# IaaS 型クラウドにおける仮想マシンのセキュリティ課題

須崎有康　（独）産業技術総合研究所

**Amazon EC2 を代表とする IaaS（Infrastructure as a Service）型クラウドコンピューティングでは，不特定多数のユーザから複数 OS をホスティングするマルチテナント環境のために仮想マシンは必須の技術になっている．仮想マシンのセキュリティについてはアメリカ国立標準技術研究所（NIST：National Institute of Standards and Technology）から出された導入ガイドライン（SP 800-125 DRAFT Guide to Security for Full Virtualization Technologies）などで導入の注意点は指摘されているが，具体的な技術的詳細については解説されていない．本稿では技術的な問題点を掘り下げて IaaS 型クラウドコンピューティングでの仮想マシンセキュリティの課題と対処方法を仮想マシン内（VM Internal）と仮想マシン間（Cross VM）に分けて解説する．ここではクラウドを利用する際に契約する SLA（Service Level Agreement）の参考になることを念頭に置いている．**

## IaaS 型クラウドコンピューティングでの仮想マシン利用の変遷

まず，IaaS 型のクラウドコンピューティングで仮想マシンが利用されてきた変遷について簡単に説明する．図 -1 にレガシーシステムからの変遷の過程を示す．レガシーシステムではハードウェアの低価格化により，1 台の高性能・高価格サーバマシンですべてのサービスを提供する方式から各サービスを複数の低性能・低価格サーバマシンに分ける分散化（Distributed Computing）が進んだ．ここでは物理マシンとサービスが 1 対 1 に対応して物理マシン単位での管理が可能になったが，メモリや CPU の資源が融通できない問題があった．

このレガシーな環境は CPU の性能向上と仮想化支援命令が追加されたことにより，仮想化技術（Virtualization）を適用することで物理マシンごとに分かれていた複数のサービス（OS）を 1 台の物理マシン上の仮想マシンとして集約できるようになった（図 -1 中段）．これがサーバコンソリデーションと呼ばれる機能で，高性能化した CPU においては十分に複数の仮想マシン（サービス）を処理することができた．また，1 台の物理マシンの中の各資源（CPU，メモリ，ストレージ）を要求に応じて適切に分配するプロビジョニングが可能であり，1 つの OS では使いきれなかった資源を複数 OS で共有することで効率的に資源活用できるようになった．仮想マシンは常に資源を使いきるわけではないので，仮想マシンの資源の合計が物理マシンの資源を超えるオーバーコミットも可能になり，柔軟に管理できるようになった．

クラウドコンピューティングではさらに仮想化の技術を活用して管理の効率化を実現している（図 -1 下段）．仮想マシンではハードディスクをファイルと同等に扱えるため，一度インストールした OS のイメージを簡単にコピーできるようになった．このイメージは VMI（Virtual Machine Image）と呼ばれ，インストールの手間を取り除くことができるようになった．また，実行中の仮想マシンを他の物理マシンに移動するライブマイグレーションを使うことで複数の物理マシンに跨った負荷分散が可能になった．ライブマイグレーションは障害時やメンテナンス時に仮想マシン退避にも活用される．オーバー

