クラウドの内部攻撃者に対する安全なリモートVM監視機構

重田 一樹† 光来 健一†

† 九州工業大学

あらまし IaaS 型クラウドにおいて VM への攻撃を安全に検知するために，侵入検知システム（IDS）のオフロードが提案されている．しかし，信頼できないクラウド上で安全に IDS オフロードを行うのは難しい．クラウドの内部攻撃者によって IDS を無効化されたり，VM 内の機密情報にアクセスされたりする恐れがあるためである．本稿では，IDS をクラウド外部の信頼できるホストにオフロードする *IDS* **リモートオフロード**を提案する．IDS をクラウド外部で動作させることで，クラウドの内部攻撃者からの攻撃を防ぐことができ，情報漏洩を防ぐための VM の安全な実行機構も共存させることができる．我々は IDS リモートオフロードを実現するシステム *RemoteTrans* を開発し，既存の IDS を効率よく動作させられることを確認した．

Secure Remote Monitoring of Virtural Machines against Insiders in Clouds

Kazuki Juda† Kenichi Kourai†

†Kyushu Institute of Technology

Abstract To securely detect attacks against virtual machines (VMs) in IaaS clouds, offloading intrusion detection systems (IDSes) has been proposed. However, it is difficult to offload IDSes securely to untrusted clouds. This is because insiders in clouds can disable IDSes and access sensitive information in VMs. This paper proposes *IDS remote offloading*, which offloads IDSes to trusted hosts outside clouds. Executing IDSes out of clouds can prevent attacks from insiders and coexist with VM's secure execution mechanisms for preventing information leakage. We have developed *RemoteTrans* to achieve IDS remote offloading and confirmed that existing IDSes could be executed efficiently.

1 はじめに

IaaS 型クラウドの普及により，ユーザは自身のサーバをクラウド上の仮想マシン（VM）で動作させることが多くなってきた．クラウド上でサーバを動作させることによって，必要に応じて台数を増減させることができ，コストを削減することができる．しかし，クラウドはユーザが所属する組織によって管理されているわけではないため，悪意のあるクラウドの内部攻撃者から攻撃を受ける危険性がある．このリスクを軽減するために，ユーザ VM をクラウド上で安全に実行できるようにする機構が提案されている [1] [2] [3]．これらの機構では，クラウド管理者が VM のメモリを参照する際に暗号化を行ったりアクセスを制限したりすることで，管理者への情報漏えいを防ぐ．

一方，外部攻撃者からの VM に対する攻撃への対策として，侵入検知システム（IDS）を用いて VM を監視することも必要である．クラウ

ドサービスとしてIDSを提供するために，VMを用いたIDSオフロード手法が提案されている [4]．この手法では，VMの外にIDSをオフロードし，安全にVM内のシステムの監視を行う．しかし，信頼できないクラウド上で安全にIDSオフロードを行うのは難しい．第一に，クラウド上にオフロードされたIDSは内部攻撃者によって無効化される恐れがある．第二に，オフロードされたIDSはVMのメモリを解析する必要があるため，VMの安全な実行機構と共存させることができない．

本稿では，IDSをクラウド外部の信頼できるホストにオフロードする***IDSリモートオフロード***を提案する．この手法を用いることで，IDSはネットワーク経由で安全にクラウド上のVMを監視することができ，クラウド管理者からの攻撃を防ぐことができる．また，VMの安全な実行機構と共存させることもできる．IDSリモートオフロードを実現するシステムである*RemoteTrans*は，クラウド内の仮想マシンモニタ（VMM）経由でVMの内部情報を取得する．信頼できないクラウド内のVMMはリモートアテステーションによって信頼する．

我々はRemoteTransをXen 4.1.3 [5]に実装した．RemoteTransはメモリ監視およびディスク監視をサポートしており，リモートから安全にメモリ監視を行えるようにリクエストとレスポンスの整合性チェックおよびリプレイ攻撃対策を行っている．また，既存のIDSをオフロードするための実行環境であるVM Shadow [6]をRemoteTransに対応させた．実験の結果，VM Shadowをリモートホスト上に構築するオーバヘッドは60%程度であった。また，chkrootkitおよびTripwireがリモートホスト上で効率よく動作することを確認した．

