ハイパバイザ内シグネチャマッチングによるマルウェア検出

大山 恵弘 †　　河崎 雄大 †

† 電気通信大学
182-8585 東京都調布市調布ヶ丘 1-5-1
oyama@inf.uec.ac.jp　　kawasaki@ol.inf.uec.ac.jp

あらまし マルウェア検出機能を有するハイパバイザ BVMD によって，MWS 2012 のマルウェアデータセットのマルウェア検体を検出した実験について報告する．BVMD は準パススルー型ハイパバイザである BitVisor を拡張して実装されている．BVMD はハードディスクなどのデバイスとゲスト OS との間を流れるデータブロックに対してシグネチャマッチングを適用する．

Malware Detection by Signature Matching in a Hypervisor

Yoshihiro Oyama†　　Yudai Kawasaki†

†The University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, JAPAN
oyama@inf.uec.ac.jp　　kawasaki@ol.inf.uec.ac.jp

Abstract We report the result of experiments in which we detected malware in the MWS 2012 malware dataset by using BVMD, a hypervisor that provides a malware detection mechanism. BVMD is implemented by extending a parapass-through hypervisor BitVisor. BVMD applies signature matching against data blocks that are transmitted between the guest OS and devices such as hard disks.

1　はじめに

マルウェアによる被害を防ぐための一つの効果的な方法はアンチウィルスの利用である．実際，アンチウィルスは広く普及している．ただし，組織におけるアンチウィルスの運用にあたっては注意が必要な場合がある．特に，組織の管理者が，組織のメンバーの用いる各 OS でアンチウィルスが確実に動作している状態を作りたい場合に，問題が生じることがある．各メンバーに OS の管理者権限を与えた場合，速度低下などの理由から，一部のメンバーはアンチウィルスを隠れてアンインストールしたり停止したりするかもしれない．また，メンバーがそのようなことをしない場合でも，ゼロデイのマルウェアがメンバーの PC 上の OS に感染し，アンチウィルスを無効化するなどの攻撃を行った場合には，同じ状況に陥る．たとえば，マルウェアの Conficker は，アンチウィルスの動作を妨害する処理を実行することが知られている [8]．各メンバーの OS の管理者権限を組織の管理者が持ち，各メンバーは一般ユーザ権限のみしか使えないようにするという運用もありうる．しかしこの運用では各メンバーによるソフトウェアのインストールやアップデートが制限され，利便性を大きく損なう．

この問題を解決するシステムとして，著者らは今までに BVMD [6] を提案している．BVMD はマルウェア検出機能を持つ仮想マシンモニタ（VMM）であり，OS の下でセキュリティ処理

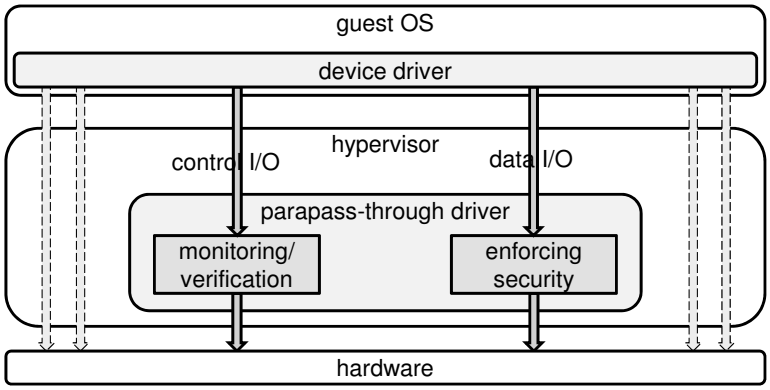

図 1: BitVisor の構成

を実行することを可能にする．BVMD はゲスト OS とハードウェアデバイスとの間でやりとりされる入出力データや，ゲスト OS のメモリデータを検査し，ゲスト OS 内のマルウェアを検出する．BVMD は BitVisor [10] を改造して作られている．BitVisor および BVMD は実ハードウェア上で直接動作するハイパバイザであり，自身の上でゲスト OS を同時にただ 1 つのみ動作させることができる．

