仮想計算機モニタによるマルウェアの監視

大月 勇人† 毛利 公一†

†立命館大学情報理工学部情報システム学科
525-8577 滋賀県草津市野路東 1-1-1
{yotuki,mouri}@asl.cs.ritsumei.ac.jp

あらまし 近年のマルウェアには難読化やアンチデバッグといった手法が用いられており, 逆アセンブラやデバッガによる解析が困難である. そこで, 我々は, OS より下位で動作する VMM に着目し, マルウェア解析のための VMM である Alkanet の開発を行っている. Alkanet は, システムコールをフックすることで, マルウェアの挙動を監視する. さらに, Windows の管理するメモリ空間にアクセスすることで, Windows 内部のオブジェクトを参照し, 詳細な情報の取得を可能にしている. 本論文では, Alkanet の構成と Alkanet を用いたマルウェア解析について述べる.

Tracing Malware by Virtual Machine Monitor

Yuto Otsuki† Koichi Mouri†

†Department of Computer Science, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga 525-8577 JAPAN
{yotuki,mouri}@asl.cs.ritsumei.ac.jp

Abstract Recent malwares are applied obufuscation and debugger detection not to be analized by disassemblers and debuggers. We are developing an "Alkanet" virtual machine monitor for malware analysis. Alkanet monitors behavior of malwares by a system call hook. In addition, Alkanet can obtain detailed information of Windows by referring to internal objects managed in memory region of Windows. In this paper, we describe structure of Alkanet and malware analyses by Alkanet.

1 はじめに

近年, コンピュータとネットワークの普及につれて, マルウェアの脅威が問題となっている. マルウェアの技術は日々着実に進歩しており, 次々と新しいマルウェアが出現している. 従来, マルウェアの解析は, 逆アセンブラやデバッガによる手法が一般的であった. しかし, 難読化やアンチデバッグが施されたマルウェアが増加し, 従来の方法による解析が困難になっている. このようなマルウェアは, OS のカーネル内部から直接監視・解析することが望ましい. しかし, Windows はプロプライエタリソフトウェアであるため, そのような手段を取ることが難しい. そこで, 我々は, OS である Windows よりも下位のレイヤで動作する仮想マシンモニタ (VMM) に注目し, Windows 上で動作するマルウェアを解析する VMM である "Alkanet" の開発を行っている [1].

Alkanet は, プロセスが発行するシステムコールをフックする機能を持つ. さらに, Windows の管理するオブジェクトを利用することで, フックしたシステムコールの詳細を得ることを可能にしている.

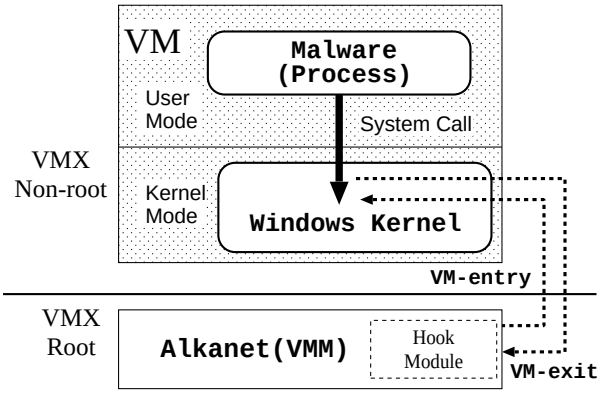

図 1: Alkanet の全体構成

2 Alkanet

2.1 全体構成

Alkanet は, 仮想マシン上の Windows で動作するマルウェアを監視・解析する VMM であり, ハイパーバイザ型の VMM である BitVisor[2] をベースとして実装している. Alkanet の全体構成を図 1 に示す. Alkanet では, 仮想マシン上の Windows で動作するプロセスが発行するシステムコールをフックすることで, マルウェアの挙動の監視・解析を行う.

システムコールのフックは, Windows システムコールのエントリポイントにソフトウェアブレイクポイントを埋め込むことで実現している. しかし, システムコールをフックしただけでは, 発行元のプロセスや, どのリソースに対する操作かといった詳細な情報が得られない. そこで, Alkanet では, Windows の使用するメモリ空間を参照し, Windows が管理するオブジェクトにアクセスすることで, 詳細な情報の取得を可能としている.

一般的にマルウェアには, 以下のような挙動が見られる.

