# 科学研究費助成事業　　研究成果報告書

平成 ２６ 年　 ６ 月　 ６ 日現在

機関番号：３４３１５
研究種目：基盤研究(C)
研究期間：2011～2013
課題番号：２３５００１０１

研究課題名（和文）仮想計算機モニタを用いたシステムコールレベルのマルウェア動的解析とその自動化

研究課題名（英文）Automatic Dynamic Analyzer for Malware by System Call Tracing using Virtualization Technology

研究代表者
毛利　公一（Mouri, Koichi）

立命館大学・情報理工学部・准教授

研究者番号：９０３１３２９６

交付決定額（研究期間全体）：（直接経費）　　3,500,000円、（間接経費）　　1,050,000円

研究成果の概要（和文）：　社会がコンピュータとインターネットに依存している現代，新種のマルウェア（コンピュータウィルスなど悪意あるソフトウェアの総称）が次々に発見されている中で，その安全性を確保することは重要な課題である．そのためには，マルウェアがどのような脅威を与えるものかを迅速に調査・対策をする必要がある．本課題では，システムソフトウェアの技術を用い，迅速にマルウェアを解析する技術を確立することが目的である．この成果として，マルウェアがどのような挙動（システムコールを発行）をしているのか，確実に記録するシステムを開発した．また，その記録からどのような意味があるのかを抽出して表示するための解析ツール群も開発した．

研究成果の概要（英文）：Nowadays our society depends on computers and networks deeply. It is very difficult to keep them safe completely. Its one big reason is that new kinds of malware appear everyday. To achieve their safety, we need to quickly know what threat malwares cause. Then we need to propose a countermeasure against them. In this research subject, the goal is to establish a method to quick analysis of malwares, using virtualization technology. As the result, we have developed system call tracer named Alkanet. It can do it more quickly than other methods. It also has a resistance facility against anti-debug function of malwares. Furthermore we developed various analyzing tools for Alkanet, which enable to know easily what the malwares cause.

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：マルウェア解析　コンピュータセキュリティ　ネットワークセキュリティ　仮想化技術　オペレーティングシステム　エンドポイントセキュリティ　システムコールトレース

## １．研究開始当初の背景

以前のマルウェアは，一部の者の悪ふざけや技術力誇示を目的としたものが多く，そのためマルウェアの挙動は派手で目につくものが多かった．しかし，最近では，金銭目的のものが中心に変わってきており，巧妙で目立たずに目的を達成するような静かな攻撃にシフトしている．最近ではマルウェアの被害に関する報道が減少していることからも，静かな攻撃への変化がつかみとれる．しかし，実際には，2009 年に 2 億 4000 万種ものマルウェアの新種が見つかっており，前年比 70%増とその勢いを増している．

理想的にはマルウェアを根絶できることが望ましいが，実際には，オペレーティングシステム(OS)やアプリケーションのバグをゼロにすることは困難であるため，マルウェアを根絶することも難しい．また，マルウェアは，ファイアウォールや侵入検知システムといったネットワークレベルの対策をすり抜けてくるため，広い範囲を一気に保護する手法がない．したがって，一般的なマルウェア対策は，ウィルス対策ソフトを個々の PC へ導入することとなっている．このウィルス対策ソフトは，大きく 2 種類あり，一つは PC に常駐し，マルウェアがファイルとして実体化する瞬間を検知し，実行される前に駆除するものがある．もう一つは，既にマルウェアが実行され感染状態にある否かをチェックし，感染状態にあれば感染前に復帰させるものがある．いずれの場合でも，マルウェアが発見されてからの迅速な挙動の把握と早期対策が重要である．

マルウェアの解析に使われている技法の一つに，マルウェアを逆アセンブルして読みほどく静的解析がある．この方法は，マルウェアの全貌を詳細に知ることができるが，時間がかかるという短所がある．最近は，マルウェアに暗号化や難読化が施されることが多いため，静的解析がさらに困難になっている．一方で，実際にマルウェアを動作させて解析する動的解析という手法がある．例えば，デバッガや命令トレーサを使って機械語の動作を 1 命令ずつ追う方法がある．この方法は，難読化や暗号化の影響は受けないが，機械語命令単位での追跡となるため，動作の意図を理解するのに時間がかかる．また，アプリケーション向けライブラリ(DLL)にフックを挿入してマルウェアがどのライブラリ関数を呼び出しているかを追跡する方法もある．この方法は，動作の意味を把握しやすいという長所がある．しかし，マルウェアがそれらを回避することは容易であるという短所がある．さらに，デバッガ検出機能を有するマルウェアもあり，デバッガを検出すると，マルウェアが挙動を変えてしまうことがある．他にも，自身の処理時間を計測し，極端に遅い場合は解析されていると判断して挙動を変えてしまうマルウェアも存在する．