組織の管理者は BVMD を用いて，以下のように，アンチウィルス機能の利用をメンバーに強制する．組織の管理者は各メンバーの PC に BVMD をインストールする．BVMD の管理者権限は組織の管理者が保持する．各メンバーは BVMD の上に（ゲスト）OS をインストールする．ゲスト OS の管理者権限は各メンバーが保持する．ゲスト OS は，BVMD の中のアンチウィルス機能によって守られる．メンバーは自分の OS 内にアンチウィルスをインストールする必要はない．メンバーはゲスト OS の管理者権限を保持しているので，ソフトウェアのインストールや更新を自由に行える．なお，BitVisor の暗号化機能により，BVMD を用いずに（すなわち，実ハードウェアの上で直接）ゲスト OS を実行できないようにすることができる．

本研究では，マルウェア対策研究人材育成ワークショップ 2012 (MWS2012) が提供する研究用データセット [5] の検体を BVMD によって検査した結果を報告する．具体的には，データセットに含まれる一部のマルウェア検体のファイルを，BVMD 上で動作するゲスト OS 上に作成し，BVMD がそれらのマルウェアを検出するかどうかを観測した．

本論文の構成は以下のとおりである．2 章では BitVisor と BVMD の概要を説明する．3 章では実験結果を示す．4 章では関連研究を述べ，5 章ではまとめと今後の課題を述べる．

2 提案システム

2.1 BitVisor

BVMD がベースにしている BitVisor について述べる．BitVisor は，ハードウェア上で直接動作するセキュリティ向上のための準パススルー型 VMM である．準パススルー型とは，ゲスト OS からハードウェアへのアクセスを可能な限りパススルー（通過）させつつ，セキュリティ機能の実現のために最低限必要なアクセスのみを VMM が捕捉する方式である．BitVisor は，一部のアクセスのみを捕捉することによって，ストレージやネットワークの暗号化などのセキュリティ機能を実現している．準パススルー型の利点には，実行時間のオーバーヘッドが小さいことや，Trusted Computing Base (TCB) を小さく抑えやすいことなどがある．

BitVisor の構成を図 1 に示す．hypervisor と書かれている部分が BitVisor である．図中の実線矢印は BitVisor に捕捉される I/O 処理を表し，点線矢印は捕捉されない I/O 処理を表している．捕捉されない I/O 処理はゲスト OS のデバイスドライバによって実行される．捕捉さ

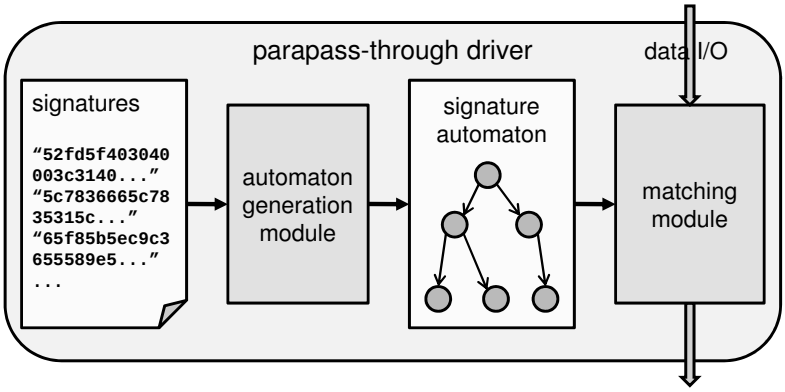

図 2: シグネチャマッチングを行う準パススルードライバ

れる I/O 処理に関しては，BitVisor 内の準パススルードライバ（図中では parapass-through driver）と呼ばれるデバイスドライバがデバイスを制御し，ゲスト OS が発行する制御 I/O とデータ I/O を捕捉する．ここで，制御 I/O はデバイスによるデータ転送を制御するための I/O で，転送するデータの場所やアクセス方法，データ転送の開始，終了などを指定する I/O である．また，データ I/O は実際にデータ転送を行う I/O である．BitVisor は制御 I/O を捕捉してアクセスの状態を把握し，データ I/O を捕捉してデータの取得や更新を行う．

