3ZC-02　モバイルワクチンシステムの学習機能の構築と検証

佐藤 信介　千石 靖　服部 進実
金沢工業大学情報工学科

1. はじめに

モバイルワクチンシステム[1]とは、ワクチン機能を持ったエージェントがネットワーク内のコンピュータを移動しコンピュータウイルスの発見、駆除を行うシステムである。様々な手間をサーバー一つで行うことが出来る。

そのシステムでは、ウイルス検出法の一つにヒューリスティック検査法[3]を用いており、その検出法に必要なデータベースの更新をウイルス検出に最適な形で行うことが出来れば、その特性を最大限に生かすことが出来る。

2. ヒューリスティック検査法

ヒューリスティック検査法とは、既存のウイルスからそれをウイルスだと特徴付けるコードを収集しておき、検査対象プログラムのウイルスらしさを判定するアルゴリズムである。また、判定した結果により自分でウイルスコードを収集し、データベースの更新を行う。そのため、亜種や新種のウイルス検出に優れており、新種登場のたびに更新を余儀なくされた従来のウイルス対策ソフトウエアに比べ、手間もかからない。

ただし、ウイルスプログラムそのもののマッチングではなく、それに含まれるウイルスの特徴的なコードで判断するため、非ウイルスプログラムをウイルスだと誤認する可能性がある。これは、データベースの影響によるところが大きい。

3. データベース

先述したように、ヒューリスティック検査法を用いる場合の問題点として、データベースの更新がある。ウイルスコード収集の際の基準、各コードごとの重み等である。重みとはウイルスらしさであり、以下危険度と表記する。これらを不適切に行うとウイルス検出の精度が落ち、誤認が起こる。

本研究で作成したデータベースは、大まかにウイルスの特徴的なコードとその危険度からなり、検査対象プログラム全体の危険度によりウイルスか否かを判定する。この判定を行う処理エンジンを検査エンジン、データベースを適切に更新し、精度や効率の向上を図る処理エンジンを学習エンジンと呼ぶ。検査エンジンがウイルスらしいと判断した検査対象プログラムのコードは、学習エンジンに送られ、既知コードか否かで危険度の更新又はデータベースへの追加を行う。危険度はウイルスらしさを数値化したものであり、その決定方法が学習エンジンの最大の課題となる。

4. 学習エンジン

4.1　ウイルスらしさとは

コンピュータの命令はウイルス作成のために用意されたのではない。しかし、組み合わせや操作の対象によりそれを可能にしている。それが、ウイルスらしさである。しかし、亜種や新種に対応するために、プログラム全体ではなくコード一つ一つ(命令とオペランドも別)をデータベース化しているため、別のウイルスらしさを考える必要がある。

ここで、何度も学習エンジンに送られてくるコードは、頻繁にウイルスらしい構造に用いられていると考えられる。よって、発見のパターンを一つの基準としてみる。

4.2　発見パターンによるウイルスらしさ

単純に件数が多ければ危険かといえば、そうではない。

Design of Learning Engine for the Mobile Vaccine System
Shinsuke Sato, Yasushi Sengoku, Shimmi Hattori
Information and Computer Engineering, Kanazawa Institute of Technology
7-1 Ohgigaoka Nonoichi Ishikawa 921-8501, Japan

ウイルスの危険性の一つに感染力がある。ここ数日間に5回発見されるのと、一年で10回発見されるのでは、明らかに前者が危険であり、なおかつウイルスらしい。

また、危険度の決定を頻度でするのもいただけない。かつて大繁殖し、以来大きく衰退したウイルスのコードの頻度というのは、ウイルスらしさに比べ危険度が低くなり、ウイルスプログラムを見過ごす可能性が高い。

これらの問題は、発見の履歴を保存しておくことで解決できる。しかし、データベースが膨らみすぎる可能性があり、それに伴い処理が重くなる。学習機能の構築は、検査を精度と効率の両方から捉える必要がある。

4.3 危険度の決定

そこで、履歴の簡略化について考えた。コードの危険度を、「初回発見日時を原点とし、現在をx、単位時間ごとに減少していき発見時に上昇する値をyとしたときの点(x、y)を通る直線の傾き」とする。これで、各コードの持つデータは、初回発見日時、現在のyの値があればよいことになる。

しかし、発見時のyの上昇値が常に等しい場合、危険度の上限及び下限を超えない限り、発見回数と初回発見日時があれば同じ値を算出できる。これでは頻度を持っているのと本質的な違いがない。

これは、連続もしくはそれに近い間隔で発見された場合のyの上昇値を通常より高くしておけば解決でき、一気に感染しワクチン等の対応により徐々に衰退していくといった、ウイルスらしい発見パターンの場合に、比較的高い危険度を保つことが出来る。

5. 検査エンジンとの連携

今回、検査エンジンはVisual Basic、学習エンジンはVisual C++で組まれている。そのため、学習エンジンはDLL化してあり、実行時に検査エンジンに読みこまれる。

まず、検査側で確保したメモリ領域のアドレスを受け取り、その領域に必要なデータを格納しておく。これは、DLL化した学習エンジンがメモリを保持できないためであり、その領域は基本的に検査エンジンは使用しない。

その後、必要なデータを検査エンジンに渡し、そのデータにより怪しいと判断された検査対象プログラムのコードが学習エンジンに渡され、その危険度を検査エンジンに返す。このとき危険度を上げる(実際に操作を受けているのは前述したyの値で、危険度はその都度算出する)。

検査が終了したら、その旨が検査エンジンから伝えられ、未発見のコードの危険度を下げる。

6. 問題点と解決策

危険度の上昇値、減少値、初回発見時の危険度等の最適な値の検証が現時点でまったく出来ていない。最高の効果を得るためには、これらの値を最適値に設定しなければならない。そのための検証用ソフト、つまりそれらの値を変えたときの状態をシミュレートできるアプリケーションの開発も進めている。

7. まとめ

上記した問題点もあるが、この学習エンジンにより、検査エンジンは効率よくヒューリスティック検査法を用いることが出来る。つまり、高い精度でウイルス検査を行うことができるが速度の低下はさほどない。

参考文献

[1] 浦澤 俊彦, 千石 靖, 服部 進実: モバイルワクチンシステムの構築, 平成10年度電気関係学会北陸支部連合大会論文集, p.276 (1998)

[2] 浦澤 俊彦, 千石 靖, 服部 進実: 学習型モバイルエージェントとゲートウェイによる不正プログラム対策に関する研究, 2000 年暗号と情報セキュリティシンポジウム講演論文集, SCIS2000-D18 (2000)

[3] 伊吹 賢一, 千石 靖, 服部 進実: ヒューリスティック検査法を用いたワクチンソフトウェアの構築と検証, 平成11年度電気関係学会北陸支部連合大会論文集, p.287 (1999)

[4] 鈴木 暢人, 浦澤 俊彦, 千石 靖, 服部 進実: モバイルワクチンシステムにおける不正プログラム検出ゲートウェイの構築と検証, 平成11年度電気関係学会北陸支部連合大会論文集, p.270 (1999)