以下，2章でIaaS型クラウドでIDSオフロードを行う場合の問題点について述べ，3章ではRemoteTransについて述べる．4章で実装の詳細について述べ，5章では実験について述べる．6章で関連研究について述べ，7章で本稿をまとめる．

2　クラウドでのIDSオフロード

クラウドの内部に関する情報はほとんど公開されていないため，すべてのクラウド管理者が信頼できるとは限らない．そのため，管理者に起因する問題により，クラウド上のユーザのVM（ユーザVM）から情報が漏えいする危険がある．特に，クラウド管理者に悪意があった場合，クラウド内部からVMに対して容易に攻撃を行うことができる．例えば，ユーザVMのメモリを盗み見ることで機密情報を取得することも可能である．

このようなクラウドの内部攻撃者への対策として，クラウド上のVMを安全に実行する機構が提案されている．クラウド管理者は管理VMと呼ばれる特権を持ったVM上で管理を行うことが多い．CloudVisor [1]やVMCrypt [2]などでは，クラウド管理者が管理VMからVMのメモリを参照する際に暗号化を行うことで情報漏えいを防ぐ．また，SSC [3]ではクラウド管理者であってもユーザVMへのアクセスを制限される．

その一方で，外部攻撃者からのクラウドに対する攻撃への対策を行うことも必要である．攻撃者はクラウドを攻撃対象とすることで，サーバを効率よく見つけ出して攻撃できるため，クラウドに集約された多くのVMは攻撃の対象になりやすい．そのため，IDSによる監視がますます重要になる．しかし，クラウド上のVMについてはユーザ自身がシステムを構築するため，必ずしもIDSが導入されるとは限らず，導入されたとしても十分にセキュアな設定が行われるとは限らない．また，VM内のIDSは侵入と同時に無効化されてしまう危険もある．

そこで，IDSを安全に動作させるために，VMを用いたIDSオフロードが提案されている [4]．この手法では，ユーザVMが動作しているホスト上の管理VMなどにIDSをオフロードし，ユーザVMの外側から監視を行う．IDSオフロードを用いることで，ユーザVMに侵入されたとしてもIDSを無効化されないため，侵入を検知することが可能になる．オフロードされたIDSは，VMイントロスペクション [4]と呼ばれる手法を用いて，VMのメモリ上のカーネル

データなどを監視する．

しかし，クラウド上で IDS オフロードを行う場合には二つの問題が生じる．第一に，オフロードされた IDS はクラウドの内部攻撃者によって容易に無効化されてしまう．クラウド管理者は管理 VM にオフロードされた IDS を停止させたり，設定を改ざんしたりすることができる．その後でユーザ VM に侵入されると攻撃を検知することができない．IDS の無効化を防ぐために，SSC のようにユーザだけがアクセスできる VM に IDS をオフロードする手法も提案されている．しかし，その VM 内のシステムに脆弱性があると不正ログインされる危険性がある．

第二に，VM の安全な実行機構と共存させることができない．オフロードされた IDS は外からユーザ VM の内部状態を参照する必要がある．その際に，ユーザ VM のメモリが暗号化されたり，アクセスが制限されたりしていると，IDS を正常に動作させることができない．一方，VM の安全な実行機構を用いないと，オフロードされた IDS 以外のソフトウェアであってもユーザ VM の内部状態を参照でき，情報漏えいを防ぐことができない．

3　IDS リモートオフロード

本稿では，IDS をクラウド外部の信頼できるホストにオフロードする IDS リモートオフロードを提案する．IDS をクラウド外部で動作させることにより，クラウド管理者によって無効化されるのを防ぐことができる．また，IDS をオフロードしたホストにのみユーザ VM のアクセスを許可することで，VM の安全な実行機構と共存させることができる．

3.1　脅威モデル

本稿では外部からの攻撃者や悪意のあるクラウド管理者によって管理 VM が悪用されることを想定している．IaaS プロバイダ自体は信頼し，VM の下で動作する VMM やハードウェアを管理する少数の管理者も信頼する．しかし，管理