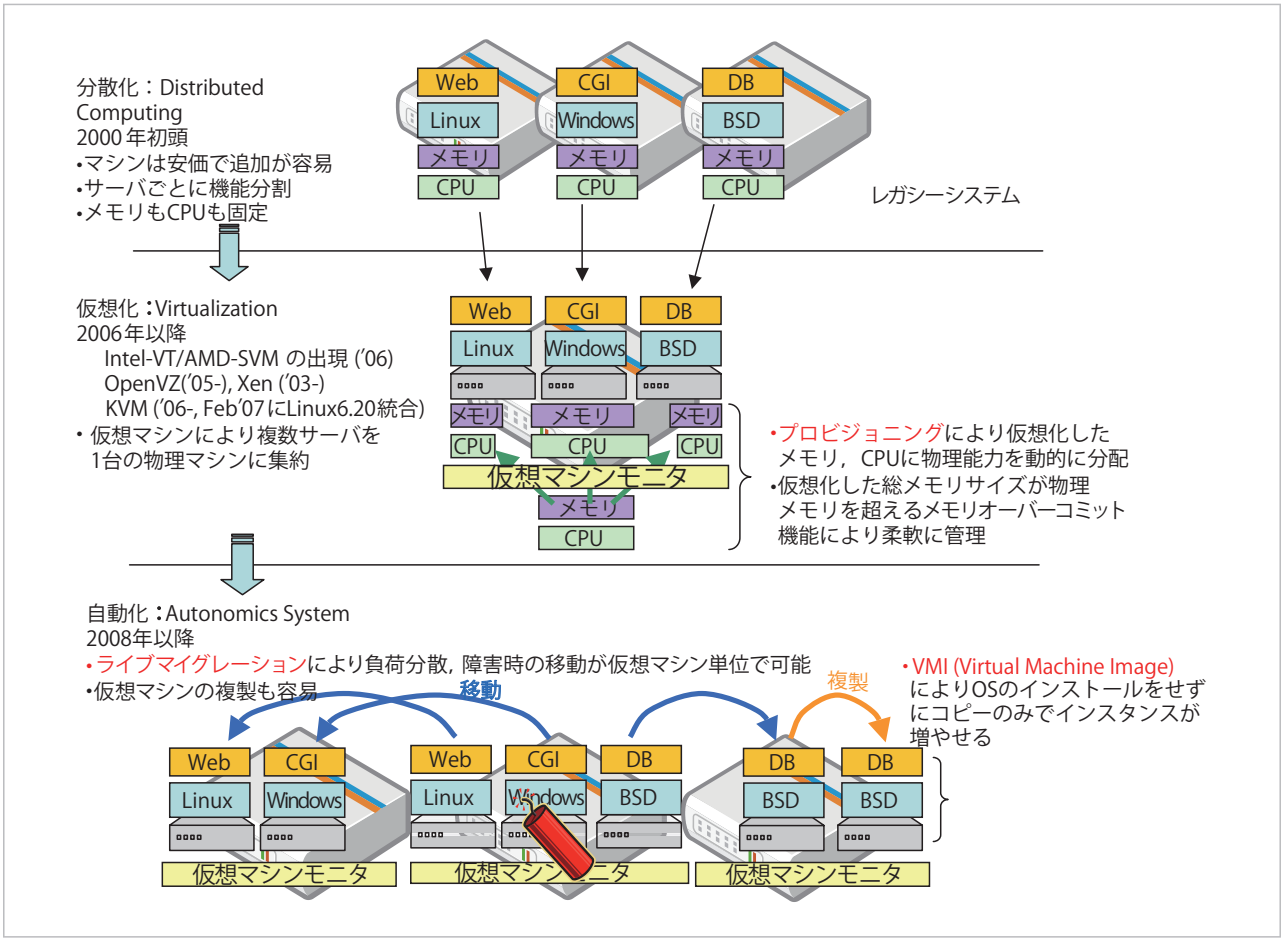


図 -1　仮想マシンによるクラウドコンピューティング環境の変遷．レガシーな分散化から仮想化，さらには自動化へ進化．

コミットした物理マシンで物理的資源が不足したときには仮想マシンが自動的にライブマイグレーションされるようになり，クラウドコンピューティングのデータセンタの管理の自動化（Automatic System）が行えるようになった．これらの機能により管理者の労力が軽減され，不特定多数の仮想マシンをホスティングするマルチテナントの大規模化が可能となった．

このように仮想マシンおよびそれから派生した技術によりクラウドコンピューティングが成り立っているが，いずれの技術もセキュリティの課題を内在している．それらの問題について以下の項目で解説していく．

