Vol.2013-OS-126 No.20
2013/8/1

クラウド上の仮想マシンの安全なリモート監視機構

重田 一樹[1]　光来 健一[1,2]

概要：侵入検知システム（IDS）を安全に実行できるようにするために，仮想マシン（VM）を用いたIDSオフロード手法が提案されている．この手法は監視対象のVMとは別のVMでIDSを動作させる手法である．しかし，クラウド内でIDSをオフロードする場合，クラウドの管理者が常に信頼できるとは限らないことからIDSの正常な実行を保証できない．本稿では，監視対象VMが動作しているクラウドとは別のホストにIDSをオフロードし，ネットワーク経由で安全に監視対象VMを監視できるようにするシステムRemoteTransを提案する．RemoteTransでは信頼できるホスト上でIDSを動作させることで，IDSの停止や改ざんを防ぐ．また，IDSとクラウド内の仮想マシンモニタ（VMM）の間で整合性チェックを行うことで，IDSが参照するデータの改ざんを検出する．RemoteTransをXenに実装し，VM Shadowを用いて既存のIDSを動作させられることを確認した．

キーワード：仮想マシン，リモート監視

Secure Remote Monitoring of Virtual Machines on Clouds

Kazuki Juda[1]　kenichi kourai[1,2]

Abstract: To execute intrusion detection systems (IDSes) securely, offloading IDSes with virtual machine (VMs) has been proposed. This technique runs IDSes on one VM and a monitored system on another VM. However, when IDSes are offloaded in clouds, trusted execution of them is not guaranteed because cloud administrators are not always trustworthy. This paper proposes RemoteTrans for securely monitoring VMs in clouds via networks by offloading IDSes on other hosts. RemoteTrans prevents IDSes from being terminated or tampered with by running them on trusted hosts. In addition, it detects the tampering with requested data by checking the integrity of the data between IDSes and the virtual machine monitor (VMM) in clouds. We have implemented RemoteTrans in Xen and confirmed that several existing IDSes could run by using VM Shadow.

Keywords: Virtual machine,remote monitoring

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，物理マシンを自分で用意する必要がなくコストを削減することができる．しかし，クラウドに集約された多くのサーバは，攻撃者による攻撃の対象になりやすい．なぜなら，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である．また，クラウドでは容易にVMを作成できるため，パッチが未適用で正しく管理されていないVMもありえる．

そのため，クラウド上のVMでサーバを動作させる場合は侵入検知システム（IDS）による監視がますます重要になる．しかし，IDSをVM内で動作させていると，VMに侵入されてIDS自身が無力化されてしまった場合，以降の攻撃を検知できなくなってしまう．この問題を解決するために，VMを用いたIDSのオフロード手法が提案されている [1]．この手法では，監視対象が動作しているVMとは

[1] 九州工業大学
Kyushu Institute of Technology
[2] 独立行政法人科学技術振興機構, CREST

ⓒ 2013 Information Processing Society of Japan

別の VM に IDS をオフロードし，VM の外からシステムの監視を行う．管理 VM などの IDS を動作させる VM ではできるだけ他のサービスを提供しないようにすることで，攻撃を受けにくくすることがで可能である．これにより，安全に IDS を動作させることが可能となる．

一方，クラウドの中で IDS オフロードを行っても IDS が正しく動作することを保証することはできない．クラウドの管理者が必ずしも信頼できるとは限らないためである．管理 VM のセキュリティ対策が不十分な場合，外部の攻撃者に侵入される恐れがある上，管理者に悪意があることも考えられる．このような場合，IDS を停止させられたり，正しく侵入を検知しないように改ざんされたりする恐れがある．また，IDS を改ざんしなくとも，IDS が監視対象 VM から取得するデータを改ざんすることで無力化することも考えられる．