- 重要なファイルやレジストリの改竄
- バックドアの設置や別のマルウェアのダウンロード
- 既存のプロセスと似た名前でのプロセスの起動や別のプロセスへのコードの挿入

マルウェアが行う上記の挙動は, 主に表 1 に示すシステムコールを発行することによって行われる. したがって, Alkanet は, これらのシステムコールをフックの対象としている.

2.2 システムコールのフック

Alkanet は, Intel 製 CPU 上で動作する 32bit 版 Windows XP SP3 を対象としている. この環境におけるシステムコールは, sysenter 命令によってカーネルモードへの切り替えが行われる. sysenter 命令が実行されると, EIP レジスタは, IA32_SYSENTER_EIP という Model Specific Register の値に上書きされる. つまり, IA32_SYSENTER_EIP に格納された値が, システムコールのエントリポイントである. Alkanet におけるシステムコールのフックの流れを図 2 に示す. Alkanet は, 初期化処理として IA32_SYSENTER_EIP からシステムコールエントリポイントを取得し, int3 命令を埋め込む. これにより, ゲスト OS 内でシステムコールが発行されるとブレイクポイント例外が発生し, VMM 側に制御を移すことが可能になる. また, VMM からゲスト OS に制御を戻す際には, int3 命令で上書きされた命令をエミュレーションする.

Windows では, システムコール発行時, EAX レジスタにシステムコールの番号, EDX レジスタに引数のリストのアドレスがそれぞれ格納される. よって, Alkanet は, フック時にこれらのレジスタの値を用いて, 発行されたシステムコールと引数のアドレスを特定している.

2.3 発行元プロセスの特定

Alkanet は, フック時の CR3 レジスタの値を用いて, フックしたシステムコールの発行元プロセスを特定する. CR3 レジスタには, ページディレクトリテーブルの物理アドレスが格納されている. Alkanet は, この値がプロセスごとに異なっていることを利用して, システムコールを発行したプロセスを特定する. しかし, 物理アドレスの値のみでは, マルウェアの挙動を

表 1: Alkanet が監視する挙動

挙動	フックするシステムコールの例
ファイルの操作	NtCreateFile, NtReadFile, NtWriteFile, ...
レジストリの操作	NtCreateKey, NtQueryValueKey, NtSetValueKey, ...
ネットワークの操作	NtDeviceIoControlFile, NtReadFile, NtWriteFile
プロセスの作成/終了	NtCreateProcess, NtCreateProcessEx, NtTerminateProcess
ドライバのロード	NtLoadDriver, NtUnloadDriver
別プロセスへのコード挿入	NtCreateThread, NtWriteVirtualMemory

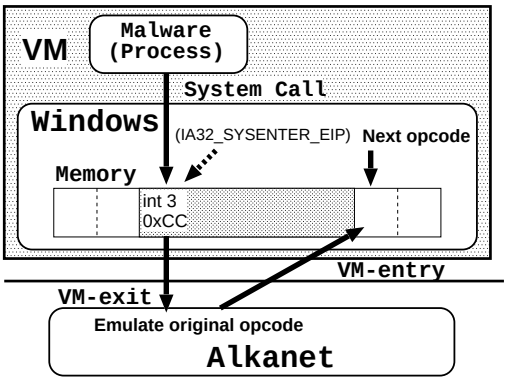

図 2: システムコールフックの流れ

追うことが困難である. そこで, Windows の管理するメモリ空間を読み出し, プロセスの情報を取得することで, これを解決する. 具体的には, プロセスを管理するプロセスオブジェクトの双方向リストを用いる. このリストと CR3 レジスタの値とを用いて, システムコール発行元プロセスに対応するプロセスオブジェクトを特定する. これにより, 特定したプロセスオブジェクトを用いた情報の取得が可能となる.

2.4 ハンドルの解決

Windows は, オブジェクトマネージャと呼ばれるコンポーネントにより, ファイルやレジストリなどのリソースをオブジェクトと呼ばれる形で統一的に管理している. プロセスがオブジェクトをオープンすると, プロセスの持つハンドルテーブルにそのオブジェクトを示すエントリが追加される. そして, そのエントリを参照するためのハンドルがプロセスへと返される. 以降, プロセスは, このハンドルを用いて, オブジェクトへのアクセスを行う.

各種オブジェクトの操作を行うシステムコールでは, 引数として操作するオブジェクトのハンドルが与えられる. Alkanet は, プロセスの持つハンドルテーブルを参照し, 与えられたハンドルの示すオブジェクトを特定する. このようにして, システムコールの引数を基に, 実際にオープンされるファイルや書き換えられるレジストリのパスなどを取得する.