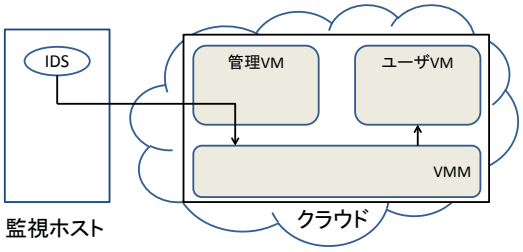

図 1: RemoteTrans におけるユーザ VM の監視

VM でユーザ VM を日常的に管理している一般のシステム管理者は信頼しない．また，VMM に脆弱性はないものとし，ハードウェアに物理的にアクセスする攻撃は想定しない．IDS をオフロードするユーザのホストは正しく管理されているものとし，攻撃を受けることは想定しない．

3.2　RemoteTrans

IDS リモートオフロードを実現するシステムである RemoteTrans の構成は図 1 のようになる．オフロードされた IDS は**監視ホスト**と呼ばれる信頼できるホスト上で動作する．監視ホストとしては，ユーザのプライベートクラウド上の VM を用いることが考えられる．監視ホスト上の IDS はクラウド内の VMM を経由してユーザ VM の監視を行う．IDS が VMM に監視リクエストを送ると，VMM がユーザ VM のメモリに直接アクセスし，取得したメモリデータをレスポンスとして IDS に返す．

RemoteTrans では，信頼できないクラウド内で改ざんされていない VMM が動作していることを保証するために，リモートアテステーションを用いる．起動時に VMM のハッシュ値を計算し，クラウド外部の信頼できる検証サーバに署名付きで送信する．ハッシュ値の計算は TPM を用いて行うことで，改ざんを防ぐ．検証サーバは署名の妥当性を確認してから，ハッシュ値を検証して VMM の完全性をチェックする．IDS は検証サーバに問い合わせることで正しい VMM が動作しているかどうかを確認することができる．起動時に正しい VMM が動作していることが確認できれば，VMM のメモリ保護機能により実行時の VMM の改ざんも防ぐことができる．

IDS と VMM の通信は VMM の機能を最小限に抑えるために管理 VM 経由で行われる．そのため，管理 VM までの通信路を暗号化したとしても，管理 VM 上の内部攻撃者によってリクエストやレスポンスが改ざんされたり盗聴されたりする危険性がある．これらを改ざんすることができれば，IDS が参照しようとしているデータとは異なるデータを返させることができてしまう．例えば，プロセスリストをたどる際に，次のプロセスの情報を取得するリクエストを次の次のプロセスの情報を取得するリクエストに書き換えられると，悪意のあるプロセスを隠されてしまう．また，レスポンスに含まれるユーザ VM のメモリデータから機密情報が漏れる恐れもある．

そこで，RemoteTrans では，リクエストおよびレスポンスが改ざんされていないことを保証するために整合性チェックを行う．VMM がリクエストとレスポンスのメッセージ認証コード (MAC) を計算してレスポンスとともに返すと，IDS 側でそれを検証する．クラウドの内部攻撃者は MAC を正しく計算することができないため，改ざんを検出できる．また，盗聴を防ぐために，VM の安全な実行機構により暗号化されたメモリ情報をレスポンスとして返し，IDS 側で復号する．

IDS と VMM は，MAC を計算するときに用いる暗号鍵を安全に共有する．IDS は実行開始時に，信頼できる鍵サーバから VMM の公開鍵を取得し，生成したセッション鍵を暗号化して VMM に送る．VMM は受け取ったセッション鍵を自身の秘密鍵で復号し，指定されたユーザ VM に対応づける．セッション鍵の送信も管理 VM を経由して行われるが，公開鍵暗号により管理 VM がセッション鍵を復号することはできない．

4　実装

我々は IDS リモートオフロードを実現するシステム RemoteTrans を Xen 4.1.3 に実装した．RemoteTrans は図 2 のように，監視ホストで動作するランタイム，管理 VM で動作するサーバ，VMM で動作するモジュールからなる．

