L-030

リソースアクセス情報に基づく未知のマルウェア検知手法

Unknown Malware Detection Method based on Resource Access Information

李 在炯† 城間 政司† 丹田 賢‡ 梅橋 一充‡
Jaehyeong Lee Tadashi Siroma Satoshi Tanda Kazumi Umehashi
長田 智和† 谷口 祐治† 名嘉村 盛和†
Tomokazu Nagata Yuji Taniguchi Morikazu Nakamura

1. はじめに

Web 上にはマルウェアが氾濫しており，それらを検知するためにパターンマッチング方式を用いるマルウェア検知方式が利用されている．しかし，亜種も含め新種のマルウェアは 1 日に数万種生み出されており[1]，従来のパターンマッチング方式では，0-day 攻撃に対応できない問題が起きている．そのため，単純なパターンマッチング方式に加え，ヒューリスティック方式が研究・開発されている[2][3]．ヒューリスティック方式の場合，マルウェアの振る舞いを検知ロジックとして組み込んでいるため，新種のマルウェア検知に対して有効である．しかし，この方式ではマルウェアが取る可能性の高い行動パターンを収集するため，マルウェアに似た行動を取る正常なプログラムをマルウェアと誤検知してしまう可能性がある．そのため，ヒューリスティック方式では，誤検知を避けるためにマルウェアとして判断する閾値を高く設定しなければならず，検知率を向上させることが難しいという問題がある．

ヒューリスティック方式の検知率を向上させるため，複数の検知手法を組み合わせる方法が考えられる．本研究では，リソースアクセス情報を利用した新たな検知手法を提案し，従来のヒューリスティック方式と異なる手法により一定の検知率を示すことで，従来のヒューリスティック方式の検知エンジンへの組み込みを検討する価値があることを確認する．

本稿の構成は以下の通りである．まず，2 節では，従来のマルウェア検知手法としてパターンマッチング方式とヒューリスティック方式について述べる．次に，3 節では，本稿で提案するリソースアクセス情報をもとにしたマルウェア検知手法について述べる．4 節では，本提案手法を評価するための評価方法について述べる．5 節では，リソースアクセス情報からマルウェアと正常系プログラムの特徴を抽出してリソースリストを作成する方法とその評価結果について述べる．6 節と 7 節では，5 節の方法に加え，更に誤検知率を下げる方法とその評価結果について述べる．8 節では，5 節から 7 節までの評価結果に対する考察を行う．最後に，9 節で本稿をまとめる．

2. 検知方式

本節では，既存のマルウェア検知方式であるパターンマッチング方式とヒューリスティック方式について述べる．

2.1 パターンマッチング方式

パターンマッチング方式とは，データベースに登録されたマルウェアのコードパターンを検査対象のファイル内容と照合することでマルウェアを検知する方式である．既知のマルウェアに対しては高い精度で検知でき，誤検知も少ない．一方，新種のマルウェアに対してはコードパターンを解析するまで無力であるため，即座に検知することが困難である．

2.2 ヒューリスティック方式

ヒューリスティック方式とは，プログラムの構造，動作，その他の属性からマルウェアを検知する方式である．「スタティックヒューリスティック方式」と「ダイナミックヒューリスティック方式」の 2 種類の分析方法があり，「スタティックヒューリスティック方式」ではパッカーやアンチデバッグ技術などの構造を中心に，「ダイナミックヒューリスティック方式」ではサンドボックスと呼ばれる仮想空間上でマルウェアの振る舞いを中心に分析する．分析には，マルウェアの特徴に基づいた複数の評価項目を設け，それぞれの結果から算出された評価値により，マルウェアか否かを判断する．そのため，未知のマルウェアに対しても有効であるが，誤検知や見落としが発生するなど問題がある．

3. 提案手法

本研究では，ヒューリスティック方式の課題である検知率を向上させるため，プログラムのリソースアクセス情報を利用した新たな検知手法を提案する．

3.1 概要

本研究でのリソースとはファイルとレジストリを意味し，アクセスとは書き込み，読み込み，削除を意味する．つまり，リソースアクセス情報とは，プログラムが実行される際に書き込み，読み込み，削除処理を行ったファイルとレジストリの情報を意味する．

