クラウドにおける仮想化システム外部からの安全な VM 監視機構

美山 翔平　光来 健一

仮想マシン（VM）内で侵入検知システム（IDS）を動作させると攻撃者の侵入時に無効化される恐れがあるため，IDS オフロードと呼ばれる手法が提案されている．この手法は IDS を監視対象 VM の外側で動作させて安全に監視を行うことを可能にする．しかし，クラウドにおいては管理者が信頼できるとは限らないため，オフロードした IDS が正しく動作していることを保証するのは難しい．本稿では，信頼できない仮想化システムの外側から安全にユーザ VM を監視するシステム V-Met を提案する．V-Met は，ネストした仮想化と呼ばれる技術を用いて従来の仮想化システム全体を VM 内で動作させ，その外部で IDS を動作させる．オフロードされた IDS は仮想化システム内のユーザ VM のメモリ，ディスク，ネットワークから情報を取得し，ユーザ VM への侵入を検知する．V-Met 上で Transcall を動作させることで，既存の IDS を実行することも可能とする．我々は V-Met を Xen 4.4 に実装し，オフロードした chkrootkit と Tripwire の実行オーバヘッドが非常に小さいことを確認した．

1　はじめに

近年，急速に普及している IaaS 型クラウドはネットワークを介してユーザに仮想マシン（VM）を提供し，ユーザは自由にシステムを構築することができる．その一方で，サーバ設定の不備やセキュリティアップデートの未適用など，クラウド内のユーザ VM は十分に管理されているとは限らない．そのため，外部から侵入されて機密情報が漏洩しないように，侵入検知システム（IDS）を用いてユーザ VM を監視することが必要となっている．ユーザ VM 内で IDS を動作させても侵入時に無効化されてしまうため，IDS を安全に実行するために IDS オフロードと呼ばれる手法が提案されてきた [4]．この手法は IDS をユーザ VM の外側で動作させることにより安全に監視を行うことを可能にする．これにより，IDS が攻撃を検知する前に攻撃者によって無効化されることを防ぐことができる．

一方，クラウドにおいては管理者が十分に信頼できるとは限らない [7] [6] [13] [10] [11] ため，IDS オフロードを行っても IDS が正しく動作していることを保証するのが難しい．クラウド管理者に悪意があった場合，IDS を無効化される危険性がある．悪意はなくとも，クラウド管理者が十分なセキュリティ対策を行っていない場合，外部から IDS が攻撃を受ける可能性も考えられる．従来，クラウド上で安全に IDS オフロードを行うために，仮想化システム内のハイパーバイザを信頼する手法が提案されてきた [3] [5]．しかし，クラウド管理者は比較的容易にハイパーバイザを攻撃することができる．また，信頼できるとは限らない一般のクラウド管理者には仮想化システムの一部しか管理させられなくなるという問題もあった．

そこで本稿では，仮想化システムの外側から安全にユーザ VM を監視するシステム V-Met を提案する．V-Met では，ネストした仮想化 [2] を用いて従来の仮想化システム全体を VM 内で動作させ，その外部で IDS を動作させる．これにより，仮想化システムの管理権限を持ったクラウド管理者でさえ IDS を無効化するのは難しくなる．また，一般のクラウド管理者が従来通りに仮想化システム全体の管理を行うことができる．オフロードされた IDS は仮想化システム内のユーザ VM のメモリ，ディスク，ネットワー

Shohei Miyama, Kenichi Kourai, 九州工業大学, Kyushu Institute of Technology.