本稿では，監視対象 VM が動作しているクラウドとは別のホストに IDS をオフロードし，ネットワーク経由で安全に監視対象 VM を監視できるようにするシステム RemoteTrans を提案する．RemoteTrans は，IDS を信頼できるホスト上で動作させることにより，IDS が停止されたり改ざんされたりするのを防ぐことができる．また，IDS とクラウド内の仮想マシンモニタ（VMM）の間で整合性チェックを行うことにより，要求した監視対象 VM のデータが正しく取得できており，データの内容も改ざんされていないことを確認する．VMM はリモートアテステーションなどを用いることで正しく動作することを保証する．

我々は RemoteTrans を Xen 4.1.3 [2] に実装した．クラウド外部の監視ホストで RemoteTrans ランタイムおよび IDS を動作させ，管理 VM 上の RemoteTrans サーバと通信する．また，Transcall [3] を RemoteTrans に対応させ，監視ホスト上でいくつかの既存の IDS を動かすことができている．監視対象 VM のカーネルデータから Shadow proc ファイルシステムを構築する実験を行い，リモートの VM の情報を取得できていること，通信データの改ざんを検知できることを確認した．また，データの取得には従来のオフロード手法の約 15 倍の時間がかかることが分かった．

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. IaaS 型クラウドにおける IDS オフロード

IDS を安全に動作させるために，VM を用いた IDS オフロード手法が提案されている [1]．この手法は，**図 1** のように監視対象システムが動作している VM と同じホスト上の別の VM に IDS をオフロードし，システムの外側から監視する手法である．IDS を動作させる VM としては，

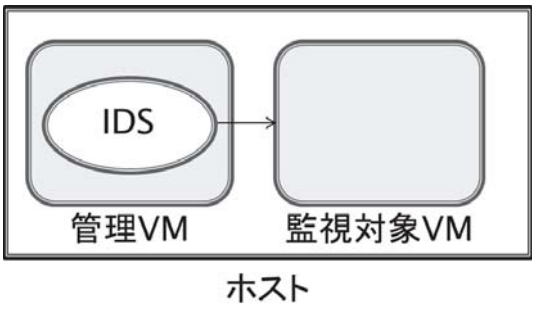

図 1 VM を用いた IDS オフロード

VM の管理を行う特権を持った管理 VM が用いられることが多いが，監視専用に開発されたドメイン M [4] などを用いることもできる．この手法を用いることにより，監視対象 VM が攻撃を受けたとしてもその中で IDS が動作していないため IDS を攻撃される恐れはない．一方，オフロード先の管理 VM では IDS 以外のシステムをできるだけ動作させないようにすることで，攻撃を受けにくくすることができる．

オフロードされた IDS は監視対象 VM のメモリを解析して情報を取得することで監視を行う．例えば，監視対象 VM に悪意のあるプロセスが動いていないかどうかを監視するには，カーネルメモリ上にあるプロセスリストの先頭からプロセス情報を順番に取得する．そして，プロセスの名前や所有者などをチェックしたり，プロセスのメモリを調べたりすることで異常の有無を検査する．他にも，監視対象 VM のディスクを検査したり，送受信するネットワークパケットを解析する IDS もオフロードすることができる．

IDS オフロード手法をクラウドに適用する際の問題点は，クラウドの管理者は常に信頼できるとは限らないということである．クラウド上の VM はマイグレーションで移動することがあり，セキュリティ意識の低いシステム管理者のいるデータセンタで VM が動作する可能性もある．このような環境で管理 VM に脆弱性がある場合，外部からの攻撃者によって管理 VM の制御が奪われる恐れがある．また，クラウド管理者に悪意があった場合，管理 VM にログインして容易に不正を行うことができる．

そのため，オフロードした IDS が正常に動作することを保証することはできない．管理 VM に侵入した攻撃者や悪意を持った管理者は，IDS を停止させることで侵入検知を回避することができる．また，IDS を改ざんすることで，悪意あるプロセスを検出しないようにすることもできる．IDS を改ざんしなくとも，IDS が監視対象 VM のメモリ領域を参照する際に，参照先を変更するだけでも，IDS の挙動を変えて無力化することができる．