## そもそも仮想マシンによってセキュリティが強化されるのか?

多くの仮想化技術開発者は仮想マシンモニタと管理 OS はゲスト OS より小さいため脆弱性も少なく，セキュリティを強化できると主張している．一般にバグの混入はコード量に比例することが示されているのでこの主張は正しく見える．しかし，仮想マシンモニタは物理デバイスの仮想化を行っており，通常のアプリケーションプログラムでは話題にならない物理デバイスの再現問題がある．特に多くの仮想マシンモニタではオーバーヘッドを隠すためのバッファリングなどの最適化が行われており，エラー処理やタイミングなど物理デバイスを完全にエミュレートできているわけではない [1]．

仮想マシンモニタはデバイスを再現するため，ゲ

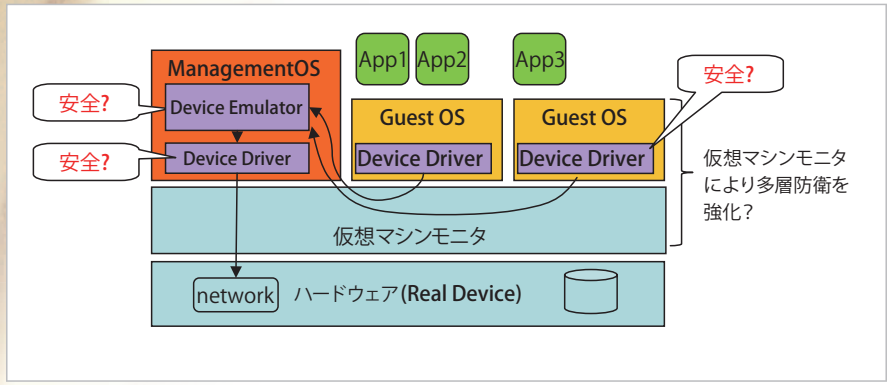


図 -2　仮想マシンモニタ自体のセキュリティ．ドライバの 2 重化による再現の困難性．

スト OS のドライバと管理 OS デバイスエミュレータ，実デバイスに対する管理 OS 内のドライバなど，ドライバが多重化されている（図 -2）．一般にドライバ開発は問題が多い．2009 年に OS 関連の学会で最も権威のある ACM SOSP（Symposium on Operating Systems Principles）で発表されて話題になった seL4[2] ではカーネルの全コード（8,700 行）を形式検証しているが，ブートローダやデバイスドライバなどハードウェアに関係する部分は形式的検証の対象外になっている．つまり，仮想マシンモニタでは形式検証が難しい部分のプログラミングを要求されているわけである．

また，仮想マシンモニタへの攻撃はすべてのゲスト OS の権限を乗っ取ることができるので被害も大きい．実際，アメリカ政府の支援を受けた非営利団体の脆弱性データベース CVE（Common Vulnerabilities and Exposures）においてもゲスト OS の権限昇格（VMWare と Xen）やゲスト OS 外のリソースアクセスの脆弱性（Xen）が報告されている．今後このような脆弱性の問題が見つからない保証はない．

変遷の章で述べたように仮想化は IaaS 型のクラウドコンピューティングでは必須の技術であるが，仮想化自体にも問題が少なくなく，常に安全性を確認する必要がある．以下，2 つの章を使って IaaS 型クラウドにおいて仮想マシン内（VM Internal）の攻撃／脆弱性と複数の仮想マシン間（Cross VM）の攻撃／脆弱性に分けて問題点と対処法を紹介していく．

## 仮想マシン内（VM Internal）の攻撃／脆弱性

単独の仮想マシン内（VM Internal）の攻撃／脆弱性はクラウド環境以外の仮想マシン利用でも気をつけければならないセキュリティの問題であるが，クラウド内で起こることにより他の仮想マシンや管理 OS に被害を及ぼす危険性がある．

