自己組織化マップを用いた異常検知についての一検討

An Examination of Anomaly Intrusion Detection using Self-Organizing Maps

柿本 圭介† 田中 英彦†
Keisuke Kakimoto Hidehiko Tanaka

１．はじめに

ブロードバンドインターネット環境の普及に伴って、情報通信基盤がますます重要な社会インフラとなる一方、ワームやウィルスなど（以下、悪意のあるプログラムを総称してマルウェアと呼ぶ）によるネットワークへの被害の拡散はより広範囲、かつ急速なものとなっている。マルウェアの数、種類は年々増加しており、オリジナルの一部を改変した亜種の急速な感染拡大や、自らプログラムの一部を変更するマルウェアの出現によって、データのバイト列を攻撃のシグネチャと単純にマッチングする方式によるセキュリティシステムでは検知できないケースが増えてきている。このような攻撃を検知するための有効な対策の一つに、プログラムの動作の監視による異常検知がある。

侵入検知は、分析手法によって不正検知と異常検知に分類されるが、不正検知が既知の攻撃のシグネチャをパターンマッチングすることにより検知を行うのに対し、異常検知はシステムやユーザの通常の状態を定義しておき、通常でない状態を閾値により判別して攻撃を検知するため、未知の攻撃に対して有効な手法である。

本研究では、未知の攻撃を検知することを目的とした異常検知において自己組織化マップを適用し、分析対象とするデータとして、保護対象となるマシン上のシステムコールを監視した情報を用いる方式についての検討を行う。

２．自己組織化マップとは

2.1 概要

自己組織化マップ（Self-Organizing Maps、以下 SOM とする）は、データマイニング手法の一つであるが、多次元のデータの統計的性質を学習し、類似した性質を持ったものどうしが近接するように低次元化を行い、2 次元平面上への写像を可能とする手法である。このような特性をもった SOM は、予測を行う問題において有効であると考えられている。1981 年に Kohonen[1]により教師なしのニューラルネットワークアルゴリズムとして提案されたのが最初であるが、その後教師ありのクラス分類が行えるように拡張したものや、時系列情報を扱うことができない性質を改善したものなど、様々な拡張版が提案されている。

2.2 異常検知に適用する上での利点・欠点

本研究では、異常検知を行うための手法として SOM を適用するが、その上での利点、欠点を以下に挙げる。

（1）利点

- SOM において、多次元データを 2 次元平面上に写像するということは、一種のデータ圧縮であり、様々な属性を持った大規模な入力データの要約が行える。
- SOM は、類似した性質を持ったものどうしが近接するように低次元化を行うため、既知のデータと未知のデータを入力データとして低次元化を行った場合、未知のデータの性質を近接する既知のデータから類推することができる。
- SOM はお互いに独立したパターン系列を解析するように設計されているため、入力データの表現方法について寛容である。すなわち、n 次元の入力データは、n 個の属性を持つといえるが、その属性を任意に選択できるため、データ空間の非線形な特徴を取り出すことが可能となる。

（2）欠点

- SOM は、学習したデータ空間を要約しているだけであり、学習したデータ空間から外側へ大きく外れる新しいケースについては、正しい判断を導くとは限らない。異常検知に SOM を適用する場合、学習段階で正常な状態とするデータ空間を網羅的に取り込めていないと誤検出(false positive)が多く発生してしまう可能性がある。
- 学習に時間を要するため、リアルタイムでの検知を考えた場合、予めマップを生成しておく必要がある。
- SOM により分類した結果は、上述のように正しいとは限らないため、正確な分類状態を自動的に保っていくためには教師ありの SOM を適用していくことを検討していく必要がある。

3. 関連研究

SOM をホスト型の異常検知に適用した研究として、Wei ら[2]の研究がある。Wei らは、正常時に呼び出されるシステムコールを入力データとして、SOM と隠れマルコフモデルを適用し、検知の正確性とパフォーマンスの両面で比較を行っている。SOM を適用した場合、システムコールの頻度情報が入力データとなるのに対し、隠れマルコフモデルを適用した場合はシステムコールの呼び出し順が入力データとなるという違いがある。実験の結果、隠れマルコフモデルに比べて検知の正確性ではやや劣っているが、パフォーマンスにおいては勝っているため、リアルタイムでの検知には SOM の方が適しているとしている。

システムコールの呼び出し列を侵入検知に用いる研究は、主に UNIX 系 OS を対象として多数行われてきているが、その先駆的な研究として Forrest ら[3]の研究が挙げられる。Forrest らは、特定のアプリケーションの通常時の動作を N-gram 法により定義しておき、不正な攻撃を受けた際の動作との比較により異常検知を行う手法を提案した。

N-gram 法でシステムコールの呼び出し列を特徴化する異常検知を Windows に適用した研究としては島本ら[4]の研究がある。島本らは、カーネルレベルで呼び出される API（System Service）を監視して得られた呼び出し列を N-gram 法により特徴化する手法を用いているが、実験による評価の結果、効果的な異常検知が実現可能であることを示す測定結果が得られている。

† 情報セキュリティ大学院大学情報セキュリティ研究科

4. 検討方式

4.1 検討方式の特徴

本研究では、ホスト型の異常検知を行うための手法として SOM を適用し、入力データとしてマルウェアから呼び出されるシステムコールの呼び出し列を用いる方式を検討する。SOM を適用する理由は、2 章で述べたように互いに独立した属性を様々なバリエーションを持たせて選択することができ、それによりデータ空間の非線形な特徴を取り出すことが可能となるためである。マルウェアの挙動は、その種類によって多岐に渡るため、様々な動作を特徴化するために SOM は有効であると考える。