以上から，次に挙げる点が重要であることがわかる．

(1) マルウェアから解析機構が検出されないこと
(2) マルウェアに解析機構を回避されないこと
(3) 短時間でマルウェアの動作の概要を理解できること
(4) ソースコードがないプロプライエタリな OS 上で動作するマルウェアでも解析できること

本応募研究課題では，これらの目標を達成するマルウェアの動的解析システムを提案する．

## ２．研究の目的

上述の(1)～(4)を達成するための具体的な技術課題 2 点について，以下に示す．

(1)と(4)を実現するために，仮想計算機モニタ(VMM)を用いた解析手法を採る．すなわち，VMM 内から，その上で動作する OS(ゲスト OS)とマルウェアを観測し，そこで得られた情報をもって解析を行う．これによって，マルウェアから解析機構が検出不可能となり(1)を実現できる．また，プロプライエタリな OS であっても，カーネルを変更する必要がないため，(4)についても解決できる．以上から，本課題では，マルウェア解析に適した VMM 構成法について明らかにする．

(2)と(3)を実現するために，解析の粒度をシステムコール単位とする．どのようなマルウェアであっても，意味のある活動をするにはシステムコールを使用しなければならず，それを回避することはできない．すなわち(2)を実現できる．また，システムコールは，ファイル入出力，ネットワーク通信，プロセスの生成・終了といった粒度のインタフェースであり，機械語命令よりも抽象度が高く，その意図を把握しやすい．さらに，例えばファイル入出力のシステムコールは，オープン・読み書き・クローズといった流れがある．これらを関連づけることができれば，マルウェアの意図を一層把握しやすくなる．また，自動起動を設定するためのレジストリアクセスなど，典型的なマルウェアの挙動については，自動解析ができればより効率的な解析が実現できる．すなわち(3)を実現できる．以上から，本課題では，VMM 内部からシステムコール単位で解析する手法，およびシステムコールの流れを自動追跡し関連づけするなど自動解析手法も明らかにする．

## ３．研究の方法

研究目的を達成するために，平成 23 年度は，マルウェア解析に特化した超軽量・超小型 VMM の構築と，システムコール単位のマルウェア解析手法の確立を個別に進める．これによって，マルウェア解析機構の基本機能を

実現する．平成 24 年度は，前年度の研究成果を合わせてマルウェア解析機構を開発する．さらに，解析結果を他の PC へ転送するための，高速で安全なロギング機構や，マルウェア解析実験環境としてのハニーポットを開発する．平成 25 年度は，システムコールの自動解析を実現することで，解析システム全体の完成を目指す．その後，解析システムのオーバヘッドなど基本性能を評価する．さらに，実際のマルウェア検体を用い，解析システムを用いた場合と，IDA Pro など他方式を用いた場合とについて，解析に要する時間，解析が不可能であったマルウェアの数，解析で得られた情報量などを比較し，本研究の評価とする予定である．

## ４．研究成果

本応募研究課題の成果として，Alkanet と名付けたマルウェア解析システムを開発した．以下では，Alkanet についてその特徴や概要について述べる．

動的解析をする上で課題となるのが，マルウェアを解析する粒度である．粒度としては，呼び出されたライブラリ（API）単位，システムコール単位，機械語命令単位の手法がある．API 単位の場合，マルウェアの挙動にどのような意図があるかを理解しやすいが，API を回避される場合もあり，完全な追跡ができない場合がある．機械語命令の場合，完全な追跡が可能であるが，オーバヘッドが極端に大きい．また，機械語から挙動の意図をつかむのが難しい．システムコールは，挙動の意図がつかみやすく，かつ回避不可能であり，オーバヘッドの抑制が可能である．ただし,対象となる OS の詳細を知る必要がある．

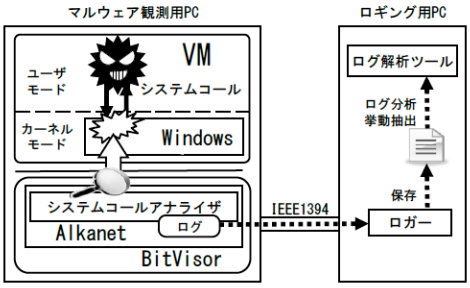
また，比較的新しいマルウェアでは，マルウェア自身が動的解析の対象となっていることを検知し，実行の停止や解析の妨害などを行うアンチデバッグ機能である．仮想計算機モニタ（VMM）やエミュレータを用いた既存研究では，アンチデバッグ機能に対処が難しい．