クから情報を取得し，ユーザ VM への侵入を検知する．V-Met 上で Transcall [14] を動作させることで，既存の IDS を実行することも可能となる．

我々は V-Met を Xen 4.4 のハイパーバイザおよび管理 VM に実装した．このハイパーバイザ上の VM 内で従来の仮想化システムを動作させ，管理 VM 内で IDS を動作させる．仮想化システム内のユーザ VM のメモリ上のデータを取得するために，V-Met はユーザ VM のページテーブルと拡張ページテーブル (EPT) を用いてアドレス変換を行う．ページテーブルと EPT はコンテキストスイッチの際に発生する VM Exit を補足して特定する．また，ebtables を用いることで，MAC アドレスに基づいて特定のユーザ VM が送受信するパケットだけを取得する．V-Met と Transcall を用いて chkrootkit および Tripwire をオフロード実行したところ，オーバヘッドは非常に小さいことが分かった．

以下，2 章でクラウドにおける IDS オフロードについて詳説し，3 章ではそれらの問題点をふまえて提案システムである V-Met について述べる．4 章では V-Met の実装について述べ，5 章で V-Met と従来手法との比較のために行った実験の結果を示す．6 章で関連研究について述べ，7 章で本稿をまとめる．

2 クラウドにおける IDS オフロード

2.1 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 [4]．この手法は，図 1 のように，ユーザ VM の外側の仮想化システム内で IDS を動作させ，安全に監視を行うことを可能にする．この手法を用いることにより，ユーザ 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，仮想化システム内のユーザ VM の外側では IDS 以外のサービス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 はユーザ VM から情報を直接取得することで監視を行う．例えば，ユーザ VM 内で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，カーネルメモリを解析してプロセス

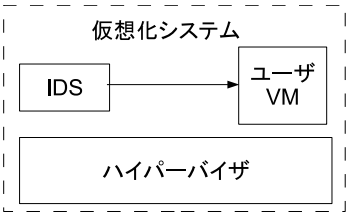

図 1　IDS オフロード

情報を取得し，プロセスの名前や所有者などをチェックする．また，ユーザ VM のファイルが改ざんされていないかどうかを監視するには，仮想ディスク上のファイルシステムを解析し，ファイルの属性や内容のチェックを行う．ネットワーク経由の不正アクセスを監視するには，仮想 NIC からパケットを取得して攻撃を検出する．

2.2 クラウドにおける IDS オフロード

クラウドおいて IDS オフロードを行うにあたって問題となるのは，管理者が常に信頼できるとは限らない [7] [6] [13] [10] [11] ということである．管理者に悪意があった場合，仮想化システム内で容易に不正アクセスを行うことができる．また，クラウド上のユーザ 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管理者やスキルの低い管理者によって管理されている仮想化システム内でユーザ VM が動作する可能性もある．仮想化システムに脆弱性がある場合，外部からの攻撃者によって制御を奪われる恐れもある．

そのため，クラウドではオフロードされた IDS が安全に動作することを担保することができない．悪意を持った管理者や仮想化システムに侵入した攻撃者は，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，攻撃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 がユーザ VM を参照する際に，参照先を変更するだけでも IDS の挙動を変えて無効化することができる．

2.3 従来手法の問題

これまでに，クラウド上で安全に IDS オフロードを行うために，仮想化システム内のハイパーバイザを信頼する手法が提案されてきた．例えば，Self-Service Cloud (SSC) [3] では，ユーザはハイパーバイザ上でサービスドメインと呼ばれる VM を安全に起動することができ，その中に IDS をオフロードすることができる．クラウド管理者であってもサービスドメインに干渉することはできない．RemoteTrans [5] はクラウドの外部に IDS をオフロードし，クラウド内で動作するハイパーバイザ経由でユーザ VM を監視する．

しかし，この手法では管理者は比較的容易にハイパーバイザを攻撃することができる．ハイパーバイザは仮想化システムの管理コンポーネントに様々なインタフェースを提供しており，管理者はそれらのインタフェースにアクセス可能であるためである．ハイパーバイザが攻撃されると，IDS を無効化されたり，改ざんされたりする可能性があり，IDS を安全に実行することができなくなる．

また，一般の管理者が仮想化システム全体を管理できなくなるという問題もある．信頼できないかもしれない管理者にハイパーバイザを管理させることはできないためである．一般に，仮想化システムのアップデートはパッケージを用いて行われるが，パッケージ間の依存関係のために仮想化システム全体を同時にアップデートする必要がある．そのため，ハイパーバイザのアップデートだけを別に行えるようにするには，管理手法の大幅な変更が必要となる．