BitVisor はストレージデータの暗号化や VPN 構築の機能は提供しているものの，マルウェアを検出する機能は備えていない．

2.2 BVMD

BVMD は，ゲスト OS とデバイスとの間で転送されるデータからマルウェアを検出する機能を提供する．BVMD では BitVisor の準パススルードライバが拡張されており，そのドライバは，捕捉したデータ I/O に対してシグネチャマッチングを適用してマルウェアを検出する．

拡張された準パススルードライバの構成を図 2 に示す．その準パススルードライバはオートマトン生成部（automaton generation module）とマッチング部（matching module）から構成される．オートマトン生成部では，マルウェアシグネチャの集合を受け取り，それを元に，マッチング処理に適したメモリ上のデータ構造であるオートマトンを生成する．オートマトンの生成では Aho-Corasick 法 [1] と呼ばれるアルゴリズムを用いる．マッチング部では，準パススルードライバが捕捉したデータのバイト列とオートマトンとの間でマッチング処理を行う．

BVMD は，ClamAV [2] が提供するマルウェアシグネチャを加工したものを，自身のマルウェアシグネチャとして用いる．そのマルウェアシグネチャは BVMD のバイナリコードに静的データとして埋め込まれる．本来はマルウェアシグネチャは VMM のバイナリファイルとは独立したファイルとして表現するほうが管理しやすい．しかし，BitVisor および BVMD では，VMM が利用可能なファイルシステムが仮想マシンの外には存在しないため，このような実装となっている．なお，マルウェアシグネチャを新しいものに更新するための機能は既に BVMD に組み込まれている [14]．その機能を用いると，BVMD もゲスト OS も止めることなくシグネチャを更新できる．

BVMD はデータ I/O を流れる低レベルデータに対して単純なシグネチャマッチングを適用する．たとえばハードディスクの I/O では，ディスクブロックに含まれるバイト列をマルウェアシグネチャと照合する．言い換えれば，BVMD はゲスト OS のファイルシステムの構造やプロセスのメモリレイアウトを意識しない．これはゲスト OS の実装を意識した検出を行えないという制限をもたらすものの，ゲスト OS の実装に依存しない検出を可能にするという利点ももたらす．実際，文献 [6] では，ゲスト OS が Windows

main.hdb:

```
d0e0c049ed7056eac8bb396429795010:162516:Worm.Kido-160
```

main.mdb:

```
12288:b0df5fa4a5e588c6e8643326536ca29c:Trojan.Agent-71044
```

main.db:

```
Worm.Blaster.A (Clam)=2077616e04edffffff746f20736179204c4f564520594f55...
```

main.ndb:

```
Trojan.Dropper-18535:1:EP+0:807c2408010f85c201000060be005000108dbe00c0ffff57
```

図 3: ClamAV が提供するマルウェアシグネチャ

であっても Linux であってもマルウェアを検出できたことが報告されている．

ハードウェアの上で直接動作する VMM を利用してセキュリティ機能を実現する際には，脅威の存在や脅威の詳細をユーザにどう伝えるかが技術的課題となる．BVMD がマルウェアを検出した際には，以下の処理のどれかまたは全てが実行される．各処理を実行するかしないかは，BVMD のコンパイル時に指定する．

1. シリアルポートに警告の文字列を出力する
2. ゲスト OS のデスクトップに警告の画像を表示する
3. データブロックの中のマルウェアシグネチャに一致した部分をゼロフィルする

1 の処理は，別の計算機をシリアルポートに接続して監視ができる環境では有効である．2 の処理はゲスト OS の VRAM のフレームバッファ領域を VMM が直接書き換えることによって実現する．実装の詳細は文献 [13] に述べられている．この処理は実計算機のグラフィクスハードウェアには依存するが，ゲスト OS には依存しないという利点がある．3 の処理はデータの内容を書き換えるものであるため，ゲスト OS やゲスト OS 上のアプリケーションの動作を変える可能性がある．ゲスト OS からは，ハードディスクなどのデバイスが，本来そうあるべきではないデータを OS に返しているように見えることになる．3 の処理は，高い安全性を実現することができるが，セマンティクスの維持や安定性とのトレードオフになる．