クラウド内で安全に IDS オフロードを行えるようにするために，Self-Service Cloud（SSC) [5] が提案されている．

ⓒ 2013 Information Processing Society of Japan

SSC では，IDS をオフロードした VM は監視対象 VM のユーザの管理下にあり，クラウドの管理者であっても干渉することができない．しかし，VMM に加えて，この VM もクラウド内で信頼する部分とする必要がある．VMM と比べて，VM 内では OS を含めてより複雑なシステムが動作しているため，その脆弱性を攻撃される危険性が高くなる．

3. RemoteTrans

本稿では，監視対象 VM が動作しているクラウドとは別のホストに IDS をオフロードし，ネットワーク経由で安全に監視対象 VM を監視できるようにするシステム RemoteTrans を提案する．ユーザが管理している信頼できるホスト上で IDS を動作させることによって，IDS が停止されたり改ざんされたりすることを防ぐことができる．IDS とクラウド内の VMM の間で整合性チェックを行うことで，IDS が参照するデータの改ざんを検出することができる．

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，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 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監視ホストは正しく管理されているものとし，監視ホストが攻撃を受けることは考えない．

3.2 システム構成

RemoteTrans を用いて IDS オフロードを行う場合のシステム構成を**図 2** に示す．クラウド外の監視ホスト上で RemoteTrans ランタイムを動作させ，このランタイム上で IDS を実行する．一方，クラウド内では RemoteTrans サーバを監視対象 VM が動作しているホストの管理 VM で動作させる．クラウド内の VMM では RemoteTrans モジュールを動作させる．IDS が参照する監視対象 VM の情報は RemoteTrans ランタイムが RemoteTrans サーバを経由して RemoteTrans モジュールにアクセスすることで取得する．

RemoteTrans ランタイムと IDS はユーザの PC やプライベート・クラウドなどの様々な環境で実行できるようにするために，仮想アプライアンス（VM）として提供する．VM で動作させることの利点は IDS を動作させる環境に左右されないことである．環境が変わるとパッケージの追加や IDS の書き換えなどを行わなければならない可能性があるが，VM であればどこでもそのまま実行することができる．さらに，VM のマイグレーションを行うことで，物理マシンのメンテナンス時も監視を継続することができる．

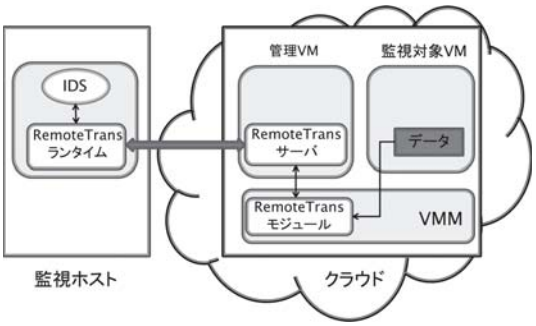

図 2 RemoteTrans のシステム構成

3.3 VM の監視

RemoteTrans ランタイムがクラウド内の RemoteTrans サーバ経由で VMM 内の RemoteTrans モジュールにアクセスすることによって，IDS はリモートの監視対象 VM の情報を参照することができる．IDS が監視対象 VM のメモリデータを参照しようとした時，そのデータの仮想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，リクエストされた仮想アドレスにあるデータを指定されたサイズ分だけ監視対象 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 ランタイムに返され，IDS が参照できるようになる．

RemoteTrans では，VMCrypt [6] やセキュアな実行環境 [7] を用いて監視対象 VM のメモリを暗号化することを想定しているため，VMM 内で VM のメモリデータを取得する．さらに，VMM 内の RemoteTrans モジュールはリクエストとレスポンスに対する改ざんの検出も行う．この整合性チェックについては 3.4 節で述べ，クラウド内の VMM を信頼するための手法については 3.5 節で述べる．VMM ではデータの取得だけを行い，データの送信は行わない．これは，RemoteTrans ランタイムと VMM が直接通信できるようにすると，VMM が攻撃を受ける可能性が高まるためである．VMM が攻撃を受けると，IDS が参照するデータの整合性チェックを無効化されてしまう恐れがある．