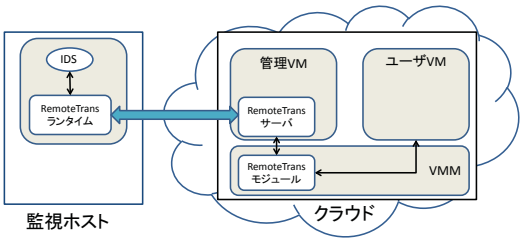

図 2: RemoteTrans のシステム構成

4.1　メモリ監視

監視ホストの IDS はユーザ VM のメモリデータを参照しようとした時，そのデータのアドレスとデータサイズからなるリクエストを RemoteTrans ランタイムに送る．RemoteTrans ランタイムがリクエストをネットワーク経由で RemoteTrans サーバに送ると，VMM 内の RemoteTrans モジュールが呼び出される．RemoteTrans モジュールは，リクエストされたアドレスにあるデータを指定されたサイズ分だけユーザ VM のメモリから取得し，RemoteTrans サーバに返す．このデータは，レスポンスとして RemoteTrans ランタイムに送られ，IDS に渡される．現在の実装では，メモリデータの暗号化はまだ行っていない．

RemoteTrans ランタイムは取得したメモリデータを高速化のためにキャッシュする．IDS がユーザ VM 内の複雑なカーネルデータを解析する場合，毎回ネットワーク経由でユーザ VM のデータを取得すると通信のオーバヘッドが大きくなってしまう．そこで，RemoteTrans ランタイムが一度取得したメモリデータは再度取得しないようにし，IDS が必要とした時にはキャッシュ上のデータを返す．さらに通信回数を削減するために，4KB のページ単位でまとめてメモリデータを取得する．このようにメモリデータをキャッシュすることで，最新のメモリデータを監視できないこともありえるが，定期的に動作する IDS の場合，多少古いデータであっても問題ないと考えられる．

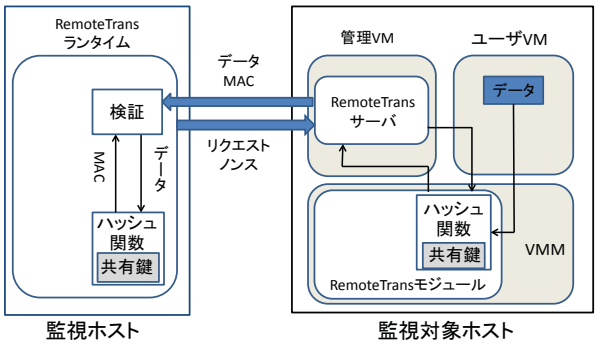

図 3: 整合性チェック

4.2 メモリ監視の整合性チェック

RemoteTrans ではリクエストおよびレスポンスが改ざんされていないことを保証するために，図 3 のように整合性チェックを行う．VMM 内の RemoteTrans モジュールが RemoteTrans ランタイムからのリクエストを受け取ると，リクエストに含まれるアドレスとデータサイズ，およびユーザ VM から取得したメモリデータから MAC を計算する．リプレイ攻撃を防ぐためにノンスと呼ばれる乱数をリクエストとともに送り，MAC の計算にはノンスも含める．

RemoteTrans サーバがレスポンスとともに MAC を返すと，RemoteTrans ランタイムでも保存しておいたアドレスとデータサイズ，ノンスおよび，レスポンスに含まれるメモリデータから MAC を計算する．受信した MAC と比較を行い，MAC の値が一致しなければ，リクエストかレスポンスのどちらかが改ざんされたとみなす．このように RemoteTrans ランタイムでリクエストとレスポンスの整合性を一括検査するため，リプレイ攻撃を検出するためにシーケンス番号や時刻を管理する必要がない．

4.3 ディスク監視

ユーザ VM のディスクを監視できるようにするために，ネットワーク・ブロックデバイス (NBD) を用いる．NBD は，リモートのディスクを仮想的なブロックデバイスとして扱うことができるツールである．RemoteTrans ランタイムは NBD を用いて，ユーザ VM のディスクを監視ホスト上にマウントする．ユーザ VM のディスクは管理 VM によって管理されており，VMM からアクセスするのは難しいため，NBD サーバは管理 VM で動作させる．