2.3 ClamAV のマルウェアシグネチャ

ClamAV のマルウェアシグネチャは複数のファイルで表現されている．個々のファイルは，別々の方法によってマルウェアシグネチャを表現している．たとえば拡張子が .hdb や .mdb であるファイルは，ハッシュ値によりマルウェアを表現している．拡張子が .db や .ndb であるファイルは，バイト列のパターンによりマルウェアを表現している．

ClamAV が提供するマルウェアシグネチャのファイルからの抜粋を図 3 に示す．ファイル main.hdb のシグネチャでは，マルウェアのファイルのハッシュ値とサイズ，マルウェア名が記載されている．ファイル main.mdb のシグネチャでは，マルウェアのファイルの PE セクションのサイズとハッシュ値，マルウェア名が記載されている．ファイル main.db のシグネチャでは，マルウェア名と，マルウェアを特徴づけるバイト列が記載されている．ファイル main.ndb のシグネチャでは，マルウェア名，ファイルの種別，マルウェアを特徴づけるバイト列のパターンとその出現場所が記載されている．

現在の BVMD では，ClamAV のマルウェアシグネチャのうち，バイト列のパターンで表現されたもののみ（拡張子が .db および .ndb であるファイルのもの）を再利用している．上述

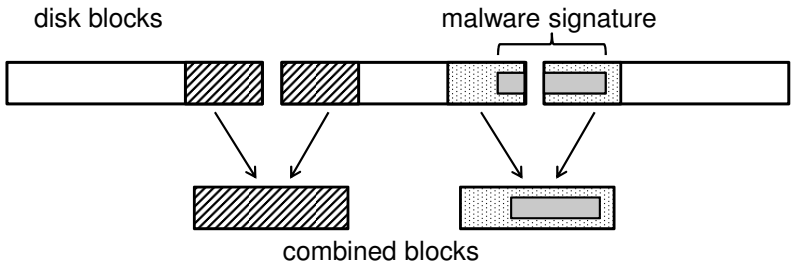

図 4: ブロックの境界に存在するマルウェアの検出

したように，BVMD はディスクブロックなどのデバイスレベルのデータを扱うことに特化しているため，ファイルを意識した検査を行うことができない．よって，ファイルのハッシュ値で表現されたマルウェアシグネチャを BVMD で再利用することは困難である．

マルウェアシグネチャの一部（拡張子が .ndb であるファイルのもの）には，ファイル内のパターンが出現する場所を指定しているものもあるが，BVMD では，そのようなマルウェアシグネチャも利用している．ただし，出現場所の情報は無視し，指定のパターンが出現するかしないかのみによって判断する．その結果，誤検出が出ることがあるとともに，本来不必要なマッチング処理をすることにもなる．しかし，我々は現段階では，たとえ誤検出やオーバヘッドが増えてもマルウェアを見逃さないことを優先させている．

2.4 低レベルデータとマルウェア検出

ディスクブロックなどの低レベルデータに対してシグネチャマッチングを適用すると，ファイルなどの高レベルデータに対して適用する場合と比較して，検出の精度は下がる．以下ではどのような原因によって精度が下がるかを述べる．

まず，誤検出（false positive）が増える場合がある．ディスクブロックに含まれるゴミデータと正規のデータの組み合わせが偶然にマルウェアシグネチャと一致した場合には，そのデータがマルウェアと誤検出される．ただ，このような誤検出が起きる確率は非常に低いと考えられる．

マルウェアの見逃し（false negative）が増える場合もある．それは，マルウェアが複数のブロックにまたがって格納された場合である．BVMD は，まず個々のブロックに対してシグネチャマッチングを行う．さらに，最も近い過去にアクセスされたブロック 1 つを記憶し，それと現在のブロックを結合させ，ブロック境界付近の部分のデータに対してシグネチャマッチングを行う（図 4）．よって，マルウェアがブロック境界をまたがって保存されても，マルウェアシグネチャを検出することはできる．ただし，結合させるのは過去の 1 ブロックのみであるため，マルウェアがまたがる 2 つのブロックが時間的に連続してアクセスされない限り，検出できないという問題がある．BVMD がファイルシステムフォーマットを解釈できるようにすればこの問題はなくなるが，前述のように，ゲスト OS への非依存性を重視し，BVMD ではその方法は採用していない．