本提案手法によるマルウェア検知の流れは次のとおりである．まず，マルウェアと正常系プログラムを仮想マシン上で実行し，リソースアクセス情報を事前に収集する．その後，収集したリソースアクセス情報からマルウェアと正常系プログラムの特徴を抽出し，リソースリストを作成する．最後に，作成したリソースリストを基にマルウェア検知を行う．

3.2 リソースアクセス情報収集

まず，マルウェアと正常系プログラムを実際に実行し，リソースアクセス情報を収集する．このときに収集する情報は「ファイル書き込み，ファイル読み込み，ファイル削除，レジストリ書き込み，レジストリ読み込み，レジストリ削除」の 6 種類である．

† 琉球大学大学院理工学研究科
‡ 株式会社フォティーンフォティ技術研究所

表 1　リソースリストに含まれる情報（レジストリ）

パス	名前	マルウェアポイント		
		書き込み	読み込み	削除
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders	Cache	123	173	0
HKEY_CURRENT_USER\Software\Microsoft\IMEJP\8.1\MSIME	deffont	0	-184	0
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Tcpip\Parameters	Domain	0	179	0
HKEY_LOCAL_MACHINE\SYSTEM\WPA\MediaCenter	Installed	0	149	0
HKEY_LOCAL_MACHINE\SYSTEM\Setup	*	0	123	0

表 2　リソースリストに含まれる情報（ファイル）

パス	名前	マルウェアポイント		
		書き込み	読み込み	削除
C:\Windows\Fonts	msgothic.ttc	0	-139	0
C:\Windows\System32	imjp81.ime	0	-128	0
C:\Windows\System32	wininet.dll	0	121	0
C:\	autoexec.bat	0	73	0
C:\Documents and Settings\All Users	desktop.ini	0	64	0

次に，収集作業で得られたリソースアクセス情報から特定のリソースにアクセスした検体の割合を集計する．例えば，マルウェア 1 万体を実行し，そのうち 1,000 体が「C:\test.txt」ファイルに書き込みを行った場合，「C:\test.txt」の書き込み処理のマルウェア利用率は 10%とする．また，「C:\test.txt」ファイルに書き込みを行った正常系プログラムの割合を求め，正常系利用率とする．読み込みと削除に対しても同様にマルウェア利用率と正常系利用率を求める．

3.3　リソースリスト

リソースリストはリソースアクセス情報からマルウェアと正常系プログラムの特徴を抽出して作成したものである．リソースリストに含まれる情報は以下のとおりであり，例を表 1 と表 2 に示す．

・リソースのパス
・リソースの名前
・書き込みのマルウェアポイント
・読み込みのマルウェアポイント
・削除のマルウェアポイント

マルウェアポイントとはそのリソースアクセス情報がどの程度マルウェアと正常系プログラムの特徴を持っているかを表す点数である．マルウェアポイントは正の値と負の値があり，正の値はマルウェアの特徴であることを意味し，負の値は正常系プログラムの特徴を意味する．すなわち，正のマルウェアポイントが高いほどマルウェアがよくアクセスするリソースであり，負のマルウェアポイントが高いほど正常系プログラムがよくアクセスするリソースとなる．

3.4　マルウェア検知

本研究で使用するマルウェア検知エンジンは，プログラムの振る舞いを常時監視しており，実行中のプログラムからリソースアクセスが発生した際，リソースリストから該当するリソースアクセスのマルウェアポイントを取得し，マルウェアポイントの合計を計算する．そして，プログラムの実行中にマルウェアポイントの合計が閾値を超えた場合，そのプログラムをマルウェアと判断する．また，プログラムの実行が終了するまでマルウェアポイントの合計が一定の値を超えない場合，そのプログラムを正常系プログラムと判断する．

3.5　提案手法の目標

株式会社フォティーンフォティ技術研究所[4]では，新たな検知手法の検知率が 30%以上かつ誤検知率が 0.001%以下の場合に，検知エンジンへの組み込みを検討するレベルとして考えている．また，検知率が 30%を超えなくても，従来のヒューリスティック方式とは異なる手法により一定の検知率を示した場合には，検知エンジンへの組み込みを検討する価値があるとしている．

