案する。Windows OS 上でのマルウェアの挙動の各種リソースアクセス等の外部観測ログを、一階述語論理ベースの定理証明によって処理することにより、解析の自動化の手法の提案と評価実験の結果を提示する。提案システム上では、感染動作から生成される代表的なログを節形式で表現し、外部観測ログとの導出により、マルウェアの挙動を検出する。本手法により、異なる環境や時間帯による実行によるログの差異を捨象し、外部観測ログの中から感染動作を抽出することができる。

An analysis of malware's behavior using automated deduction system

Ruo Ando†　　Youki Kadobayashi †　　Yoichi Shinoda †

†National Institute of Information and Communications Technology,
4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795 Japan

Abstract　In this paper we propose an analysis of malware's behavior using automated deduction system. In our system the extenal log of malware's behavior is analyzed by FoL (First order Logic) based theorem prover. The some pattern to be discovered is describles as counterexample which deduced from log of Windows OS on which malware was executed. By proposed method we can find the patterns regardless of noise and specification of Operating systems and malicious softwares.

1　はじめに

近年、Malware (Malicious Software) の数は増加し続けている。Malware による攻撃自体が現象的に新しかった時期は、Malware の数が少なかったため、解析の方法やノウハウは、一部の専門家によって所有され解析が行われている状況であったが、近年の Malware の増加は、解析手法の自動化の需要を高めている。本論文では、Automated deduction system を用いたマルウェア外部観測ログの解析の自動化を提案する。Automated deduction system には、一階述語論理の定理証明系を採用し、Malware の外部観測ログを CSV ファイルとして保存し、節形式に変換し、発見したい挙動パターンを反例として記述することで、解析を自動化する。

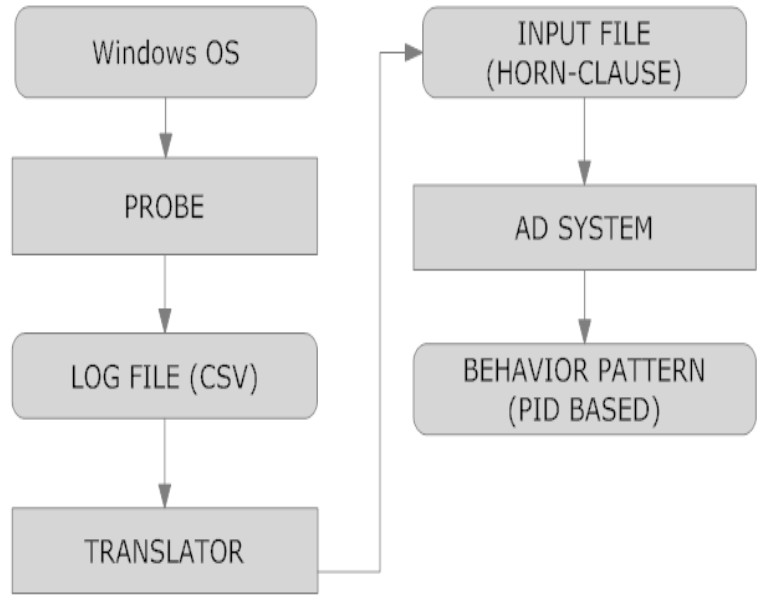

図 1: 解析の自動化の流れ。Windows OS に観測用の probe を埋め込み、ログファイルを出力する。これを節形式に変換し、自動定理証明器に処理させることで、発見したい挙動パターンのログファイル中の有無を判定する。

2　提案手法

図１に解析の自動化の流れを示した。提案システムでは、Windows OS に観測用の probe を埋め込み、ログファイルを出力する。これを節形式に変換し、自動定理証明器に処理させることで、発見したい挙動パターンのログファイル中の有無を判定する。処理対象ファイルは、外部観測ログ、発見したいパターンの表現、導出規則の３つからなり、次節で述べるアルゴリズムを適用して充足可能性を判定する。導出された節表現群と反例がコンフリクトを起こせば挙動パターンが発見され、導出過程での各種パラメータが参照され、解析が終了する。

