様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成２４年６月５日現在

機関番号：１１２０１
研究種目：若手研究（Ｂ）
研究期間：2010～2011
課題番号：２２７０００６２
研究課題名（和文）　Wine のログを利用した未知ウイルス検出手法の研究

研究課題名（英文）　Unknown Virus Detection Technique Using the Wine Log

研究代表者
中谷　直司（NAKAYA NAOSHI）
岩手大学・工学部・助教
研究者番号：20322969

研究成果の概要（和文）：この研究では、Unix 系 OS 上で Windows プログラムを実行するツールである Wine を使い実行ファイルの API 関数の呼び出しログを取ることで､未知のウイルスも検出可能な、動的ヒューリスティック手法に基づくいくつかの検出手法を提案した。ウイルスと無害な実行ファイルを用意し提案手法をテストしたところ、実験結果から Wine を用いて API 関数の呼び出しログを取得することは、動的ヒューリスティック手法として有効であることが示された。

研究成果の概要（英文）：In this research, I proposed some detection methods employing dynamic heuristics and capable of detecting unknown viruses by using Wine, a tool for executing Windows programs on a Unix-like OS, to capture a log of API function calls made by executable files. When these proposed methods were run on a test suite containing both viruses and harmless non-virus files, these experimental results showed that using Wine to capture a runtime log of API function calls is an effective technique for dynamic heuristic methods.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
２０１０年度	900,000	270,000	1,170,000
２０１１年度	600,000	180,000	780,000
年度			
年度			
年度			
総　計	1,500,000	450,000	1,950,000

研究分野：総合領域
科研費の分科・細目：情報学、計算機システム・ネットワーク
キーワード：ネットワークセキュリティ技術、コンピュータウイルス

１．研究開始当初の背景

（１）インターネットをはじめとするネットワークの急速な発展にともない、コンピュータウイルス（以降、ウイルス）による被害は深刻なものとなってきている。特に近年では、一般家庭においてもネットワークに常時接続されていることが当たり前になっており、ネットワークを媒介として広まるウイルスの感染速度は極めて速く、わずかな時間に世界規模の被害を発生させている。しかし現在のウイルス対策は、既知のウイルス特徴点（以降、シグネチャ）をもとにしたパターンマッチによりウイルスの検出を行っているため、シグネチャの無い未知ウイルスは基本的に検出できず新種のウイルスによる被害がたびたび生じている。これに対抗するには、

未知ウイルスに対応したシグネチャによらないウイルス検出方式を用いるか、シグネチャ作成に要する時間を今以上に短縮する必要がある。

（２）これまでにシグネチャによらない未知ウイルス検出手法もいくつか提案されており、既存のウイルス対策ソフトにもオプション的な扱いだが実装されているものも存在する。これらの手法はウイルスの挙動を静的もしくは動的に解析し、その挙動からルールベースの発見的手法を用いるのが主流となっている。しかし、このようなルールベースの手法では、ウイルスを作成する人間もウイルス対策ソフトを解析するか、あるいは作成したウイルスが実際にウイルス対策ソフトに検出されるかを試すことによって、ルールを回避し検出を逃れる手段を講じることができる。そこで、未知ウイルスをルールではなく確率的に検出する手法として、これまでに「過去のウイルスの機能をベイズ学習アルゴリズムにより学習し、その結果をもとにウイルスを検出する」という手法の研究を行い成果を挙げることができた。また、「ウイルスの機能を多次元ベクトルで表現し、そのベクトル間の距離をもとにウイルスを検出する」という手法を提案し有効であることを確認した。

（３）しかしこれまでの研究から、現在のウイルスの多くは実行可能圧縮とでも言うべき圧縮状態で存在していることが多く、上記手法で“ウイルスの機能”を抽出するには圧縮されたウイルスを解凍するという作業が必要なことがわかった。ウイルスを機能抽出が可能な状態に自動解凍する手法の研究も行っているが、現状では 70%程度のウイルスしか自動では解凍できないでいる。そこで、本研究ではウイルスを解凍することなく機能抽出する手法を研究する。具体的には Linux 上で動作する Windows 互換レイヤーである Wine を用いてウイルスを実際に実行し、その挙動をログに取ることで上記手法で用いる“ウイルスの機能”を抽出する一種の動的解析手法を試みる。