管理 VM 上にオフロードして同一ホスト上の VM を監視するように開発された IDS については，監視対象 VM のデータ取得部を RemoteTrans ランタイムが提供する API を用いて書き換えるだけで，リモートの VM を監視することが可能となる．また，VM Shadow [3] を RemoteTrans に対応させることにより，既存の IDS を動作させることも可能となる．VM Shadow とは，既存の IDS に修正を加え

ⓒ 2013 Information Processing Society of Japan

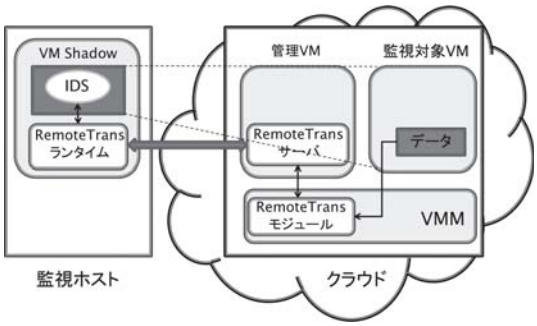

図 3 RemoteTrans に対応した VM Shadow

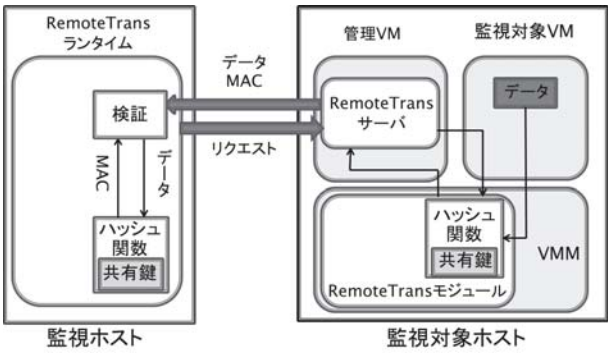

図 4 整合性チェック

ることなくオフロードして動作させられるようにするための実行環境である．**図 3** に VM Shadow を用いた場合の構成を示す．

3.4 監視データの整合性チェック

RemoteTrans への攻撃として，RemoteTrans ランタイムから RemoteTrans サーバへのリクエストおよびそのレスポンスの改ざんが考えられる．通信路が暗号化されていたとしても，クラウド内の管理 VM 上でこれらの改ざんが行われる危険性がある．もし，リクエストが改ざんされた場合，指定された仮想アドレスを変更し，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ために次のプロセスのアドレスを取得するリクエストが送られてきた場合，次の次のプロセスを指すアドレスに書き換えることができる．これにより悪意のあるプロセスの情報を隠されてしまう恐れがある．同様に，レスポンスを改ざんされると，実際のデータとは異なるデータを返すことができる．悪意のあるプロセスの情報を書き換えることで，IDS による検知を回避されてしまう恐れがある．

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，**図 4** 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仮想アドレスとデータサイズ，それに基づいて監視対象 VM から取得したデータからメッセージ認証コード（MAC）を計算する．RemoteTrans サーバが取得したデータとともにこの 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，受信したデータから MAC を計算し，受信した MAC と比較を行う．MAC 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．攻撃者が MAC を計算できないようにするために，MAC の計算に用いる鍵は，RemoteTrans モジュールと RemoteTrans ランタイムだけで共有する．

3.5 リプレイ攻撃対策

IDS が参照するデータを改ざんする手段として，リプレイ攻撃も考えられる．リプレイ攻撃とは，以前に通信したリクエストとレスポンスを保存しておき，同じリクエストが送られた時に保存しておいたレスポンスを返す攻撃である．攻撃の例としては，IDS が正常時のプロセス情報を入手する際に，攻撃者はリクエストとレスポンスの組を保存しておく．その後，悪意のあるプロセスを監視対象 VM で動作させ，IDS がプロセス情報を入手する際には，保存しておいたレスポンスを返す．これにより，IDS に正常時のプロセス情報を参照させることができる．攻撃者はリクエストされるアドレスとデータサイズ，および，レスポンスで返されるデータを改ざんする必要がないため，MAC を用いてもこのような攻撃を検知することはできない．