Alkanet はこれらを解決する，次のような特徴を有する動的解析システムである．

(1) マルウェアによって回避が不可能である．
(2) マルウェアの挙動の意図をつかみやすい粒度での解析を実現する．
(3) オーバヘッドを軽減できる．
(4) 解析に耐性を持つマルウェアでも解析できる．

また，評価を行い，Alkanet が従来の動的解析システムと比較して十分に高速であることを確認した．さらに，典型的なアンチデバッグ手法に対する耐性評価を行い，Alkanet が十分高い耐性を持つことを確認した．

Alkanet は，既存の VMM である BitVisor をベースとしており，システム全体の概要は次




の図のようになっている．解析時には，2 台の計算機を用いる．観測用 PC（左側）ではマルウェアを動作させ，システムコールをトレースする．ロギング用 PC（右側）では，トレース結果のログを各種の解析ツールを用いて解析するものである．両者は IEEE1394 ケーブルで接続され，Direct Memory Access を用いて高速にログ転送が可能としている．

システムコールトレースの動作概要は次の通りである．

(1) マルウェアは自身の活動のためにシステムコールを発行する．
(2) カーネル内のシステムコールハンドラに処理が移った瞬間に，さらに Alkanet に制御が移る．
(3) Alkanet により，システムコールを発行したプロセス・スレッドの情報，システムコールの種類や引数などを調査・記録する．
(4) 制御をカーネルに戻し，システムコールを実行する．
(5) 再度 Alkanet に制御を移し，システムコールの成否や戻り値を記録する．
(6) 制御をカーネルに戻し，さらにマルウェアに処理を戻す．

検査対象とするシステムコールは，以下に示すような，マルウェアが典型的に利用するものに限っている．

- ファイルのオープン，作成，読出し，書込み
- レジストリの参照，設定
- 仮想メモリの読出し，書込み，確保，権限の変更
- ファイルマッピングの作成，オープン，ビューの作成
- ネットワークの送信，受信
- プロセスの作成，終了
- スレッドの作成，終了，停止，再開，コンテキスト変更
- ドライバのロード，アンロード
- 処理時間の計測
- スリープ

また，取得する具体的な情報は下記の通りである．

- システムコール発行元の Cid （プロセス ID とスレッド ID の組）と実行ファイル名
- システムコール番号
- システムコールの引数と戻り値

なお，システムコールの引数については，重

要な情報（各種のオブジェクト）へのポインタとなっていることが多い．Alkanet では，ポインタ値を取得するだけでなく，各種オブジェクトの内容も取得する．

性能評価としては，Alkanet 全体としてオーバヘッドの抑制が実現できているか否かを示すために，Alkanet による命令トレース機能を動作させた状態で PCMark05 ベンチマークを用いてスコアを計測した．その結果，ハードウェア上に直接 Windows をインストールして動作させた場合を 100 とすると，Alkanet を有効にした場合でも 81 程度となり，実用的な速度で動作することが確認された．

機能評価として，これまで 30 体を超えるマルウェア検体を実際に動作させた．その結果，セキュリティベンダーが報告するマルウェアの挙動と，Alkanet で解析した結果が一致することを確認した．

機能評価のもう一つの重要な点であるアンチデバッグ機能に対する有効性についても評価した（下の表を参照）．アンチデバッグ機能の実現に使われると報告されている 34 種類の手法のうち，33 種類について Alkanet がマルウェアに検出されないことが示された．なお，対処されていない 1 種類については，対応可能であることが確認されており，今後の課題としている．

| アンチデバッグの名称 | 回避可能か |
|---|---|
| PEB NtGlobalFlag | ○ |
| IsDebuggerPresent | ○ |
| CheckRemoteDebuggerPresent | ○ |
| Heap Flags | ○ |
| NtQueryInformationProcess - ProcessDebugPort | ○ |
| Debug Objects - ProcessDebugObjectHandle Class | ○ |
| Debug Objects - ProcessDebugFlags Class | ○ |
| NtQuerySystemInformation - SystemKernelDebuggerInformation | ○ |
| OpenProcess - SeDebugPrivilege | ○ |
| Alternative Desktop | ○ |
| Self-Debugging | ○ |
| RtlQueryProcessDebugInformation | ○ |
| Hardware Breakpoint | ○ |
| OutputDebugString | ○ |
| BlockInput | ○ |
| Parent Process | ○ |
| Device Names | ○ |
| OllyDbg - OutputDebugString | ○ |
| FindWindow | ○ |
| SuspendThread | ○ |
| SoftICE - Interrupt 1 | ○ |
| SS register | ○ |
| UnhandledExceptionFilter | ○ |
| Guard Pages | ○ |
| Execution Timing | △ |
| Software Breakpoint Detection | ○ |
| Thread Hiding | ○ |
| NtSetDebugFilterState | ○ |
| Instruction Counting | × |
| Self-Execution | ○ |
| DbgBreakpoint Overwrite | ○ |
| CPU Instructions Results Comparison | ○ |
| VMWare - IN Instruction | ○ |
| VirtualPC - Invalid Instruction | ○ |