### --- デバイスの仮想化は完全ではない ---

先に述べたデバイスの仮想化であるが，この部分が脆弱であることはすでに多く知られている．仮想マシンの仮想デバイスを過剰にアクセスすることや想定外の要求を行うことで，管理 OS 乗っ取りや悪意あるコードの挿入を行うことが可能である．具体的な仮想マシンの攻撃については Google の Tavis Ormandy 氏の報告書[3] に詳しく記載されている．ここでは，高頻度の読み書きやランダムデータ読み書きによるアクセスなど，想定外の処理を要求することでデバイスドライバのテストを行う I/O fuzzing の技術を悪用する（図 -3）．この攻撃により仮想デバイスで障害を起こし，ゲスト OS の管理者権限や管理 OS へのコード挿入が可能であることが示されている．具体的には CRASHME というツールを使ってビデオデバイスの bitblt（bit block transfer）heap overflow やネットワークカードの netsock heap overflow を起こすことが可能であったことが報告されている．また，元 Symantec の Peter Ferrie（現 MicroSoft）の報告書[4] においては仮想マシンが仮想

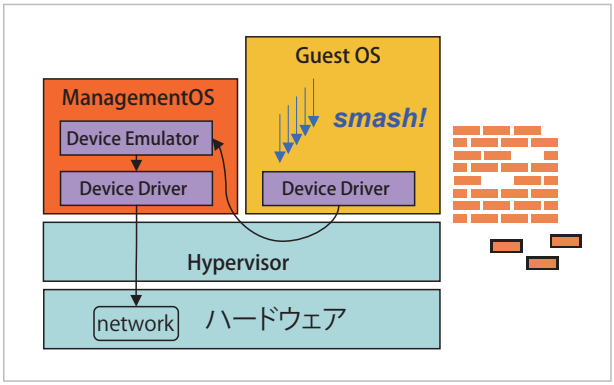


図 -3　I/O Fuzzing による仮想デバイスへの攻撃

化しきれないデバイスを攻撃する手法を紹介している．

このような仮想デバイスへの攻撃に対する基本的な対策は不要なデバイスを付けないことである．当たり前であるがレガシーなデバイスであるフロッピーなどを無効にする．クラウドコンピューティングでビデオを使うことがなければビデオカードの設定は無効にすべきである．

### --- 乱数がランダムにならない ---

現在の多くの仮想マシンモニタは仮想マシンの実行途中を保存するスナップショット機能がある．カーネルの更新などを試したい場合に更新前の状態に戻ることができる．この機能は便利であるが，そこにセキュリティの問題が含まれている．

通常の OS では乱数生成に疑似乱数を用いるが，スナップショットイメージを使うと前の疑似乱数生成を繰り返すことになり，同じ乱数列を生成してしまう．さらに問題なのは，疑似乱数の生成の元となる「種」の値は時計などの物理的要因から取られるが，仮想マシンモニタによってはスナップショット再開後に時間を同期しないため，同じ時間から再開され，種が同じになってしまう危険性がある．物理マシンなら明らかに時間が異なるが，仮想マシンの柔軟性から派生した新たな問題である．この危険性については 2010 年の NDSS の論文 [5] において暗号で同じ乱数が使われている危険性の詳細が述べられている．この対処として文献 5）の著者らは Hedged cryptography という暗号化レベルで仮想マシンの乱数生成の問題を意識して回避する手法を提案している．乱数や暗号に限らず，一時的であるはずのデータがスナップショットの再開により繰り返されしまうため，この機能は注意して使う必要がある．

### --- メモリエラーを引き金にする ---

クラウドコンピューティングは数万台を超える大規模なサーバ群から構成されるため，各デバイスの障害も半端ではない．2009 年の SIGMETRICS の論文 [6] において Google のサーバ群におけるメモリのエラーレートが報告された．ここでは通常考えられている以上に物理的なメモリエラーが起こることが示されている．Google サーバのメモリはエラー訂正機能である ECC（Error Correcting Code）を付けているが，それでもエラー訂正できない障害が発生した．特に最近 1 カ月以内に訂正可能なエラーが発生したメモリモジュールでは，訂正されないメモリエラーの確率が非常に高く，場合によっては 431 倍になることが報告されている．通常の PC ならば多くのデータがハードディスクに格納されるので深刻な問題は起こりにくいが，クラウドコンピューティングでは応答性を考えてメモリ上の処理が多く，このようなメモリエラーはサーバの障害となる．サーバが停止するのみならまだ良いのだが，このような物理メモリエラーを想定した攻撃も考慮する必要がある．