RemoteTrans では，リプレイ攻撃対策としてノンスと呼ばれる乱数を含めて MAC の計算を行う．まず，RemoteTrans ランタイムはアドレスとデータサイズに加えてノンスも RemoteTrans サーバへ送る．RemoteTrans モジュールはノンスも含めてリクエストと取得したデータから MAC を計算し，RemoteTrans ランタイムに返す．RemoteTrans ランタイムは保存しておいたノンスも含めて MAC を計算し，受信した MAC と比較を行う．ノンスを MAC の計算に含めることによって，以前に使用したリクエストとレスポンスの組を再利用することはできなくなり，リプレイ攻撃を防ぐことができる．

3.6 VMM の完全性チェック

RemoteTrans では，クラウド内で正しい VMM が動作していることを確認するために，リモートアテステーションを用いる．サーバの起動時に VMM のハッシュ値を計算し，クラウドの外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耐タンパ性ハードウェア（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．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

ⓒ 2013 Information Processing Society of Japan

能により実行時の VMM の改ざんも防ぐことができる．

3.7 鍵管理

RemoteTrans では，MAC を計算するときに用いる共有鍵を RemoteTrans ランタイムと VMM 内の RemoteTrans モジュールの間で安全に共有する．RemoteTrans ランタイムが RemoteTrans サーバにアクセスする際に接続先の VMM の公開鍵を鍵サーバから取得する．この鍵サーバは信頼できるものとし，あらかじめ正当な VMM の公開鍵が登録されているものとする．RemoteTrans ランタイムで生成した共有鍵を VMM の公開鍵を用いて暗号化し，RemoteTrans サーバに送る．RemoteTrans サーバは暗号化された公開鍵を RemoteTrans モジュールに送り，VMM にある秘密鍵を用いて復号化することで IDS の実行ごとに異なる共有鍵を使う．VMM の秘密鍵は TPM を用いて封印（暗号化）しておくことで，正しい VMM が起動したときだけ封印を解除（復号化）することができる．

4. 実装

我々は RemoteTrans を Xen 4.1.3 に実装した．特権を持っている VM であるドメイン 0 で RemoteTrans サーバを動作させ，通常の VM であるドメイン U を監視対象 VM とした．監視対象 VM の OS として Linux 2.6.27.35 を用いた．

4.1 RemoteTrans ランタイム

RemoteTrans ランタイムでは，リモートの VM からデータを取得するために rt_get_data 関数と rt_get_proc_data 関数を提供する．

rt_get_data 監視対象 VM のカーネルデータを参照するときに呼び出す．引数として仮想アドレス，データサイズを取り，取得したデータを確保したメモリに格納し，そのアドレスを返す．

rt_get_proc_data 監視対象 VM の中のプロセスのカーネルデータを参照するときに呼び出す．例えば，プロセスを起動したときのコマンドラインはプロセスのメモリ上に格納されている．この関数は仮想アドレスとデータサイズに加えて，プロセスのページグローバルディレクトリ（PGD) の仮想アドレスを引数に取り，取得したデータが格納されたメモリのアドレスを返す．

RemoteTrans サーバにリクエストを送る際には，ノンスとして生成した乱数も一緒に送信する．レスポンスが返されると MAC の検証を行い，受信した MAC の値と異なる場合は関数の返り値として NULL を返す．

これらの関数を最初に呼び出した時に，RemoteTrans サーバに接続する．その際に 3.7 節のように共有鍵を送る仕様となっているが，現在の実装では事前に鍵を共有している．