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）

〔雑誌論文〕（計 5 件）


(1) 大月 勇人，瀧本 栄二，齋藤 彰一，毛利 公一，「Alkanet におけるシステムコールの呼出し元動的リンクライブラリの特定手法」，コンピュータセキュリティシンポジウム 2013(CSS2013)論文集，査読:無，Vol. 2013，No. 4，pp. 753-760，2013.
http://id.nii.ac.jp/1001/00098272/

(2) Yuto Otsuki，Eiji Takimoto，Takehiro Kashiyama，Shoichi Saito，Eric W. Cooper，and Koichi Mouri，“Tracing Malicious Injected Threads Using Alkanet Malware Analyzer，” IAENG Transactions on Engineering Technologies，査読：有，Lecture Notes in Electrical Engineering 247，pp. 283-299，2013.
DOI:10.1007/978-94-007-6818-5_21

(3) 大月 勇人，若林 大晃，瀧本 栄二，齋藤 彰一，毛利 公一，「マルウェアアナライザ Alkanet によるマルウェア解析報告 2012」，コンピュータセキュリティシンポジウム 2012(CSS2012)論文集，査読：無，Vol. 2012，No. 3，pp. 106-113，2012.
http://id.nii.ac.jp/1001/00086635/

(4) Yuto Otsuki，Eiji Takimoto，Takehiro Kashiyama，Shoichi Saito，Eric W. Cooper，and Koichi Mouri，“Alkanet: A Dynamic Malware Analyzer based on Virtual Machine Monitor，” World Congress on Engineering and Computer Science 2012 (WCECS 2012)，査読：有，Vol. 1，pp. 36-44，2012.
http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp36-44.pdf

(5) 大月 勇人，瀧本 栄二，樫山 武浩，毛利 公一，「マルウェア挙動解析のためのシステムコール実行結果取得法」，コンピュータセキュリティシンポジウム 2011(CSS2011)論文集，査読：無，Vol. 2011，No. 3，pp. 95-100，2011.
http://id.nii.ac.jp/1001/00077912/


〔学会発表〕（計 6 件）


(1) 大月 勇人，瀧本 栄二，齋藤 彰一，毛利 公一，「Alkanet におけるシステムコールの呼出し元動的リンクライブラリの特定手法」，コンピュータセキュリティシンポジウム 2013(CSS2013)，2013 年 10 月 23 日，かがわ国際会議場(香川県)．

(2) 大月 勇人, 瀧本 栄二, 毛利 公一, 「BitVisor へのシステムコールトレース機能追加によるマルウェアアナライザの実現」, BitVisor Summit, 2012 年 12 月 4 日, 筑波大学東京キャンパス文京校舎(東京都).
(3) 大月 勇人, 若林 大晃, 瀧本 栄二, 齋藤 彰一, 毛利 公一,「マルウェアアナライザ Alkanet によるマルウェア解析報告 2012」, コンピュータセキュリティシンポジウム 2012(CSS2012), 2012 年 10 月 30 日, くにびきメッセ(島根県).
(4) Yuto Otsuki, Eiji Takimoto, Takehiro Kashiyama, Shoichi Saito, Eric W. Cooper, and Koichi Mouri, "Alkanet: A Dynamic Malware Analyzer based on Virtual Machine Monitor," World Congress on Engineering and Computer Science 2012 (WCECS 2012), 2012 年 10 月 25 日, Berkeley, California (USA).
(5) 大月 勇人, 瀧本 栄二, 樫山 武浩, 毛利 公一,「仮想計算機モニタを用いたマルウェアの挙動解析」, コンピュータシステム・シンポジウム ポスター・デモセッション, 2011 年 11 月 30 日, 立命館大学朱雀キャンパス(京都府).
(6) 大月 勇人, 瀧本 栄二, 樫山 武浩, 毛利 公一,「マルウェア挙動解析のためのシステムコール実行結果取得法」, コンピュータセキュリティシンポジウム 2011(CSS2011), 2011 年 10 月 19 日, 朱鷺メッセ(新潟県).

〔図書〕(計 0 件)

〔産業財産権〕

○出願状況 (計 0 件)

○取得状況 (計 0 件)

〔その他〕
ホームページ等
http://www.asl.cs.ritsumei.ac.jp/

６．研究組織
(1)研究代表者
毛利 公一 (MOURI KOICHI)
立命館大学・情報理工学部・准教授
研究者番号：90313296