内部攻撃者によるユーザ VM のディスクの改ざんを防ぐために，ユーザ VM の中で Linux の dm-crypt を用いてディスクを暗号化する．管理 VM 上の攻撃者にディスクを改ざんされると，IDS が正常に動作していても侵入を検知できなくなるためである．RemoteTrans ランタイムでは，暗号化されたディスクを復号してから IDS に参照させる．この手法では改ざんそのものを検出することはできないが，暗号化を行うことで，攻撃者は意図したようにディスクを改ざんできなくなる．

4.4 VM Shadow の移植

我々は，既存の IDS をオフロードするための実行環境 VM Shadow [6] を RemoteTrans 上に移植した．VM Shadow はシステムコール・エミュレータと Shadow ファイルシステムで構成される．システムコール・エミュレータによって VM Shadow の中で動作するプロセスが発行したシステムコールをエミュレートし，ユーザ VM のカーネル内の情報を返す．Shadow ファイルシステムはユーザ VM 内で使われているものと同じファイルシステムを提供する．特に，特殊なファイルシステムとして Shadow proc ファイルシステムを提供する．このファイルシステムはユーザ VM 内のカーネルの現在の状態や実行中のプロセス情報などを提供する．例えば，ps コマンドや netstat コマンドは proc ファイルシステムを参照して実行される．

5 実験

RemoteTrans を用いた VM 監視のセキュリティおよび性能を調べる実験を行った．ユーザ VM を動作させる監視対象ホストには，Intel Xeon E3-1290 の CPU，16GB のメモリを搭載したマシンを使用し，VMM として Xen 4.1.3 を動作させた．RemoteTrans サーバを動作させるドメ

イン 0 には 8 つの仮想 CPU と 12GB のメモリを割り当て，Linux 3.2.0 を動作させた．ユーザ VM には 1 つの仮想 CPU と 4GB のメモリを割り当て，Linux 2.6.27.35 を動作させた．IDS と RemoteTrans ランタイムを動作させる監視ホストには，Intel Core i7-870 の CPU，8GB のメモリを搭載したマシンを使用し，OS には Linux 3.2.0 を用いた．これらのホストはギガビットイーサネット・スイッチで接続した．

5.1 改ざんの検知

RemoteTrans サーバにおけるメモリ監視のリクエストとレスポンスの改ざんが検知できるかどうかを確認する実験を行った．具体的には，特定のアドレスがリクエストに含まれる時にそのアドレスとデータサイズを改ざんしたり，レスポンスに含まれるデータを改ざんしたりした．実験の結果，RemoteTrans ランタイムにおいて，MAC が一致せず，リクエストまたはレスポンスが改ざんされていることを検知することができた．

また，メモリ監視においてリプレイ攻撃を検知できることを確認する実験を行った．そのために、RemoteTrans サーバで，以前に返されたレスポンスを保存しておき，同じデータへのリクエストを受け取った時にそれを RemoteTrans ランタイムに返すようにした．実験の結果，RemoteTrans ランタイムにおいて MAC が一致せず，リプレイ攻撃においてもレスポンスの改ざんを検知することができた．

5.2 マイクロベンチマーク

RemoteTrans におけるメモリ監視性能を調べるために，ユーザ VM のメモリを読み込むベンチマークを行った．このベンチマークでは，VM のメモリをページ単位でアクセスして 1MB のデータを取得するのにかかる時間を測定した．この実験は監視ホスト上，管理 VM 上，およびユーザ VM 内で行った．実験結果を図 4 に示す．この結果より，監視ホストからの読み込みには管理 VM からの 45 倍の時間がかかることがわかった．これは，通信によるオーバヘッドおよび MAC 検証のオーバヘッドである．メモリデータの暗号化を行うとさらにオーバヘッドは大きくなると考えられる．