4.2 RemoteTrans サーバ

RemoteTrans サーバは，RemoteTrans ランタイムからリクエストを受けとってから，ハイパーコールを用いて VMM 内の RemoteTrans モジュールに受信したリクエストを送る．RemoteTrans モジュールではまず，受け取った仮想アドレスをマシンメモリのフレーム番号（MFN）に変換する．rt_get_data 関数からのリクエストの場合，カーネルのページテーブルをたどることで変換を行う．rt_get_proc_data 関数からのリクエストの場合，PGD が指すプロセスのページテーブルをたどることで変換を行う．次に，見つけた MFN が指すメモリページをマップして必要なデータを取得する．リクエストされたデータがページ境界にまたがる場合，連続するページのデータをマップしデータを取得する．

その後，リクエストと取得したデータから MAC を計算する．MAC の計算はハッシュ関数の SHA-1 を用いる．RemoteTrans モジュールが取得したデータと計算した MAC を RemoteTrans サーバに返すと，RemoteTrans サーバはそれをレスポンスとして RemoteTrans ランタイムに送信する．

4.3 Transcall の RemoteTrans 対応

Transcall は，既存の IDS をオフロードするための実行環境 VM Shadow を提供するシステムである．Transcall の構成を**図 5** に示す．Transcall 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するシステムコールをエミュレートし，監視対象 VM のカーネル内の情報を返す．ただし，オフロード先 VM の機能を利用できるメモリ管理などのシステムコールは，その VM 内の OS に対して発行する．Shadow ファイルシステムは Transcall 実行時に監視対象 VM のディスクイメージをオフロード先 VM にマウントする．それにより，監視対象 VM 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．また，特殊なファイルシステムとして Shadow proc ファイルシステムも提供する．このファイルシステムは監視対象 VM の proc ファイルシステムの情報を提供し，カーネルの現在の状態や現在実行中のプロセス情報などを含んでいる．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．Transcall では，監視対象 VM のカーネルメモリから直接情報を取得することで proc ファイルシステムの情報を取得する．

Transcall を RemoteTrans に対応させ，監視対象 VM の proc ファイルシステムをネットワーク経由で取得できるよ

ⓒ 2013 Information Processing Society of Japan

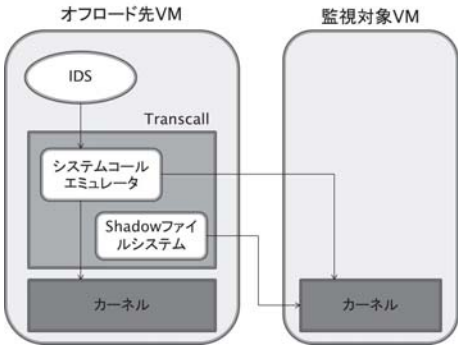

図 5 Transcall のシステム構成

うにした．我々はリモートのホストに既存の IDS をオフロードできるようにすることを目標としている．現在のところ，システムコール・エミュレータと Shadow proc ファイルシステムを RemoteTrans ランタイム上で実行でき，監視対象 VM の proc ファイルシステムをネットワーク経由で取得することができている．そのために，監視対象 VM のメモリを参照している部分を RemoteTrans が提供する API を用いて書き換えた．監視対象 VM の仮想ディスクを参照できるようにすることは今後の課題である．

5. 実験

まず，RemoteTrans によりリクエストおよびレスポンスの改ざんが検出できることを確認する実験を行った．次に，RemoteTrans を用いてリモートの監視対象 VM の proc ファイルシステムの情報を取得し，Shadow proc ファイルシステムを構築するのにかかる時間を測定した．監視対象ホストには Intel Core i7 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 は OS に Linux 3.2.0，監視対象 VM の OS には Linux 2.6.27.35 を用いた．RemoteTrans ランタイムを動作させる監視ホストには，Intel Core i7 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 リクエストおよびレスポンスの改ざん検知