当然ながら，ファイルを暗号化するファイルシステムがゲスト OS で用いられている場合には，BVMD はマルウェアを検出できない．より一般的には，暗号化ファイルシステムに限らず，ファイルのデータをそのままディスクブロックに格納する方式をとらないファイルシステムでは，マルウェア検出は成功しない．ただ，少なくとも，Linux で標準的に用いられている ext3 ファイルシステムや Windows で標準的に用いられている NTFS ファイルシステムでは，暗号化機能をオフにしている限りは，マルウェアを検出できることは既に確認している．

3 実験

BVMD を用いて MWS 2012 のマルウェアデータセットのマルウェア検体を検出する実験

表 1: 実験環境

computer	Dell Optiplex 990
CPU	Intel Core i7-2600 3.8 GHz
chipset	Intel Q67 Express
memory	16 GB
hard disk	Seagate ST3320413AS
VMM	BitVisor 1.2
guest OS	Ubuntu 12.04, Linux 3.2.0-29-generic-pae

を行った．実験環境を表 1 に示す．

データセット中には 10538 個のマルウェア検体ファイルが存在する．まず，BVMD を用いずに，OS 上でアンチウィルスソフトウェア ClamAV を動作させて，それらの検体のファイルを検査した．ClamAV のバージョンは 0.97.5 であり，ClamAV のマルウェアシグネチャは 2012 年 8 月 16 日時点のものである．マルウェアと判定されたファイルは 10527 個，マルウェアではないと判定されたファイルは 11 個であった．上記の 10527 個のファイルの中には，ファイルの内容が互いに異なるが同一種類のマルウェアと判定されているものが多く存在した．そのような重複を 1 つとして数えると，検出されたマルウェアは 425 種類であった．

次に，425 種類のマルウェアが ClamAV のどのマルウェアシグネチャにマッチしたのかを調べた．202 種類のマルウェアはハッシュ値の一致により検出されており，223 種類のマルウェアはバイト列のパターンの一致により検出されていた．それらのパターンのうち，ワイルドカードを用いているものは 218 種類であり，用いていないものは 5 種類であった．この 5 種類のマルウェアは，具体的には，Trojan.Crypt-106, Trojan.Downloader-59911, Trojan.Dropper-18535, Trojan.Dropper-20380, Worm.Autorun-1883 である．

上記 5 種類のうち，Trojan.Downloader-59911, Trojan.Dropper-20380, Worm.Autorun-1883 の 3 種類は，著名なパッカーである UPX でパックされていることがわかった．ClamAV は，特に指示しなくても，UPX などの主要なパッカーを想定した展開を試みるため，これらのマルウェアも検出することができる．展開をオフにするオプションを与えたところ，ClamAV もこれらのマルウェアを検出しなくなった．

残りの 2 種類のマルウェアは，展開をオフにしても ClamAV により検出される．10538 個の検体のうち，これらのマルウェアであると ClamAV が判定したファイルは 114 個（Trojan.Crypt-106 が 1 個，Trojan.Dropper-18535 が 113 個）である．これらのマルウェアのファイルは，BVMD により検出できる可能性が高い．そこで，BVMD の上で動作するゲスト OS 上にこれらのファイルを作成し，BVMD がそれを検出できるかどうかを観測した．その結果，114 個のマルウェアのファイル全てをマルウェアとして検出できたことを確認した．

4 関連研究

VMwatcher [4] は，VM 上で動作するゲスト OS のハードディスクやメモリ上の情報をホスト OS で動くアンチウィルスが検査することを可能にするシステムである．VMwatcher ではホスト OS 上で動くプログラムがアンチウイルスの機能を提供しているが，BVMD ではハードウェア上で直接動く VMM がアンチウィルスの機能を提供している．BVMD ではホスト OS を持たないことにより，マルウェアシグネチャを管理しにくいなどの制限も生じるが，組織に従わないユーザやマルウェアがホスト OS を悪用する可能性を排除できるという利点も生まれる．

ホスト OS や他のゲスト OS を仮定した VMM ベースのセキュリティシステムは，VMwatcher 以外にも多数提案されている．たとえば，Zhang らのシステム [12]，Trend Micro Deep Security [11]，Livewire [3]，Lares [7] などがある．これらのシステムでは，BitVisor や BVMD に比べて TCB が大きくなりやすい．