2.5 SysCallViewer

Alkanet で取得した情報を閲覧するために, SysCallViewer というログビューアを作成した. SysCallViewer は, ゲスト OS の Windows 上で動作するアプリケーションであり, Alkanet から解析ログを取得し, 整形して画面上に表示する.

SysCallViewer の実行画面を図 3 に示す. 各項目は, 左から, ログの通し番号, プロセス ID, プロセス名, システムコール番号, システムコール名, 備考となっている. 備考には, 操作するファイルやレジストリなどのオブジェクトの名前や, 詳細な付加情報などが表示される.

3 Alkanet によるマルウェア解析

3.1 解析対象

Alkanet の性能評価として, 実際に Alkanet 上でマルウェアの挙動解析を行った. なお, 検証時はネットワークには接続していない.

検体として, CCC DATAset 2010[3] の中で活動が記録されているマルウェア検体の中から, ClamAV[4] に Worm.Palevo-2648 と Trojan.Downloader.Bredolab-1420 として検出されるものを用いた. マルウェアのイメージ名は, それぞれ Palevo.exe と Bredolab.exe である.

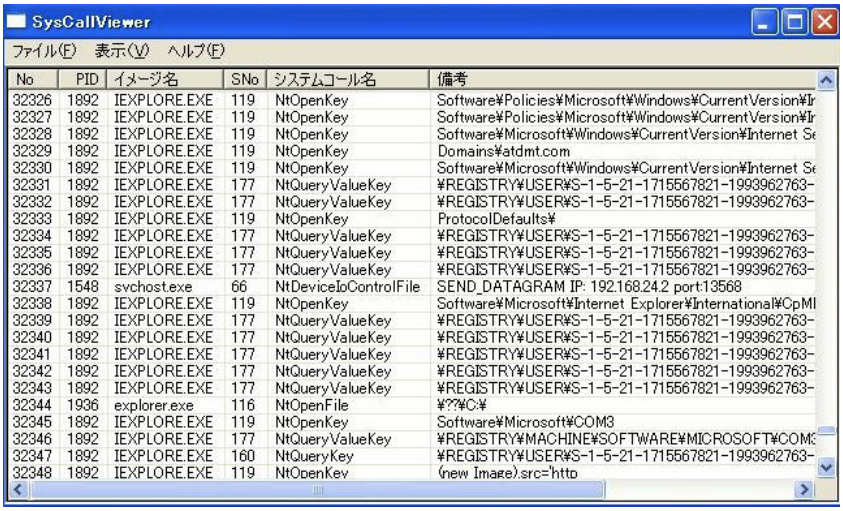

図 3: SysCallViewer

3.2 Palevo.exe

Alkanet から取得したログの一部を図 4 に示す. 1753〜1757 行目では, 存在しないユーザのゴミ箱フォルダに psyjo3.exe や Desktop.ini を作成している. 自身をオープンしていることから, 自身のコピーを作成していると思われる.

1758〜1766 行目では, 特定のレジストリのキーを参照し, 書き換えている. これらは, Windows のスタートアップ時に起動するアプリケーションを登録するキーと, ログオンプロセスによって利用されるキーである. これらのレジストリを確認すると, Windows 起動時に psyjo3.exe が起動するよう設定がされていた. また, タスクマネージャが psyjo3.exe に置き換えられていた.

1769〜1773 行目では, explorer.exe のメモリ空間を書き換え, 新たにスレッドを作成し, 自身を終了している. これは, explorer.exe に自身の持つコードを挿入し, 実行させるという挙動である. この後, explorer.exe は, 拡張子とアプリケーションの関連付けを行うレジストリを変更したり, 特定のポートを開くなどの不審な挙動を示した.

しかし, 現在の Alkanet は, 実行単位をプロセスで区別しているため, 正規の explorer.exe の挙動とコードを挿入されたことによる挙動とを厳密に区別することは難しい. この問題は, Windows における実行単位であるスレッドの情報や, 実行されているイメージの情報などを取得することで解決できる.

3.3 Bredolab.exe

Alkanet から取得したログの一部を図 5 に示す. 2204〜2210 行目では, 名前付きパイプにアクセスし, lsass.exe と通信を行っている. また, svchost.exe とも同様に通信を行っていた. しかし, これらのプロセスのその後の挙動が, 正規のものであるか否かの区別は困難である.