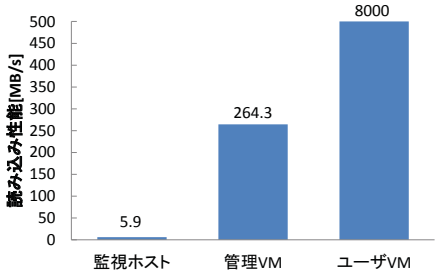

図 4: メモリ読み込み性能

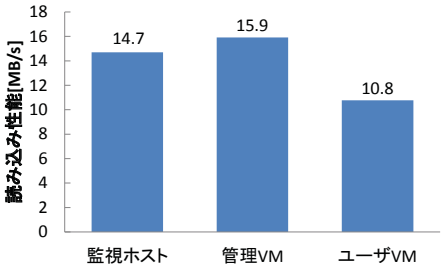

図 5: ディスク読み込み性能

次に，IOZone を用いてディスクを読み込む性能を測定した．この実験では，VM のディスク上の 1GB のファイルを読み込む性能を測定した．どの実験環境でも dm-crypt を用いた．実験結果を図 5 に示す．この結果より，監視ホストでの読み込み性能は管理 VM より少し低いが，ユーザ VM よりは高いことがわかった．ユーザ VM は仮想化によるオーバヘッドが原因で読み込み性能が低いと考えられる．また，監視ホストは通信のオーバヘッドにより管理 VM より読み込み性能が低いと考えられる．

5.3 VM Shadow 構築時間

VM Shadow の構築にかかる時間を従来システムと RemoteTrans とで比較した．実験結果を表 1 に示す．RemoteTrans では，従来システムの 1.6 倍程度の時間がかかっていることがわかる．RemoteTrans では 1635 回の通信を行い、6.4MB のデータを取得していた．RemoteTrans

表 1: VM Shadow 構築時間（秒）

	実行時間
従来システム	1.1
RemoteTrans	1.8

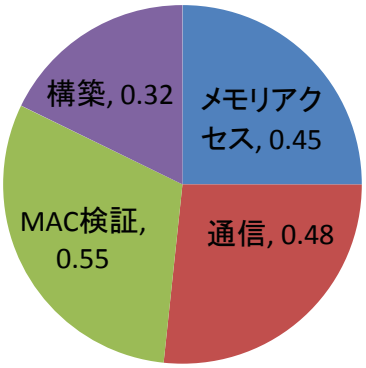

図 6: VM Shadow 構築時間の内訳 (秒)

における VM Shadow の構築時間の内訳を図 6 に示す．この結果より，RemoteTrans サーバとの通信や VMM を介したメモリアクセスよりも，MAC の検証に時間がかかっていることがわかった．実運用環境では監視ホストとクラウド間の通信により時間がかかるため，通信がボトルネックになると考えられる．

5.4 chkrootkit

chkrootkit を VM Shadow を用いてリモートの監視ホストおよびクラウド内の管理 VM で実行した場合と，ユーザ VM 内で実行した場合について実行時間を測定した．chkrootkit はルートキットを検知する IDS である．図 7 の実験結果より，監視ホストでの実行時間は管理 VM とほぼ同じであることがわかった．これらはユーザ VM 内での実行時間より短いが，VM Shadow では一部の情報が正しく取得できていないことも原因の一つと考えられる．

5.5 Tripwire

ファイルシステムの整合性を検査する IDS である Tripwire を用いてユーザ VM のディスク監視にかかる時間を測定した．5.4 節と同様に，

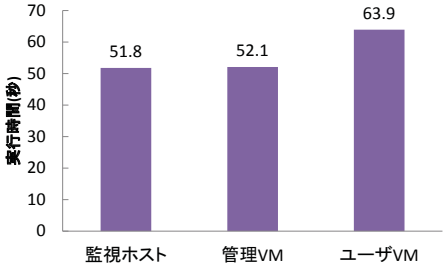

図 7: chkrootkit の実行時間

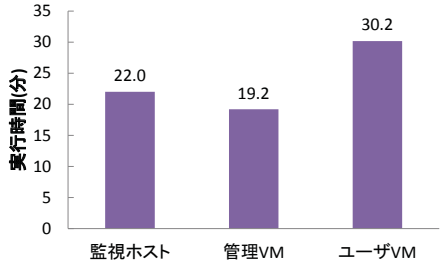

図 8: Tripwire の実行時間

監視ホストと管理 VM では VM Shadow を用い，ユーザ VM 内では直接実行した．実験結果を図 8 に示す．監視ホストでの実行時間は管理 VM より少し長いが，これはネットワーク経由でディスクを監視しているためである．それでもユーザ VM で実行したときと比べると，ディスク仮想化のオーバヘッドがない分だけ高速であることがわかった．