次に SOM への入力データとしてシステムコールの呼び出し列を用いる理由であるが、3 章で述べた研究からシステムコールの呼び出し列を異常検知に用いる有効性が実証されているためである。しかし、その研究における SOM への入力データは頻度情報であり、通常の SOM では時系列情報を扱うことができない。そこで、本研究では、N-gram 法を用いてシステムコールの呼び出し列を抽出し、その頻度情報を入力データとして与える方式により時系列情報を取り込む。

4.2 検討方式の構成

本方式は、以下の 3 つのシステムからなる。

- マルウェア収集システム
- マルウェア自動実行システム
- マルウェア解析システム

マルウェア収集システムでは、ローインタラクション型のハニーポットを用いて検体の収集を行う。ハニーポットは、ハイインタラクション型とローインタラクション型に分類されるが、前者が侵入可能な実際のオペレーティングシステム全体とアプリケーション全体を提供するのに対して、後者はシステムやサービスをエミュレートすることによって機能するものである。今回、実装が容易でかつ運用も安全に行えるため、後者を用いたが、前者の方がより詳細な情報が得られること、また、エミュレートした環境を検出して回避するようなマルウェアも存在することから前者を用いることも検討していく必要がある。

マルウェア自動実行システムでは、収集したマルウェアを実行するための仮想環境を構築し、実行後にマルウェアから呼び出される API をトレースした。マルウェアを実行した仮想環境における OS は WindowsXP で、API のトレースには、Strace for NT を用いてカーネルレベルで呼び出される API のトレースを行った。

マルウェア解析システムでは、トレースした API の情報から N-gram 法によってある任意の数 N の API 呼び出し列を抽出する。たとえば、トレースした API が[A,B,C,D,A,B,C]である時に N=3 として N-gram 法で抽出される呼び出し列は、[A,B,C]、[B,C,D]、[C,D,A]、[D,A,B]となる。この際、全ての呼び出し列を抽出するとその数が膨大になり、SOM への入力データとして有用でない情報も含まれると考えらたため、以下に示すマルウェアの挙動によって呼び出される API が含まれる呼び出し列のみを抽出対象とした。

- ファイル作成・削除・改ざん
- レジストリ登録・削除・改ざん
- ポート接続

前述の例において D のみが上記の挙動によって呼び出される API であるとすると、[A,B,C]は抽出対象外となる。

次にこの方法によって、抽出した呼び出し列が一つのマルウェアを一定時間実行した際に何回呼び出されるかという頻度情報を複数のマルウェアから抽出し、これを SOM への入力データとする。

本方式により SOM を実行したところ、以下のような結果が得られている。

- 2007 年 6 月 15 日時点で ClamAV によるウィルススキャンにおいて検出できなかったマルウェアと、検出できたマルウェアとの間で入力データの特徴が類似しており、SOM により出力されたマップにおいて互いに近接しているものが存在した。検出されなかったマルウェアがウィルス定義ファイルの更新により、検出されたマルウェアと同種のものとして検出されることが確認されれば、本方式が未知のデータの性質をマップ上近接する既知のデータから類推できる性質を持つということが言える。
- 同種のマルウェア同士で入力データの特徴が類似しており、マップ上近接しているものもあるが、入力データの特徴が異なり、マップ上近接していないものも存在したため、様々な種類のマルウェアをマップ上近接するように正しく分類するためには入力データの与え方について更に検討していく必要がある。

5. まとめと今後の課題

本稿では、異常検知において SOM を適用し、その入力データとしてシステムコール呼び出し列の頻度情報を用いる方式について検討した結果を述べた。

今回、マルウェア実行時のシステムコール呼び出し列のみを解析対象としたが、異常検知に SOM を適用するためには、特定のプロセスについて正常時のシステムコール呼び出し列と異常時のシステムコール呼び出し列を解析し、その比較を行う方式を検討していく必要がある。更に、その際に解析対象とする情報としてシステムコール呼び出し列のみではなく、システムコールへの入出力パラメータについても取り込む方式についての検討を行う予定である。

また、SOM により適切な分類を行うためには、別々に分類されるべきデータごとにその特徴が異なっている必要がある。そのために、異なる特徴が現れているデータについては重み付けを行い比重を高くするといった調整方法についての検討も行っていく必要がある。

参考文献

[1] T.コホネン, 自己組織化マップ 改訂版,シュプリンガーフェアラーク東京,(2005).

[2] Wei Wang, Xiaohong Guan, Xiangliang Zhang, Liwei Yang, “Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data,” Computer & Security 25, pp. 539-550(2006).

[3] Forrest S., Hofmeyr S.A., Somayaji A., Longstaff T.A., “A sense of self for Unix processes,” In: Proc. 1996 IEEE Symp. On Research in Security and Privacy, pp. 120-128(1996).

[4] 島本 大輔, 大山 恵弘, 米澤 明憲, “System Service 監視による Windows 向け異常検知システム機構,” 情報処理学会論文誌 コンピューティングシステム, Vol. 47, No. SIG 12(ACS 15), pp. 420-429(2006).