２．研究の目的

（１）ウイルスが分刻みで出現する現状では、シグネチャを利用してウイルスを検出する既存のウイルス対策ソフトでは対応することのできない、未知のウイルスに遭遇する確率が高まっている。そこで、Linux 上で動作する Windows 互換レイヤーである Wine を用いて Windows 上で動作するウイルスの挙動ログを抽出し、そのログと既知のウイルスの挙動ログとの類似性に基づく手法、あるいはルールベースの発見的手法を用いて未知ウイルスを検出する手法に関する研究を行う。

（２）本研究で用いる“Linux 上で動作する Windows 互換レイヤーである Wine を用いてウイルスを実際に実行し、その挙動をログに取る”という手法は、ウイルスの動的解析手法の一種に分類されるものである。これまでの動的解析ではウイルスの実行環境に Windows の動作環境を丸ごと仮想化したものを用いるものや、Windows の動作を疑似的に実現するある種のエミュレート環境を用いるものなどがあった。しかし、これらの環境下での実行には、実行されるウイルス自身が仮想環境やエミュレート環境で実行されていることを認識する機能を持ち、それらの環境下での解析を回避するために動作を中断するという問題が発生している。その点において Wine による実行を認識する手段は未だ開発されていないため、本研究ではこれまでの動的解析手法では解析不可能なウイルスの解析を行うことが可能になる。

３．研究の方法

（１）検出手法に関する研究を始めるに辺り、予備調査として Wine のログを解析した。Wine は環境変数 WINEDEBUG に適当な値を設定した上で実行すると、非常に詳細なデバッグ用の動作ログを出力することができる。本研究ではこのログからウイルス検出に有効な“ウイルスの機能”、すなわちウイルスの挙動を示す API の呼び出しログを得る必要があり、検出手法の検討を始める前に検出に必要な情報がログから得られることを確認した。

（２）そこで、Wine のログから API の呼出をシーケンスとして抽出し、そのシーケンスを多次元ベクトルとしてとらえ、そのベクトル間の距離を利用して未知ウイルスを検出する手法に関して研究を行った。すなわち、既知のウイルスの API シーケンスが作るベクトルと、未知のウイルスが作るベクトル間の距離を算出することでウイルスを検出する手法である。距離の算出には次式を用いた。なお、この時 X と Y はそれぞれベクトルとする。

$$\frac{S_{Cosine}(X,Y) + S_{Jaccard}(X,Y)}{2}$$

$$S_{Cosine}(X,Y) = \frac{X^T Y}{\sqrt{X^T X Y^T Y}}$$

$$S_{Jaccard}(X,Y) = \frac{X^T Y}{X^T X + Y^T Y - X^T Y}$$

より具体的には API シーケンスを適当なハッシュ関数で数列に対応させ、既知ウイルスの数列と検出対象の実行ファイルの数列を２つの多次元ベクトルとして捉え、両者の間の距離を測定し十分に近ければウイルスとして検出する。ただし、２つの実行ファイルの API シーケンスの長さは基本的に一致しない

ので、ベクトル間距離を算出する前にグローバルアライメント処理を行う必要があった。図１にグローバルアライメントの概要を示す。

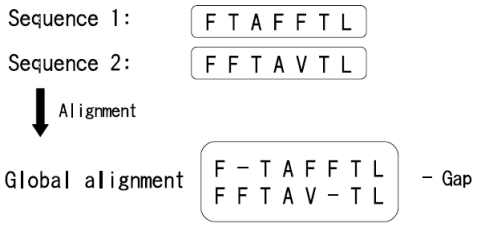

図１　グローバルアライメント

（３）さらに、コルモゴロフ複雑性に基づく正規化圧縮距離（NCD）を Wine のログに対して適用した。２つのデータ x と y があった時に、C(x)を x を圧縮アルゴリズム C で圧縮した時のビット長とすると、NCD は次式で表される。

$$NCD(x,y) = \frac{C(xy) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}}$$