6 関連研究

CloudVisor [1] は VMM の下にセキュリティモニタを導入することで，VMM も含めて信頼できないクラウドの中でも安全に VM を動作させることができる．VM が他の VM を監視する機能は提供されておらず，管理 VM が VM のメモリを参照する時には暗号化されるため，IDS をオフロードすることはできない．

VMCrypt [2] は VM のメモリやレジスタから管理 VM へ情報が漏洩することを防ぐシステムである．管理 VM がユーザ VM のメモリをマップしようとすると，VMM がそのメモリ内容を暗号化する．これらの機構を用いるとメモリが暗号化されるため，従来の IDS オフロードでは

監視を行うことができなくなる．RemoteTransを用いれば信頼できる監視ホストでメモリデータを復号することで，オフロードしたIDSの実行が可能になる．

Self-Service Cloud（SSC）[3]は，クラウドのユーザだけに自身のVMを管理する権限を与え，クラウドの管理者からの干渉を防ぐ．ユーザはサービスドメインと呼ばれるVMを安全に起動し，他のVMを監視することができる．クラウドの管理者がサービスドメインの中のIDSを停止したり改ざんしたりすることはできない．しかし，サービスドメイン内のシステムに脆弱性があると攻撃を受ける可能性がある．

HyperCheck [7]はCPUの安全なモードであるSMMを使ってVMMのメモリをリモートホストに送り，完全性のチェックを行うシステムである．メモリを安全にリモートホストに送ることができる点でRemoteTransに似ている．しかし，SMM上のコードはリモートホストからのリクエストを受け取ることができないため，定期的にメモリ全体を送信する必要がある．そのため，データ通信量が多くなるという問題がある．

7　まとめ

本稿では，IDSをクラウド外部の信頼できるホストにオフロードするIDSリモートオフロードを提案した．この手法により，IDSはネットワーク経由で安全にクラウド内のユーザVMを監視でき，クラウドの内部攻撃者によるIDSの無効化を防ぐことができる．また，VMの安全な実行機構と共存させることもできる．我々はIDSリモートオフロードを実現するシステムRemoteTransを開発し，既存のIDSを動作させるための実行環境VM Shadowを提供できるようにした．実験の結果，chkrootkitとTripwireを従来システムとほぼ同等の性能で動作させられることを確認した．

今後の課題はユーザVMのネットワーク監視である．また，RemoteTransをVMの安全な実行機構と統合することも課題の一つである．

参考文献

[1] Zhang, F., Chen, J., Chen, H. and Zang, B.: CloudVisor: Retrofitting Protection of Virtual Machines in Multi-Tenant Cloud with Nested Virtualization, *Proc. Symp. Operating Systems Principles*, pp. 203–216 (2011).

[2] Tadokoro, H., Kourai, K. and Chiba, S.: Preventing Information Leakage from Virtual Machines' Memory in IaaS Clouds, *IPSJ Trans. Advanced Computing Systems*, Vol. 5, No. 4, pp. 101–111 (2012).

[3] Butt, S., Lagar-Cavilla, H. A., Srivastava, A. and Ganapathy, V.: Self-Service Cloud Computing, *Proc. Conf. Computer and Communications Security*, pp. 253–264 (2012).

[4] Garfinkel, T. and Rosenblum, M.: A Virtual Machine Introspection Based Architecture for Intrusion Detection, *Proc. Network and Distributed Systems Security Symposium*, pp. 191–206 (2003).

[5] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and the Art of Virtualization, *Proc. Symp. Operating Systems Principles*, pp. 164–177 (2003).

[6] 飯田，光来 : VM Shadow : 既存IDSをオフロードするための実行環境，第119回OS研究会 (2011).

[7] Wang, J., Stavrou, A. and Ghosh, A.: HyperCheck: A Hardware-Assisted Integrity Monitor, *Proc. Intl. Symp. Recent Advances in Intrusion Detection*, pp. 158–177 (2010).