本提案手法は株式会社フォティーンフォティ技術研究所のアンチウイルスソフトで採用しているヒューリスティック方式の検知エンジンに組み込むことができる性能を目標としている．そのため，誤検知率を 0.001%以下に抑えると同時に検知率が 30%を越えることを目標としている．ここで，もし，検知率が 30%を超えない場合は，従来のヒューリスティック方式の検知エンジンに実装されていない新たな手法として，誤検知率を 0.001%以下に抑えると同時に一定の検知率が確認できればよいとする．

4. 評価方法

VMware vSphere[5]で仮想マシンを構築し，仮想マシン上でマルウェアと正常系プログラムのリソースアクセス情報を収集する．その後，収集したリソースアクセス情報をもとにしてマルウェア及び正常系プログラムの特徴を抽出したリソースリストを作成する．このリソースリストをもとに仮想マシン上でマルウェア検知を行う．

4.1　システムの構成

システムは図 3 のようにコントローラー，ホスト，ゲストの 3 種類のマシンで構成される．各マシンの機能の詳細について述べる．

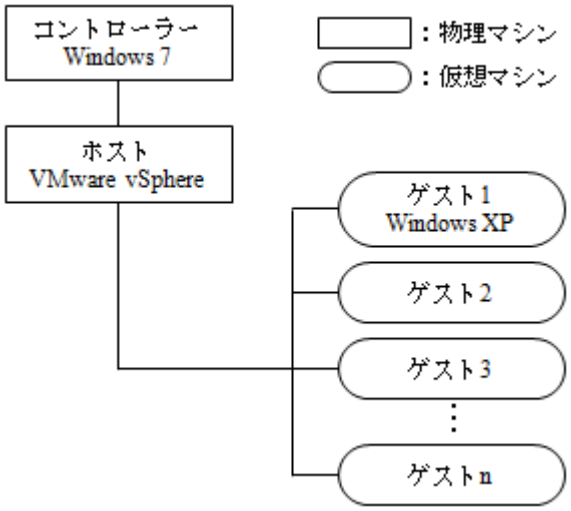

図 3　システムの構成図

（1）コントローラー

コントローラー用情報収集プログラムとリソースアクセス検知エンジンをインストールし，ホスト及びゲストを操作する環境である．

実験で使用するマルウェアと正常系プログラムを保持する．

（2）ホスト

VMware vSphere をインストールする環境である．VMware vSphere をインストールした後は仮想マシンであるゲストを作成する．リソースアクセス情報収集と検知能力評価を効率よく行うため，ゲストを複数作成して並列処理する．

（3）ゲスト

マルウェアと正常系プログラムを実際に実行する仮想マシンのことである．

ゲスト用情報収集プログラムとリソースアクセス検知エンジンをインストールする．また，検体を実行した際のリソースアクセス情報を取得するため，Process Monitor[6]をインストールする．

検体を実行した後，マルウェアに感染した状態で処理を続行した場合，正しいリソースアクセス情報を取得できない可能性がある．この問題を避けるため，ゲストの初期状態で，VMware 上でスナップショットを取っておき，ゲストで検体を実行した後はスナップショットに戻す処理を行う．

4.2　リソースアクセス情報収集

検証用のマルウェア 983,329 体のうち，1 万体をランダムに選択し，リソースアクセス情報収集を行う．また，正常系プログラム 111,311 体のうち，1 万体をランダムに選択し，リソースアクセス情報収集を行う．

リソースアクセス情報収集はゲストで行われる．本研究では Windows XP 環境を用いた．ゲストは事前に Process Monitor をインストールしておき，プログラムの挙動を監視する．Process Monitor はプログラムが行った処理（ファイル，レジストリ，プロセスおよびスレッドの活動）をリアルタイムで表示するツールである．

まず，検証用として用意した検体のうち，マルウェアや正常系プログラムを 1 つ選択し，ゲストにコピーする．その後，コピーした検体をゲストで実行する．ゲストで検体が実行されると，Process Monitor により検体が行った処理が記録される．本提案手法では「ファイル書き込み，ファイル読み込み，ファイル削除，レジストリ書き込み，レジストリ読み込み，レジストリ削除」の 6 種類の処理を利用している．