SecVisor [9] は，BitVisor や BVMD と同じく，ハードウェア上で直接動作する小さい VMM によってセキュリティを確保するシステムである．SecVisor は，ゲスト OS のカーネルレベルで動作するコードの integrity を保証する機能を提供

するが，ゲストOS内のデータに対してシグネチャマッチングを行う機能は提供しない．

5 まとめと今後の課題

本論文では，BitVisorをマルウェア検出機能で拡張したシステムであるBVMDにより，MWS 2012のマルウェアデータセットに含まれる検体を検出する実験を行った結果を報告した．

今後の課題には様々なものがある．第1の課題は，正規表現を用いたマルウェアシグネチャも扱えるようにすることである．第2の課題は，メモリ上のデータや，ネットワークパケットなど，ストレージ以外のデータに対してもマルウェア検出の実験を行うことである．

謝辞

本研究の一部はJSPS科研費23700032の助成を受けたものです．

参考文献

[1] Alfred V. Aho and Margaret J. Corasick. Efficient String Matching: An Aid to Bibliographic Search. *Communications of the ACM*, 18(6):333–340, 1975.

[2] Clam AntiVirus. `http://www.clamav.net/`.

[3] Tal Garfinkel and Mendel Rosenblum. A Virtual Machine Introspection Based Architecture for Intrusion Detection. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*, 2003.

[4] Xuxian Jiang, Xinyuan Wang, and Dongyan Xu. Stealthy Malware Detection and Monitoring through VMM-Based "Out-of-the-Box" Semantic View Reconstruction. *ACM Transactions on Information and System Security*, 13(2), 2010.

[5] MWS2012 実行委員会. 研究用データセット MWS 2012 Datasets について. `http://www.iwsec.org/mws/2012/about.html#datasets`.

[6] Yoshihiro Oyama, Tran Truong Duc Giang, Yosuke Chubachi, Takahiro Shinagawa, and Kazuhiko Kato. Detecting Malware Signatures in a Thin Hypervisor. In *Proceedings of the 27th ACM Symposium on Applied Computing*, pages 1807–1814, 2012.

[7] Bryan D. Payne, Martim Carbone, Monirul Sharif, and Wenke Lee. Lares: An Architecture for Secure Active Monitoring Using Virtualization. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 233–247, 2008.

[8] Phillip Porras, Hassen Saidi, and Vinod Yegneswaran. An Analysis of Conficker Logic and Rendezvous Points. Technical report, SRI International, 2009. `http://mtc.sri.com/Conficker/`.

[9] Arvind Seshadri, Mark Luk, Ning Qu, and Adrian Perrig. SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integrity for Commodity OSes. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles*, pages 335–350, 2007.

[10] Takahiro Shinagawa, Hideki Eiraku, Kouichi Tanimoto, Kazumasa Omote, Shoichi Hasegawa, Takashi Horie, Manabu Hirano, Kenichi Kourai, Yoshihiro Oyama, Eiji Kawai, Kenji Kono, Shigeru Chiba, Yasushi Shinjo, and Kazuhiko Kato. BitVisor: A Thin Hypervisor for Enforcing I/O Device Security. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2009)*, pages 121–130, 2009.

[11] Trend Micro. Deep Security. http://emea.trendmicro.com/emea/products/enterprise/deep-security/.

[12] Youhui Zhang, Yu Gu, Hongyi Wang, and Dongsheng Wang. Virtual-Machine-based Intrusion Detection on File-aware Block Level Storage. In *Proceedings of the 18th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD '06)*, pages 185–192, 2006.

[13] 小川 夏樹, 大山 恵弘. ADvisor: ゲスト OS の操作に連動した広告を表示するハイパバイザ. 情報処理学会研究報告「システムソフトウェアとオペレーティング・システム（OS）」, volume 2011-OS-118, 2011.

[14] 河崎 雄大, 大山 恵弘. VMM を用いたマルウェア検出システムのためのシグネチャデータ更新機能とメモリデータ検査機能. 情報処理学会研究報告「システムソフトウェアとオペレーティング・システム（OS）」, volume 2012-OS-122, 2012.