実際，2003 年の IEEE Symposium on Security and Privacy の論文 [7] においてメモリエラーを想定した攻撃が紹介されている．ここではメモリエラーが起こった際にランダムに制御が移ることを想定し，悪意のあるコードにジャンプする仕組みをメモリ内に敷き詰める．この攻撃はスプレー攻撃と呼ばれるものであり，何かの拍子にそこに行けば悪意のコードが実行されてしまう．論文では Java のバーチャルマシンを想定しているが，手法としてはクラウドコンピューティング内でも同一のことが起こり得る．

この対処としては SELinux が提供しているような各権限での強制アクセス制御を施すことである．乗

っ取られると被害が大きい管理者権限などは，何か障害があっても他に影響を与えないように設定しておくべきである．また，制御がデータ領域に飛んだ場合の対処としてはAMDのNXビットやIntel CPUのDXビットのようなデータ領域がコード実行に使われないデータ実行防止機能（DEP：Data Execution Prevention）も有効である．

### --- 仮想マシンのOSのメンテナンス ---

仮想マシンへのOSインストールはVMIの利用により簡単になった．クラウドコンピューティングによってはVMIの作成が簡単なメニューから選択するのみで可能になっている．しかしプライバシーの問題もあり，VMIのOSのセキュリティアップデートは各ユーザが行うことになっている．特に仮想マシンを長い間立ち上げずにOSのセキュリティアップデートが行われていないと無防備な状態で起動して攻撃される危険性が多い．

クラウドコンピューティング提供者および商用仮想マシンモニタのベンダでもこの問題を認識しており，VMware Update Manager（VUM）やMicrosoft Offline Virtual Machine Servicing Toolなどの VMIを直接扱うツールをリリースしている．これらのツールの使用についてはベンダのサービスとしてのみではなく，ユーザのプライバシーや既存のOSを使い続けたい意向など，要求に適した組合せを探す必要がある．

## 仮想マシン間（Cross VM）の攻撃／脆弱性

仮想マシン間（Cross VM）の攻撃／脆弱性は，複数の仮想マシンによるデバイス共有やライブマイグレーションによる物理マシン移動から派生する問題である．IaaS型のクラウド環境ではマルチテナント環境になっており，他の仮想マシンからの攻撃にも考慮しなければならない．

### --- 隔離技術でどこまで隔離するか ---

IaaS型のクラウドコンピューティングでは，不特定多数のユーザから多く仮想マシンを預かるマルチテナント環境である．この環境では仮想マシンが覗き見られないための隔離（isolation）技術が重要である．各仮想マシンは仮想マシンモニタにより論理的に隔離されているために他の仮想マシンを覗き見ることができないことになっているが，実際には物理デバイスを共有しているためにその動作から覗き見られる危険性がある．特に2009年のセキュリティで権威のある学会CCS（ACM Computer and Communications Security Conference）で発表された論文[8]は仮想マシンからキャッシュの動作を通して他の仮想マシンを覗き見るサイドチャネル攻撃で話題になった．これはCross VM Side Channel Attackと呼ばれる．

この攻撃ではマルチコアCPUでコアごとに仮想マシンが割り当てられ，同時に複数の仮想マシンがキャッシュを共有することを悪用している．まず，あるコアで実行する仮想マシン（攻撃側）が物理CPUキャッシュをアクセスし続ける．他のコアで実行する仮想マシン（被害側）からアクセスがあった場合に攻撃側のアクセスの遅れが出るため，そのパターンから被害側の仮想マシンの処理を推定する攻撃である（図-4）．