NCD により 2 つのログが類似していれば距離が短く算出されるため、既知ウイルスと未知ウイルス間の距離を測ることでウイルスを検出できる。ただし、単純にログを圧縮しただけでは、ログ中に存在する実行するたびに変動する要素（実行時に決定されるアドレスなど）やログのファイルサイズの違いに影響を受ける問題があった。そこで、NCD を算出する前に Wine のログから実行するたびに変動する要素を予め取り除き、さらにログのサイズを同サイズに揃えることで類似性に対する精度を向上させる手法を提案した。
（４）また、ウイルスの挙動は Windows という OS をターゲットとしている以上、既知のものも未知のものも本質的には違いはない。そこで、Wine のログから API の呼出をシーケンスとして抽出し、そのシーケンスに対して既知ウイルスの挙動から作成した検出ルールを適用するルールベースの発見的手法に関する研究を行った。

４．研究成果
（１）研究の方法（１）で述べたように Wine のログを調査したところ、ログには実行したアプリケーションの挙動、すなわち API の呼出状況とその返り値を逐一記録することが可能であることがわかった。Wine のログの極一部を一例として図２に示すが、この例では GetDesktopWindow()、SetWindowLongW()、LoadIconW()の順に API が呼ばれていることがわかる。なお、図２で各行の先頭にある「0017」は Wine の内部で各スレッドに割り当てられるスレッド ID であるが、Wine で実行したプログラムに割り当てられる初期スレッド ID は、必ず 0009 になることも確認された。また、プロセスやスレッドの分岐状況等も環境変数に適当な値を設定することでログに出力することができるため、これによりウイルスの挙動をログから追跡可能なことが確認された。

```
0017:Call user32.GetDesktopWindow() ret=7ee9cfd9
0017:Ret  user32.GetDesktopWindow()
    retval=00010020 ret=7ee9cfd9
0017:Call user32.SetWindowLongW(00010020, fffffffc,
    7ee9d810) ret=7ee9d318
0017:Ret  user32.SetWindowLongW() retval=7ecb5360
    ret=7ee9d318
0017:Call user32.LoadIconW(00000000, 00007f05)
    ret=7ee9d32f
0017:Ret  user32.LoadIconW() retval=000010f6
    ret=7ee9d32f
```

図２　Wine のログの例

（２）研究の方法（２）で述べた API シーケンスを多次元ベクトルとしてとらえ、そのベクトル間の距離を利用して未知ウイルスを検出する手法に関して検出実験を行った。なお、ここで言う API シーケンスとは図２の例で言えば、GetDesktopWindow()、SetWindowLongW()、LoadIconW()といった具合に呼び出された API の関数名だけを並べたものであり、それ以外の引数や返り値は含まないものとする。現在、発見される未知ウイルスのほとんどは、既存の既知ウイルスを改変した亜種ウイルスであり、また亜種ではない未知ウイルスであっても既知ウイルスと何らかの共通点を持つものが大半である。したがって、提案手法で既知ウイルスの API シーケンスと検出対象ファイルの API シーケンスを比較すれば、未知ウイルスであっても検出することが可能と考えられる。実験の結果、提案手法はウイルスと通常の実行ファイル（以降、ノンウイルス）を明確に識別でき、かつウイルスを亜種ごとに分類することも可能なことが確認された。
（３）研究の方法（３）で述べたように、Wine のログを圧縮し NCD を算出することで未知ウイルスの検出を試みた。すなわち、ここでも既知ウイルスのログと検出対象のログ間の距離が近ければ、検出対象をウイルスとする手法である。まず、Wine のログをそのまま使って実験を行ったところ、NCD によるウイルスの検出は有効であることが確認され、ウイルス亜種の分類にもある程度の成果が見られた。しかし、NCD が類似度を算出するものである割には、ウイルス亜種の分類にあまり効果がないという欠点もあった。そこで、NCD を算出する前に Wine のログから実行するた