3157〜3170 行目では, ドライブのマウントに関するレジストリのキーの変更を行っている. このレジストリは, ドライブの自動再生機能を制御する Autorun.inf のキャッシュとして利用される [5]. {a66607c8-a9f7-11df-a9a4-806d6172696f} という GUID を持つドライブを調査すると, C ドライブとしてマウントされるハードディスクであった. これらの操作によって, このマルウェアがシステム起動時に自動的に実行されるようになる.

3359 行目, 3367〜3368 行目では, cmd.exe を起動し, 自身のファイルを削除させている. また, この他に以下のような挙動が見られた.

- DirectX やシステムの情報の収集
- お気に入りや最近使ったフォルダなど設定の書換え

```
                                        :
1753 496 Palevo.exe 37 NtCreateFile
        \??\C:\Documents and Settings\yotuki\My Documents\Palevo.exe
1754 496 Palevo.exe 37 NtCreateFile
        \??\C:\RECYCLER\S-1-5-21-0243556031-888888379-781863308-1455\psyjo3.exe
1755 496 Palevo.exe 274 NtWriteFile
        \RECYCLER\S-1-5-21-0243556031-888888379-781863308-1455\psyjo3.exe
1756 496 Palevo.exe 37 NtCreateFile
        \??\C:\RECYCLER\S-1-5-21-0243556031-888888379-781863308-1455\Desktop.ini
1757 496 Palevo.exe 274 NtWriteFile
        \RECYCLER\S-1-5-21-0243556031-888888379-781863308-1455\Desktop.ini

1758 496 Palevo.exe 116 NtOpenFile
        \??\C:\RECYCLER\S-1-5-21-0243556031-888888379-781863308-1455\psyjo3.exe
1759 496 Palevo.exe 41 NtCreateKey SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
1760 496 Palevo.exe 247 NtSetValueKey
        \REGISTRY\MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON
1761 496 Palevo.exe 119 NtOpenKey
        \REGISTRY\USER\S-1-5-21-1715567821-1993962763-682003330-1003
1762 496 Palevo.exe 41 NtCreateKey SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
1763 496 Palevo.exe 177 NtQueryValueKey
        \REGISTRY\USER\S-1-5-21-1715567821-1993962763-682003330-1003
        \SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON
1764 496 Palevo.exe 247 NtSetValueKey
        \REGISTRY\USER\S-1-5-21-1715567821-1993962763-682003330-1003
        \SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON
1765 496 Palevo.exe 41 NtCreateKey Software\Microsoft\Windows\CurrentVersion\Run
1766 496 Palevo.exe 247 NtSetValueKey
        \REGISTRY\USER\S-1-5-21-1715567821-1993962763-682003330-1003
        \SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN
                                        :
1769 496 Palevo.exe 277 NtWriteVirtualMemory PID: 1880, ProcessName: explorer.exe
1770 496 Palevo.exe 277 NtWriteVirtualMemory PID: 1880, ProcessName: explorer.exe
1771 496 Palevo.exe 277 NtWriteVirtualMemory PID: 1880, ProcessName: explorer.exe
1772 496 Palevo.exe 53 NtCreateThread PID: 1880, ProcessName: explorer.exe
1773 496 Palevo.exe 257 NtTerminateProcess PID: 496, ProcessName: Palevo.exe
                                        :
```

図 4: Palevo.exe のシステムコール取得結果

- キャッシュや特定のドメインに対するセキュリティレベルなど, Internet Settings の書換え
- ネットワークを扱うデバイスファイルへのアクセス

3.4 考察

検体の解析結果から, マルウェアのシステムコールをフックし, 実際に操作されるオブジェクトの情報を取得できることが確認できた. これにより, ファイルやレジストリの改竄, 別のプロセスを利用した攻撃などの基本的な挙動を解析できた. したがって, Alkanet がマルウェア解析に有効であることが確認できた. 一方, 現在の実装では, マルウェアによる攻撃を受けたプロセスの挙動が, 正規の挙動であるか否かの判断が難しいことが確認された. ただし, スレッドやイメージに関する情報を用いることで解決可能である.