現在の多くのCPUのキャッシュはセットアソシアティブと呼ばれる手法で管理されている．この方式ではあるサイズで領域を分け，下位ビットでその領域（キャッシュライン）ごとにデータ管理する．キャッシュは通常多段（図中では2way）になっており，個々のキャッシュラインごとにLRU（Least Recently Used）により保持される．この環境でまず悪意のある仮想マシンは各キャッシュラインに連続にアクセスする．被害者の仮想マシンがあるキャッシュにアクセスすると悪意のある仮想マシンのキャッシュアクセスは遅くなるため，被害者がどのようにアクセスしているかを知ることで何の処理をしているかを推定する．この攻撃自体は2005年にHyper-Threadingの脆弱性として知られていた問題

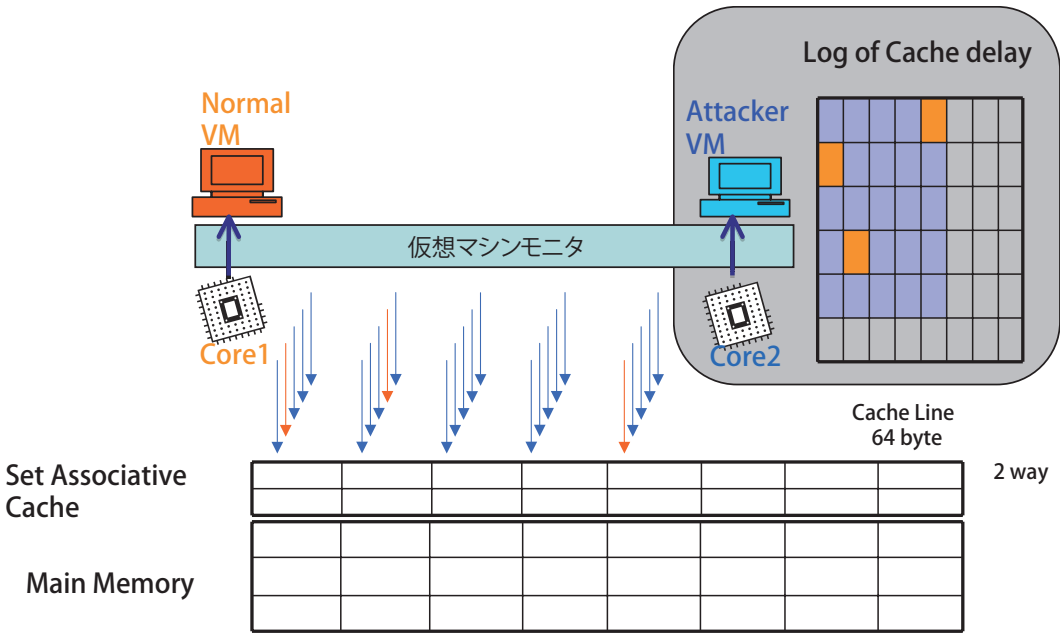


図 -4　Cross VM Side Channel Attack．キャッシュの動作から同じ CPU 上の仮想マシンの振舞いを推定する．

であるが，今回は実際に Amazon EC2 のクラウドコンピューティングで使われたことが報告されており，ある条件では暗号鍵の漏えいまで可能であることが報告されている．ただし，この攻撃は被害者が実行しているコードがあらかじめ分かっていることを前提としているので現状では実害がないが，このような攻撃があることを想定して対処しておく必要はある．

防御方法としてはいくつがあるが，ポイントは「同時に物理的に共有するキャッシュを使わない」ようにできればよい．物理的には CPU コアごとに異なる物理キャッシュを持つような構成が作れればこの攻撃は回避されるが，現在提供されている CPU を改良するのは困難である．仮想マシンモニタでも対処が可能であり，「利害関係のある仮想マシンはキャッシュを共有する CPU コアで同時に実行しない」ポリシーを記述できればよい．Xen ではこの機能を XSM（Xen Security Module）の sHype を通して設定できる．この機能を提供するクラウドコンピューティングは少ないと思うが，クリティカルな処理を要求する側は物理的に分離する方法も含めて契約時の SLA（Service Level Agreement）で検討すべきである．