Process Monitor により記録される処理のうち，本提案手法で利用する情報に該当する処理内容を表 3 に示す．

表 3　リソースアクセス種類と Process Monitor における Operation 種類の関係

リソース種類	アクセス種類	Process Monitor における Operation 種類
ファイル	書き込み	WriteFile
	読み込み	ReadFile
	削除	SetDispositionInformationFile
レジストリ	書き込み	RegSetValue
	読み込み	RegQueryValue
	削除	RegDeleteValue

4.3　マルウェア検知

検証用のマルウェア 983,329 体のうち，1 万体をランダムに選択し，マルウェア検知を行う．また，正常系プログラム 111,311 体のうち，1 万体をランダムに選択し，マルウェア検知を行う．ただし，未知マルウェア検知の有効性を確認するため，リソースアクセス情報収集の対象とした検体をそのまま使用するのではなく，新たに選択した検体をマルウェア検知に使用する．

マルウェア検知もリソースアクセス情報収集と同様の環境で行う．本研究では，Windows XP を用い，事前にマルウェア検知エンジンをインストールした．まず，検証用として用意した検体のうち，マルウェアや正常系プログラムを 1 つ選択し，ゲストにコピーする．その後，コピーした検体をゲストで実行する．ゲストで検体が実行されると，マルウェア検知エンジンによりマルウェアポイントの合計が計算され，マルウェア判定が行われる．

5.　リソースリスト作成と評価

リソースアクセス情報からマルウェアの特徴と正常系プログラムの特徴を抽出し，リソースリストを作成する．その後，リソースリストをもとにマルウェア検知を行い，検知率と誤検知率を確認した．

5.1　特徴抽出方法

5.1.1　マルウェアの特徴

マルウェアでよく見られるリソースアクセスには，正のマルウェアポイントを付与する．

表 3　類似リソースの例 1

パス	名前	マルウェアポイント		
		書き込み	読み込み	削除
HKEY_CLASSES_ROOT\.exe	(default)	0	62	0
HKEY_CLASSES_ROOT\.asp	(default)	0	55	0
HKEY_CLASSES_ROOT\.bat	(default)	0	55	0
HKEY_CLASSES_ROOT\.cer	(default)	0	46	0
HKEY_CLASSES_ROOT\.cmd	(default)	0	46	0

表 4　類似リソースの例 2

パス	名前	マルウェアポイント		
		書き込み	読み込み	削除
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\CurrentVersion\Internet Settings	disableimprovedzonecheck	0	118	0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\CurrentVersion\Internet Settings	sharecredswithwinhttp	0	96	0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\CurrentVersion\Internet Settings	bypasshttpnocachecheck	0	95	0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\CurrentVersion\Internet Settings	bypassssslnocachecheck	0	95	0
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ Windows\CurrentVersion\Internet Settings	dialupuselansettings	0	95	0

次のいずれかに当てはまるリソースアクセス情報をマルウェアの特徴と判断し，正のマルウェアポイントを付与する．

・マルウェア利用率 >= 40% かつ 正常系利用率 < 15%
・マルウェア利用率 >= 30% かつ 正常系利用率 < 10%
・マルウェア利用率 >= 20% かつ 正常系利用率 < 8%
・マルウェア利用率 >= 10% かつ 正常系利用率 < 4%
・マルウェア利用率 >= 5% かつ 正常系利用率 < 1.6%

マルウェアポイント = (マルウェア利用率 - 正常系利用率) × 5

5.1.2　正常系プログラムの特徴

正常系プログラムでよく見られるリソースアクセスには，負のマルウェアポイントを付与する．

次のいずれかに当てはまるリソースアクセス情報を正常系プログラムの特徴と判断し，負のマルウェアポイントを付与する．

・正常系利用率 >= 40% かつ マルウェア利用率 < 20%
・正常系利用率 >= 30% かつ マルウェア利用率 < 15%
・正常系利用率 >= 20% かつ マルウェア利用率 < 11%
・正常系利用率 >= 10% かつ マルウェア利用率 < 5%
・正常系利用率 >= 5% かつ マルウェア利用率 < 1.6%