今後の課題として, カーネルモードで動作するマルウェアへの対応が挙げられる. このようなマルウェアは, システムコールを使用せず, カーネルの関数を直接呼び出す. したがって, システムコールのフックによる解析が困難である. この問題は, 各カーネル関数のエントリポイントでのフックを行うことによって解決する. また, カーネルモードマルウェアは, カーネルのメモリ空間にアクセスできるため, オブジェクトの隠蔽や改竄といったことも可能である [6]. したがって, Alkanet の利用するオブジェクトの保護, 隠蔽されたオブジェクトの追跡などの対応が必要である.

4 関連研究

FFR GreenKiller[7] は, カーネルモードで動作するマルウェアを自動解析するツールである. Alkanet と同じく, BitVisor をベースとしている. 株式会社フォティーンフォティ技術研究所によって, 研究・開発が行われている. FFR

```
                                        :
2204 812 Bredolab.exe 37 NtCreateFile \??\PIPE\lsarpc
2205 1000 lsass.exe 183 NtReadFile \lsass
2206 812 Bredolab.exe 274 NtWriteFile \lsarpc
2207 1000 lsass.exe 274 NtWriteFile \lsass
2208 1000 lsass.exe 183 NtReadFile \lsass
2209 812 Bredolab.exe 183 NtReadFile \lsarpc
2210 1000 lsass.exe 183 NtReadFile \lsass
                                        :

3157 812 Bredolab.exe 116 NtOpenFile
        \??\STORAGE#Volume#1&30a96598&0&Signature472E472DOffset7E00Length752C56200
        #{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
                                        :
3168 812 Bredolab.exe 37 NtCreateFile \??\MountPointManager
3169 812 Bredolab.exe 41 NtCreateKey
        Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2
        \{a66607c8-a9f7-11df-a9a4-806d6172696f}\
3170 812 Bredolab.exe 247 NtSetValueKey
        \REGISTRY\USER\S-1-5-21-1715567821-1993962763-682003330-1003\SOFTWARE\MICROSOFT
        \WINDOWS\CURRENTVERSION\EXPLORER\MOUNTPOINTS2\{A66607C8-A9F7-11DF-A9A4-806D6172696F}
                                        :

3359 812 Bredolab.exe 48 NtCreateProcessEx \WINDOWS\system32\cmd.exe
                                        :
3667 1256 cmd.exe 116 NtOpenFile \??\C:\DOCUME~1\yotuki\MYDOCU~1\Bredolab.exe
3668 1256 cmd.exe 224 NtSetInformationFile DELETE: \DOCUME~1\yotuki\MYDOCU~1\Bredolab.exe
                                        :
```

図 5: Bredolab.exe のシステムコール取得結果

GreenKiller は, デバイスドライバとして動作するマルウェアを監視し, カーネルの関数の呼び出しをトレースし, ログを出力する.

FFR GreenKiller がカーネルモードで動作するマルウェアを対象としているのに対し, Alkanet はユーザモードのマルウェアを対象としている点が異なる. また, 今後は, Alkanet もカーネルモードのマルウェアに対応していく予定である.

5 おわりに

本論文では, VMM として動作し, Windows 上で発行されるシステムコールをフックし, マルウェアの挙動を監視するシステム Alkanet について述べた. さらに実際にマルウェアを動作させて得た解析結果と, その考察を述べた. 今後の課題として, 別のプロセスを利用した攻撃や, カーネルモードで動作するマルウェアへの対応が挙げられる.

参考文献

[1] 野村 他: “仮想計算機モニタを使ったマルウェアの挙動解析,” CSS2009, Vol.2009, No.11, pp.787–792, 情報処理学会 (2009).

[2] 筑波大学: “The Homepage of BitVisor,” http://www.bitvisor.org/ (2009).

[3] 畑田 他: “マルウェア対策のための研究用データセット ～MWS 2010 Datasets～,” MWS2010 (2010).

[4] ClamAV: “Clam AntiVirus,” http://www.clamav.net/ (2010).

[5] US-CERT: “US-CERT Technical Cyber Security Alert TA09-020A – Microsoft Windows Does Not Disable AutoRun Properly,” http://www.us-cert.gov/cas/techalerts/TA09-020A.html (2009).

[6] J. Butler: “DKOM (Direct Kernel Object Manipulation),” Black Hat Windows Security 2004, http://www.blackhat.com/presentations/win-usa-04/bh-win-04-butler.pdf (2004).

[7] 村上: “FFR GreenKiller - Automatic kernel-mode malware analysis system,” AVAR2009, http://www.fourteenforty.jp/research/research_papers/avar-2009-murakami.pdf (2009).