Cross VM Side Channel Attack は物理的キャッシュの振舞いを通した覗き見であったが，メモリの覗き見は仮想マシンモニタを乗っ取れれば容易に行うことができる．幸いにもこれに対する仮想マシンのメモリを暗号化する方式（Overshadow など）がすでに提案されている．

メモリの暗号化は有用であるが，最近はメモリ内で同一内容のブロックがある場合には共有してメモリ使用量を抑制する「重複除外（Deduplication）」技術が出ており，メモリの隔離を難しくしている．VMWare ESX では Content-Based Page Sharing があり，Xen においては Differential Engine, Satori がある．KVM においては Linux kernel 2.6.32 から採用された KSM（Kernel Samepage Merging）を管理 OS で有効にすることで，仮想マシンの重複除外が可能である．いずれも仮想マシン上の OS に意識させることなく，メモリの集約を可能にする．重複除外技術は共有を促進する技術であり，覗き見などの攻撃を誘発する危険性がある．

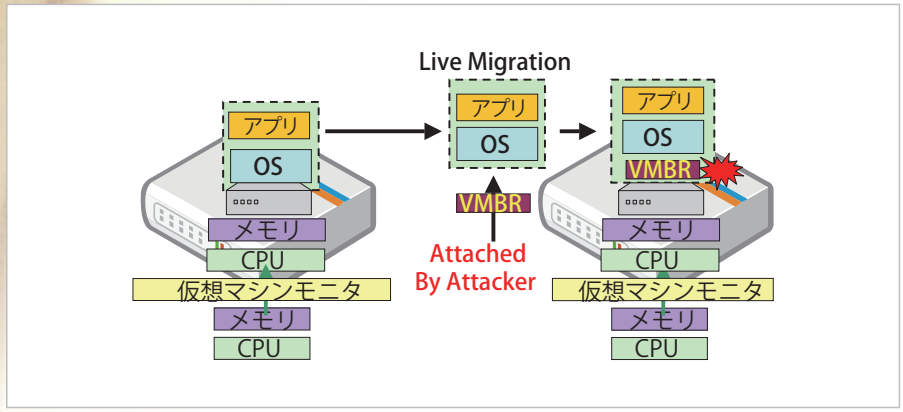


図-5　ライブマイグレーション時のルートキット挿入攻撃

### --- 仮想デバイスがセキュリティの邪魔をする ---

多くの仮想マシンモニタでは，同一物理マシン上の仮想マシン間を仮想ネットワークで繋ぎ，高速な通信経路として利用できる．物理ネットワークを通さずに仮想マシンモニタ内においてすべてメモリ上で処理される．これは効率化のためによいことであるが，全通信プロトコルが仮想マシンモニタ内に閉じ込められ，表に出てこない．つまり，ネットワークトラフィックが他から見えなくなり，フィルタリングできなくなってしまう．仮想マシンモニタ上の特定仮想マシンへのブルートフォースアタックも検出できず，かつ効率的に行われてしまう．

この対処として仮想マシンモニタが仮想マシンファイアウォールを提供するようになってきている．たとえば，日本IBMではVMwareに特化したセキュリティ対策ソフト「Virtual Server Security for VMware（VSS）」をリリースした．ここではVMwareがセキュリティ対策ソフトベンダ向けに公開するAPIのVMsafeを使って，各仮想マシンの通信内容をチェックする．これ以外にも仮想マシンファイアウォールや仮想マンモニタ内にIDS（Intrusion Detection System）を含めたものがあるので必要に応じて機能を確認すべきである．

### --- ライブマイグレーション時の脅威 ---

ライブマイグレーションは任意の仮想マシンを効率よく管理する上で必須の技術であるが，ここにもセキュリティの問題が潜んでいる．ライブマイグレーションでメモリのイメージを転送する際にルートキットによる感染の危険性が2008年のBlackHat DCの論文[9]で報告されている．ここでは，2006 IEEE Symposium on Security and Privacyの論文[10]で紹介されたVMBR（Virtual Machine Based Rootkit）をライブマイグレーションする際に挿入している（図-5）．VMBRは仮想マシンモニタを利用したルートキットであり，攻撃対象のOSから仮想マシンモニタを隠蔽してしまうため，検出が難しい．