マルウェアポイント = (正常系利用率 - マルウェア利用率) × -5

5.2　検知能力評価

特徴抽出方法により作成したリソースリストをマルウェア検知に利用した際の検知率と誤検知率を確認した．その結果を表 5 に示す．

表 5　評価結果 1

検知率	34.95%
誤検知率	4.00%

検知率が 34.95%であり，検知率が 30%を超えたため，検知率の目標を達成している．しかし，誤検知率が 4.00%となり，誤検知率 0.001%以下という条件を満たしていない．そのため，リソースアクセス情報から特徴抽出方法にてリソースリストを作成するだけでは，本提案手法を従来のヒューリスティック方式の検知エンジンに組み込むことはできず，更に誤検知率を下げるための方策が必要である．

6. 誤検知防止手法と評価

5.1 節で述べた特徴抽出方法のみでリソースリストを作成した場合，誤検知率が高い問題があり，従来のヒューリスティック方式の検知エンジンに組み込むためには誤検知率を下げる必要がある．この節では類似リソースによる誤検知とその誤検知を防止する手法を説明し，その評価結果について考察する．

6.1　誤検知防止手法

6.1.1　類似リソース圧縮

リソースにはパスや名前が類似しているものがあり，検体は実行時に類似しているリソース全てにアクセスす

るか，一部にアクセスすることがある．正常系プログラムの場合，正のマルウェアポイントが付いている類似リソースにアクセスすると，マルウェアポイントの合計が一気に高くなり，誤検知の原因となる．このような誤検知を防止するため，類似している複数のリソースを 1 つのリソースとして扱うことが考えられる．

類似している複数のリソースを 1 つのリソースにする方法を以下で説明する．

(1)　パスが部分一致・名前が一致

パスの一部だけ異なる複数のリソースがある場合，異なる部分にワイルドカードを使用することで，一つのリソースとして扱うことができる．その例を表 3 に示す．

(2)　パスが一致

パスが一致し，名前が異なる複数のリソースがある場合，名前にワイルドカードを使用することで，一つのリソースとして扱うことができる．その例を表 4 に示す．

6.1.2　類似リソース圧縮の対象

類似リソース圧縮を行うと，そのリソースにアクセスする検体のマルウェアポイントの合計値が低くなる．そのため，全てのリソースに対して類似リソース圧縮を行うと誤検知率だけでなく，検知率も大きく下がることになる．なるべく検知率を下げないよう類似リソース圧縮の対象となるリソースを選択する．

類似リソース圧縮の対象となるリソースを選択する方法は，まず，正常系プログラムのみ検知能力評価を行い，そこで誤検知した検体だけを選択する．その後，誤検知の原因となったリソースを抽出し，類似リソース圧縮の対象とする．誤検知の原因となったリソースとは，誤検知した正常系プログラムでアクセスしたリソースのうち，正のマルウェアポイントが付いているリソースのことである．

6.2　検知能力評価

「5.1 特徴抽出方法」にて作成したリソースリストに「6.1 誤検知防止手法」を加え，リソースリストを改良した．そのリソースリストを利用し，マルウェアと正常系プログラムに対してマルウェア検知を行った際の検知率と誤検知率を確認した．その結果を表 6 に示す．

表 6　評価結果 2

検知率	10.00%
誤検知率	0.03%

表 5 の評価結果 1 と比べ，誤検知率が 4.00%から 0.03%に下がり，類似リソース圧縮による誤検知率の減少が確認できた．その反面，検知率も 34.95%から 10.00%に下がる結果となった．類似リソース圧縮を利用した手法を従来のヒューリスティック方式の検知エンジンに組み込む際の有効性を確認するためには，現状の検知率を維持すると同時に誤検知率を 0.001%以下に抑える必要がある．

7. 静的解析情報利用

株式会社フォティーンフォティ技術研究所のヒューリスティック方式の検知エンジンには，スタティックヒューリスティック方式による静的解析エンジンが含まれており，本提案手法を従来のヒューリスティック方式の検知エンジンに組み込むことで，検体を静的解析した情報を利用することができる．

株式会社フォティーンフォティ技術研究所の静的解析エンジンは検体を静的解析し，脅威レベルを数値化して管理しており，脅威レベルが低いほど正常系プログラムの特徴を持っていることを意味し，脅威レベルが高いほどマルウェアの特徴を持っていることを意味する．表 6 の評価結果 2 の検知能力評価で誤検知した正常系プログラムを静的解析エンジンで解析した結果，脅威レベルが一定の数値より低いことが分かった．このことを利用し，本提案手法でマルウェアと判定しても静的解析による脅威レベルが閾値以下であれば，正常系プログラムと判定することで，更に誤検知率を下げることができる．