3　適用アルゴリズム

3.1　支持集合戦略

支持集合戦略 は、１９６５年に Wos らによって提案されたものである。この計算戦略は制限戦略の１つで、自動推論プログラムに目標とする解空間に関係ないところを探索せずに、対象としている問題に集中させるようにする。節集合 S、T があり、S-T が充足可能であるとき、T は S の支持集合である。このとき、支持集合に属さない節同士では導出を行わず、支持集合に属する節との間で、導出を行う方針を支持集合戦略という。

3.2　超導出

超導出 は１９６５年に Robinson らによって提唱された手法で、通常の導出系の手法では１対の節から順次導出を行うのに対して、２個以上の節に対して導出を行う。超導出の意味は、何段階もの２項導出にあたる作業を１つにまとめたもので、通常の２項導出に比べて、多くの導出が起こるという事を指す。

3.3　包摂

定理証明を用いた推論プロセスでは、目標とする節を導出する過程で、いくつかの節が保持

され、新しい節が生成された時点で、過去に保持された節との間で、改めて定理が適用される。この保持されている節のうち、より一般的な節を残す処理を包摂という。

3.4 デモジュレーション

デモジュレーションとは、あらかじめ等価代入を行うための節を定理証明系に加えて、処理節群の簡略化あるいは正準化を行う処理である。デモジュレーションは、それ自体で冗長な命令、あるいは、MOV 命令によるメモリ、レジスタ、変数間の転送によるプログラムの状態遷移を簡略化するのに有効である。

4 Windows OSの修正

4.1 フィルタドライバ

Microsoft Windows のフィルタドライバは、ＸＰから積極的に導入、活用が始まったソフトウェアモジュールである。フィルタドライバは、ＩＯマネージャとカーネルドライバの間に位置して、ファンクションドライバ（既存のデバイスドライバ）の前後に既存の Windows が提供する機能を利用して呼び出され、新しい機能を追加したり、機能修正、デバッグなどを行うソフトウェアである。フィルタドライバを用いることで、Native API フックを行うことができる。Native API Hook は、カーネルモードでのイベント検出に用いられる。ユーザモードでの API 発行は、ntdll.dll 経由でカーネル空間に伝達される。Windows では、カーネルモードで動作する API を、native API といい、これらは、SystemServiceDescriptorTable という構造体によって、制御されている。そのため、Native API Hook を行うためには、SystemServiceDescriptor に修正を加える必要がある。

4.2 DLL Injection

開発環境や稼動する状況にも依存するが、通常のフィルタドライバを用いたファイルＩＯのフックは、安定動作しない場合がある。フィルタドライバは、Microsoft が新たに提供をはじめたファイルシステムフィルタドライバである。フィルタドライバは、大量に呼び出される割り込みやＡＰＩに関して、順序などを適切に制御し、サードパーティの開発を簡素化、支援することを主眼に設計されている。

4.3 DLL Injection

DLL Injection とは、任意の関数をライブラリ化して特定の API が発行されたときに、実行させるもので、既存のアプリケーションに新しい機能を加えたいとき、ソフトウェアのデバッグの際に用いられる。開発の要請から、DLL (独自 API) を他のプロセスに任意のタイミングで実行させる DLL Injection という技術が用いられることがある。DLL Injection の方法には、Microsoft Windows が提供している機能を使うもの、デバッグ機構の機能を使うもの、リモートスレッドを使うもの、そしてモジュールのインポートセクションの変更によるものなどがある。本論文では、モジュールのインポートセクションの変更による方法により、対象ソフトウェアのデータ送受信関数をフックし、ここに適切な操作を入れることにより、ソフトウェアの挙動を追跡し、問題となるトラフィックが発生または到着する前に防止を行うことを可能にした。