この問題はライブマイグレーションに移動前後のイメージの完全性を検証する仕組みがないことに起因する．また，通信路の暗号化により外部からのVMBR挿入攻撃は防げるが，内部の仮想マシンから通信を覗き見られていた場合には対処できない．この対処としてはライブマイグレーションの前後で転送するメモリイメージのハッシュ値などの完全性検証を導入する必要性がある．

### --- プロバイダ管理者からの覗き見や情報漏洩 ---

仮想マシンでは管理者権限があれば仮想化されたデバイスの状態を覗き見ることができる．特に最近の仮想マシンモニタではセキュリティ向上を目的とした仮想デバイスにアクセスする仕組み（VMWareのVMSafeやXenのXenAccessなど）が付加されている．これらのインタフェースはカーネルに対するルートキットのようなOS自体からは確認できない攻撃に対して，仮想デバイス状態と照らし合わせて検出することを目的としているが，逆に悪用される

恐れもある．悪用を防ぐ対策としては仮想デバイスのアクセスできる機能はその権限分離とアクセス制御を行い，必要なこと以外で利用することを禁止する必要がある．

## おわりに

仮想マシンは便利であるが，上記のように仮想マシン内（VM Internal）および仮想マシン間（Cross VM）で思いがけないセキュリティの問題が含んでいる．特にクラウドコンピューティングでは何万という仮想マシンがデータセンタで実行されているため，個人のパソコンで使っていたときには見逃されていた問題が明らかになってきている．現在，クラウドの普及に向けてガイドラインが多く発行されているが，セキュリティ関連の学会や BlackHat などで発表されている新しい問題に追いついていないのが現状である．少なくともクラウドプロバイダとの契約で SLA を結ぶ際に本稿で挙げた問題を考慮しているか確認してほしい．

**参考文献**

1) Garfinkel, T. et al. : Compatibility is Not Transparency : VMM Detection Myths and Realities, HotOS'07.
2) Klein, G. et al. : seL4 : Formal Verification of an OS Kernel, SOSP '09.
3) Ormandy, T. : An Empirical Study into the Security Exposure to Hosts of Hostile Virtual Environments, http://taviso.decsystem.org/virtsec.pdf
4) Ferrie, P. : Attacks on Virtual Machine Emulators, SYMANTEC ADVANCED THREAT RESEARCH, http://www.symantec.com/avcenter/reference/Virtual_Machine_Threats.pdf
5) Ristenpart, T. and Yilek, S. : When Good Randomness Goes Bad : Virtual Machine Reset Vulnerabilities and Hedging Deployed Cryptography, NDSS'09.
6) Schroeder, B. : DRAM Errors in the Wild: A Large-scale Field Study, SIGMETRICS/Performance'09,
7) Govindavajhala, S. and Appel, A. W. : Using Memory Errors to Attack a Virtual Machine, IEEE Symposium on Security and Privacy 2003.
8) Ristenpart, T. et al. : Hey, You, Get Off of My Cloud : Exploring Information Leakage in Third-Party Compute Clouds, CCS'09.
9) Oberheide, J. : Exploiting Live Virtual Machine Migration, BlackHat DC 2008.
10) King, S. et al. : SubVirt : Implementing Malware with Virtual Machines, IEEE Symposium on Security and Privacy 2006.

（平成 22 年 8 月 31 日受付）

**須崎有康**（正会員）k.suzaki@aist.go.jp

1991 年に東京農工大学大学院中退．同年より電子技術総合研究所．2001 年に改組を経て（独）産業技術研究所情報セキュリティ研究センター主任研究員．日本クラウドセキュリティアライアンス（CSA Japan Chapter）ボードメンバ．情報理工学博士（東京大学）．