RemoteTrans における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．そのために，RemoteTrans サーバに送られてくるリクエストと RemoteTrans ランタイムへ返すレスポンスの改ざんを行った．具体的には，特定のプロセス情報を持つアドレスが送られてきた時にリクエストに含まれる仮想アドレスとデータサイズ，レスポンスに含まれるデータの改ざんをそれぞれ行った．RemoteTrans を用いて proc ファイルシステムの取得を行おうとしたところ，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．

表 1 Shadow proc ファイルシステム構築時間（秒）

	実行時間
従来システム	1.1
RemoteTrans	16.4

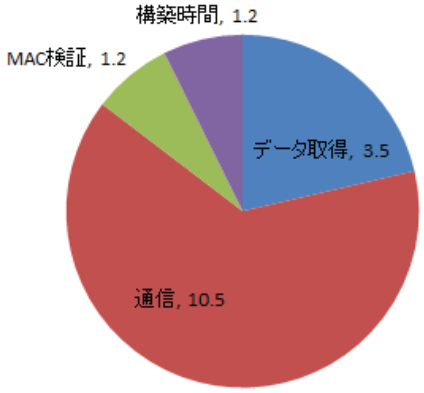

図 6 RemoteTrans の実行時間の内訳（秒）

5.2 リプレイ攻撃検知

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．リプレイ攻撃として，以前に返されたレスポンスを保存しておき，後でそれを RemoteTrans ランタイムに返すようにした．実験の結果，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．

5.3 既存の IDS の動作確認

RemoteTrans に対応した VM Shadow を用いて，既存の IDS が動作することを確認する実験を行った．Shadow proc ファイルシステムを参照する ps と netstat コマンドの動作確認を行ったところ，それぞれ監視対象 VM の実行結果を返すことを確認した．ただし，netstat の実行結果の一部が正しく取得できていなかった．

5.4 Shadow proc ファイルシステム構築時間

Shadow proc ファイルシステムの構築にかかる時間を従来システムと RemoteTrans を用いたシステムとで比較した．実験結果を表 1 に示す．RemoteTrans を用いたシステムでは，従来システムの 15 倍程度の時間がかかってしまっていることがわかる．

次に，構築におけるボトルネックを調べるために実行時間の内訳を調べた．RemoteTrans を用いた構築の際には，データ通信，ハイパーコールを用いた VMM でのデータ取得および MAC の計算，RemoteTrans ランタイム上での MAC の検証などが行われる．実験結果を**図 6** に示す．この結果より，通信に 10.5 秒かかっており，実行時間の 64%を占めていることがわかった．Shadow proc ファイルシステムを構築するためにデータ送受信は 34210 回行われていた．

ⓒ 2013 Information Processing Society of Japan

6. 関連研究

Self-Service Cloud（SSC）[5] 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．

CloudVisor [8] 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．

Copilot [9] はカーネルの整合性をリモートからチェックできるシステムである．PCI カードを用いてリモートホストにカーネルメモリを送り，カーネルの改ざんを検出することができる．しかし，クラウド内のすべてのホストに専用の PCI カードを導入するのは現実的ではない．

HyperCheck [10] 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．

HyperGuard [11]，HyperSentry [12]，Flicker [13]，SPE Observer [14] は，ハードウェアの機能を用いることによってクラウド内で安全に IDS を動作させることができる．HyperGuard は SMM 上で VMM のメモリチェックを行う．HyperSentry は SMM を利用して VMM 内の IDS を安全に実行する．Flicker は Intel TXT や AMD SVM を用いて安全に IDS を実行する．SPE Observer は Cell/B.E. の Isolation モードを用いて SPE 上で安全に IDS を実行する．しかし，監視中にシステムの他の部分を停止させなければならなかったり，一般的でないハードウェアが必要となったりする．

セキュアな実行環境 [7] や VMCrypt [6] 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．また，VM のメモリの改ざんも検出することができる．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 を用いればリモートの監視ホストでメモリデータを復号化することで，オフロードした IDS の実行が可能になる．