5 評価実験

今年度配布された検体のうち数体を用いて、提案システムの評価実験を行った。Windows OS のライブラリの修正やフィルタドライバを用いて検体の挙動の外部観測を行い、これによって得られたログを Automated deduction system によって解析した。表１は、ログ採取時にインターセプトしたＡＰＩのリストである。発見したパターンは、ネットワーク機能の利用の観点から正規とはみなされない socket のライブラリのロードとソケットアクセスをしたプロセス（あるいはＰＩＤ）が、同時にシステムフォルダへのファイル生成、不正なレジストリアクセ

スを行った際の挙動パターンを反例として記述し、検出を行った。

6 まとめと今後の課題

本論文では、automated deduction system を用いたマルウェアの外部観測ログの解析の自動化を提案した。Windows OS 上でのマルウェアの挙動の各種リソースアクセス等の外部観測ログを、一階述語論理ベースの定理証明によって処理することにより、解析の自動化の手法の提案と評価実験の結果を提示した。提案システム上では、感染動作から生成される代表的なログを節形式で表現し、外部観測ログとの導出により、マルウェアの挙動を検出する。本手法により、異なる環境や時間帯による実行によるログの差異を捨象し、外部観測ログの中から感染動作を抽出することが可能になった。

参考文献

[1] Larry Wos, George A. Robinson, Daniel F. Carsonm "Efficiency and Completeness of the Set of Support Strategy in Theorem Proving", Journal of Automated Reasoning, 1965.

[2] Larry Wos: The Problem of Explaining the Disparate Performance of Hyperresolution and Paramodulation. J. Autom. Reasoning 4(2): 215-217 (1988)

[3] Larry Wos: The Problem of Self-Analytically Choosing the Weights. J. Autom. Reasoning 4(4): 463-464 (1988)

[4] Larry Wos: The Problem of Choosing the Type of Subsumption to Use. J. Autom. Reasoning 7(3): 435-438 (1991)

[5] Larry Wos, George A. Robinson, Daniel F. Carson, Leon Shalla, "The Concept of Demodulation in Theorem Proving", Journal of Automated Reasoning, 1967.

[6] Larry Wos, "The Power of Combining Resonance with Heat", Journal of Automated Reasoning, 1996.

[7] Diomidis Spinellis, "Reliable identification of bounded-length viruses is NP-complete", IEEE Transactions on Information Theory, 2000.

[8] Ruo Ando, Yoshiyasu Takefuji, "Faster resolution based metamorphic virus detection using ATP control strategy", WSEAS TRANSACTIONS ON INFORMATION SCIENCE AND APPLICATIONS, Issue 2, Volume 3, February 2006 ISSN 1709-0832,pp260-2266,February 2006

[9] Ruo Ando, "Faster parameter detection of polymorphic viral code using hot list strategy", ICONIP 2008 - 15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly, November 25-28, 2008, Auckland, New Zealand

[10] Ruo Ando, "Parallel analysis of polymorphic viral code using automated deduction system", 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD2007), July 2007, Qingdao, China

API name	API type A	API type B
AllocVirtualMemory	Native API	Memory allocation.
CreateSection	Native API	Memory section creation .
LoadLibrary	User API	Invoking library.
MapViewOfSection	Native API	Memory Allocation.
PreRead	filter function	File Access.
PreWrite	filter function	File Access.
ZwCreateEvent	Native API	Memory Allocation.
ZwOpenKeys	Native API	Register Access.
CreateProcess	User API	Process Operation.
ZwOpenKeys	Native API	Registry Access.
WSARecv	User API	Socket Access.
Send	User API	Socket Access.

表 1: 評価実験でフックしたＡＰＩのリスト

	検体１	検体２	検体３
処理ファイル行数	7813	5812	10796
利用メモリサイズ	27443	19351	39063

表 2: 実験結果（処理サイズとメモリ）

	検体１	検体２	検体３
生成された節数（発見時）	382	374	6104
生成された節数（未発見時）	15626	5819	10733

表 3: 実験結果（生成された節数）

	検体１	検体２	検体３
clauses generated	382	367	6104
res generated	13	8	30
demod generated	357	366	6074
clauses, wt deleted	353	363	0
clauses subsumed	23	6	6099
sos size	15274	5460	4636

表 4: 実験結果（定理証明系の出力）