びに変動する要素を予め取り除き、さらにログのサイズを同サイズに揃えることで、ウイルス亜種の分類にも有効に機能することが確認された。これは例えば図２において、１行目の「ret=7ee9cfd9」のイコール以降の 16 進数の部分など、実行環境に依存し亜種ウイルスであろうと一致しそうにない、NCD の算出においてノイズとなっている部分が比較対象から外れることで精度が上がった結果と考えられる。また、同じ亜種ウイルスのログでも極端にサイズの違うものが存在し、これも NCD 算出時に問題となっていた。実際にログファイルの中身を確認すると、最初の内は似通った API を呼んで同じように進んでいるが、途中から片方がループに入り同じ API 呼び出しを繰り返していることが確認された。NCD は理論上はループを無視し、ループがない場合と同じ類似度を算出できるはずだが、あくまでそれは理論上であって現実にはループの影響を無視できない。そこでサイズを制限し、ループに入る前のログだけを利用することで対応した。

（４）研究の方法（４）で述べたルールベースの検出手法における検出ルールには以下の７つを用いた。まず、ウイルスの機能としては古典的な「ファイルの作成、削除」、「レジストリの改ざん」、「外部との通信」を検出するルールである。これらはウイルスがコンピュータに感染し、さらに他のコンピュータに拡散するための基本的な機能を検出する。また、アンチウイルスソフトによる検出を回避するための機能に対応する、以下の３つの検出ルールを採用した。まずは「アンチウイルスソフト等の停止」、これは文字通りの意味でウイルスによるアンチウイルスソフトの強制終了を検出する。次に「デバッガの検知」、デバッガを用いたウイルス解析に対抗する機能を持つウイルスが存在するが、このルールは対抗のきっかけとなるデバッガの検知機能を検出する。そして最後に「自身の名称確認」、これはウイルスが自分自身のファイル名を確認し、変更されていた場合は動作しないという機能を検出する。この機能はおそらく、ウイルスを解析のためにコピーする際にファイル名が変更される可能性が高い事に着目して、解析を回避するためのものと思われる。そして７つ目のルールとして、「金融機関のドメイン参照」を採用した。このルールはウイルスが、ネットバンキングの ID とパスワードを盗む機能を検出する。以上の７つのルールに対応した API シーケンスが、検出対象のファイルを実行した時の Wine ログにあった場合、そのファイルをウイルスとして検出することにした。実験の結果、ウイルス検出率は 99.2%と極めて高く、またノンウイルスをウイルスとしてしまう誤検出率も 2.7%と十分に低い値が得られた。

（５）以上のことから、Wine のログを未知ウイルス検出に利用することは有用であると言える。また、本研究により未知ウイルスの検出に新たな選択肢が増えることは、ウイルスによる被害を未然に防ぐ可能性の増加を意味し、ネットワーク全体のセキュリティの向上に貢献するものと思われる。しかし、Wine による Windows プログラムの実行は未だ完全ではないため、実際には実行できないプログラムも多数存在する。比較的動作が単純で環境に依存しないように作られているウイルスは実行できる場合も多いが、一般の実行ファイルには動作しないものも多く、今後はこの点についても考えていかねばならない。

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔学会発表〕（計２件）

① 村上智裕、中谷直司、厚井裕司、Wine を用いた API ログによるコンピュータウイルスの検出、平成 23 年度第 4 回情報処理学会東北支部研究会、2012.01.21、岩手大学

② 辺仙龍、中谷直司、厚井裕司、圧縮による類似度比較を適用したウイルスの検出手法、平成 22 年度第 3 回情報処理学会東北支部研究会、2010.12.18、岩手大学

６．研究組織

(1)研究代表者

中谷　直司（NAKAYA NAOSHI）

岩手大学・工学部・助教

研究者番号：20322969