7. まとめ

本稿では，監視対象 VM が動作しているクラウドとは別のホストに IDS をオフロードし，ネットワーク経由で安全に監視対象 VM を監視できるようにするシステム RemoteTrans を提案した．RemoteTrans は IDS を信頼できるホスト上で動作させることにより，IDS の停止・改ざんを防ぐ．また，IDS とクラウド内の VMM の間で整合性チェックを行うことで，参照するデータの改ざんを検出する．RemoteTrans を Xen に実装し，リクエストおよびレスポンスの改ざんを検知できることを確認した．また，Transcall を RemoteTrans に対応させることで，既存の IDS を動作させられるようにした．

今後の課題はデータ取得の高速化である．現在は一つずつデータをリクエストして取得しているが，一括して取得できるようにする方法が考えられる．例えば，1 回のリクエストでプロセスリストをたどってすべてのプロセスデータを取得すれば，通信回数を大幅に削減できる．また，メモリデータだけでなく，ディスクデータを取得できるようにすることも課題の一つである．ディスクデータはメモリデータと比べて巨大であるため，通信量を減らす工夫が必須である．

参考文献

[1] Garfinkel, T. and Rosenblum, M.: A Virtual Machine Introspection Based Architecture for Intrusion Detection, *Proceedings of Network and Distributed Systems Security Symposium*, pp. 191–206 (2003).
[2] 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, *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 164–177 (2003).
[3] 飯田貴大，光来健一：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).
[4] 宇都宮寿仁，光来健一：VM マイグレーションを可能にする IDS オフロード機構，第 28 回日本ソフトウェア科学会大会 (2011).
[5] Butt, S., Lagar-Cavilla, H. A., Srivastava, A. and Ganapathy, V.: Self-service cloud computing, *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 253–264 (2012).
[6] Tadokoro, H., Kourai, K. and Chiba, S.: Preventing Information Leakage from Virtual Machines' Memory in IaaS Clouds, *IPSJ Transactions on Advanced Computing Systems*, Vol. 5, No. 4, pp. 101–111 (2012).
[7] Li, C., Raghunathan, A. and Jha, N. K.: Secure Virtual Machine Execution under an Untrusted Management OS, *Proceedings of IEEE CLOUD'10*, pp. 172–179 (2010).
[8] Zhang, F., Chen, J., Chen, H. and Zang, B.: CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization, *Proceedings of*

ⓒ 2013 Information Processing Society of Japan

the Twenty-Third ACM Symposium on Operating Systems Principles, pp. 203–216 (2011).

[9] Petroni, Jr., N. L., Fraser, T., Molina, J. and Arbaugh, W. A.: Copilot - a coprocessor-based kernel runtime integrity monitor, *Proceedings of the 13th conference on USENIX Security Symposium*, pp. 13–13 (2004).

[10] Wang, J., Stavrou, A. and Ghosh, A.: HyperCheck: A Hardware-Assisted Integrity Monitor, *Proceedings of International Symposium of Recent Advances in Intrusion Detection*, pp. 158–177 (2010).

[11] Rutkowska, J., Wojtczuk, R. and Tereshkin, A.: Xen 0wning Trilogy, *Black Hat USA* (2008).

[12] Azab, A. M., Ning, P., Wang, Z., Jiang, X., Zhang, X. and Skalsky, N. C.: HyperSentry: enabling stealthy in-context measurement of hypervisor integrity, *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 38–49 (2010).

[13] McCune, J. M., Parno, B., Perrig, A., Reiter, M. K. and Isozaki, H.: Flicker: An Execution Infrastructure for TCB Minimization, *Proceedings of European Conference of Computer Systems*, pp. 315–328 (2008).

[14] Kourai, K. and Nagata, T.: A Secure Framework for Monitoring Operating Systems Using SPEs in Cell/B.E., *Proceedings of Pacific Rim International Symposium on Dependable Computing*, pp. 41–50 (2012).

ⓒ 2013 Information Processing Society of Japan