本提案手法に加え，静的解析情報を利用した際の検知率と誤検知率を表 7 に示す．

表 7　評価結果 3

検知率	9.05%
誤検知率	0.00%（誤検知 0 体）

誤検知率が 0.00%となり，誤検知率を 0.001%以下に抑える目標が達成できた．それと同時に検知率が 9.05%となったことで一定の検知能力があることも確認できた．

8. 考察

本研究では，評価結果 1 から 3 まで計 3 回の評価を行った．評価結果 1 は，リソースアクセス情報からマルウェアと正常系プログラムの特徴を抽出して作成したリソースリストによる検知能力評価である．検知率は 30%を超えているものの，誤検知率が 4.00%と目標値である 0.001%を大きく上回っており，類似リソースを個別に扱う単純なリソースリストでは誤検知率に問題があることが読み取れる．

評価結果 2 は，類似リソースの圧縮を行ったリソースリストによる検知能力評価である．評価結果 1 と比べ，誤検知率が 4.00%から 0.03%に下がり，類似リソース圧縮による誤検知率の減少が確認できた．一方で検知率は 34.95%から 10.00%に下がり，類似リソース圧縮の対象として誤検知の原因となったリソースを選択しても，そのリソースにアクセスするマルウェアも多く存在し，検知率が大きく減少することが分かった．

評価結果 3 は，更に誤検知率を下げるため，静的解析エンジンによる静的解析情報を利用した場合の検知能力評価である．評価結果 3 では，検知率が 9.05%，誤検知率が 0.00%となった．この結果から，本提案手法は静的解析情報を利用することで，検知率の減少を 1%未満に抑えながら，誤検知率を 0.00%にまで低減できることが分かった．

以上の本研究により，本提案手法が一定の検知率を持ち，かつ誤検知率を 0.001%以下に抑えられることが確認できた．また，本提案手法が従来のヒューリスティック方式とは異なる新たな手法であることから，本提案手法に，従来のヒューリスティック方式の検知エンジンへの組み込みを検討する価値があることを確認した．

9. まとめ

本研究では，マルウェア検知に利用する特徴としてプログラムのリソースアクセス情報を利用することを考えた．それを実現するため，まず，ゲストでマルウェアと正常系プログラムを実際に実行し，リソースアクセス情報を収集した．収集したリソースアクセス情報からはマルウェアと正常系プログラムの特徴を抽出し，その特徴をマルウェア検知に利用した．

検知能力評価では，リソースアクセス情報の収集に利用したマルウェアと正常系プログラムとは別の検体に対しマルウェア検知を行うことで，未知のマルウェア検知の有効性を確認した．また，株式会社フォティーンフォティ技術研究所のアンチウイルスソフトで採用しているヒューリスティック方式の検知エンジンに組み込むことができる性能目標を設定し，本提案手法を従来の検知エンジンへ組み込むことを検討する価値があることを確認した．

謝辞

本研究は，公益財団法人沖縄県産業振興公社の「おきなわ新産業創出投資事業」の支援を受け，株式会社フォティーンフォティ技術研究所と共同で行われたものである．

本研究を進めるにあたって，有益な助言と協力を頂いた関係者各位に深く感謝致します．

参考文献

[1] McAfee 脅威レポート:2010 年第三四半期
http://www.mcafee.com/japan/media/mcafeeb2b/international/japan/pdf/threatreport/threatreport10q3.pdf

[2] 神薗 雅紀，白石 善明，森井 昌克，“仮想ネットワークを使った未知ウィルス検知システム，”情報処理学会研究報告　コンピュータセキュリティ，Vol.16，No.22，pp. 113-120（2003）

[3] 岩本 一樹，和崎 克己，“コンピュータウイルスのコード静的解析による特徴抽出と分類について，” 電子情報通信学会技術研究報告，Vol.107，No.397，pp.107-113（2007）

[4] 株式会社フォティーンフォティ技術研究所
http://www.fourteenforty.jp/

[5] VMware. Inc.
http://www.vmware.com/

[6] Windows Sysinternals TechCenter
http://technet.microsoft.com/ja-jp/sysinternals/bb896645.aspx