3 V-Met

本稿では，ネストした仮想化 [2] を用いて仮想化システムの外側で IDS を動作させ，安全に VM を監視するシステム V-Met を提案する．図 2 に V-Met のシステム構成を示す．V-Met では，仮想化システムの中から外側へのアクセスは大きく制限される．そのため，仮想化システムの外側で動作している IDS を攻撃するのは難しい．その一方で，一般のクラウド管理者に仮想化システム全体を管理する権限を与えることができるため，従来通りの管理を行うことが可能となる．

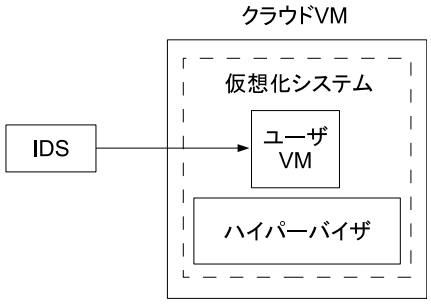

図 2　V-Met のシステム構成

ネストした仮想化は，従来の仮想化システム全体を VM 内で動作させることを可能にする技術である．本稿ではこの VM をクラウド VM と呼ぶ．V-Met では，クラウド VM の外側に IDS をオフロードし，IDS はクラウド VM 内のユーザ VM を監視する．V-Met や IDS はクラウドの信頼できる管理者が管理し，クラウド VM 内の仮想化システムは信頼できるとは限らない一般のクラウド管理者が管理する．ネストした仮想化による性能低下が起こるが，オーバヘッドを削減するために様々な手法が提案されている [3]．これらの手法を用いることで性能低下を 6〜8%程度に抑えることが可能である．

V-Met では，オフロードした IDS はクラウド VM のメモリの中からユーザ VM のデータを見つけて監視を行う．クラウド VM のメモリ上には複数のユーザ VM のメモリが含まれているため，監視対象のユーザ VM のメモリを特定し，さらにその中にある目的のデータを特定する必要がある．そのために，V-Met では 3 回のアドレス変換を行う．まず，ユーザ VM 内のページテーブルを用いて，対象データの仮想アドレスをユーザ VM の物理アドレスに変換する．次に，ユーザ VM 用の拡張ページテーブルを用いて，クラウド VM の物理アドレスに変換する．最後に，クラウド VM 用の拡張ページテーブルを用いてマシンアドレスに変換する．

ユーザ VM が送受信するネットワークパケットはクラウド VM の仮想 NIC から取得する．ユーザ VM が送信したパケットはクラウド VM を経由して送信され，ユーザ VM へのパケットもクラウド VM を経由してユーザ VM に受信される．このように，クラ

ウド VM の仮想 NIC では複数のユーザ VM のパケットが混ざり合う．そのため，V-Met は宛先や送信元の MAC アドレスを基にパケットを振り分け，IDS が特定のユーザ VM のパケットだけを取得できるようにする．また，IDS がユーザ VM の仮想ディスクにアクセスできるようにするために，V-Met ではネットワークストレージを用いて仮想化システムと IDS とでディスクイメージを共有する．クラウドにおいてユーザ VM はマイグレーションされるため，ネットワークストレージを用いる構成は従来のクラウドにおけるものと同じである．

V-Met 上で Transcall [14] を動作させることで，既存の IDS を実行することができる．Transcall は，既存の IDS をオフロードするための実行環境を提供するシステムである．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 IDS が発行するシステムコールをエミュレートし，ユーザ VM のカーネル情報を返す．Shadow ファイルシステムはユーザ 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．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．Shadow proc ファイルシステムはユーザ VM のカーネルの状態や実行中のプロセス情報などを含んでいる．

4 実装

我々は V-Met を Xen 4.4 に実装した．V-Met の実際のシステム構成を図 3 に示す．V-Met を実装したハイパーバイザをクラウドハイパーバイザとして動作させ，その上でクラウド VM とクラウド管理 VM を動作させる．クラウド管理 VM はクラウド VM の管理を行うための権限を持った VM である．クラウド VM 内では既存のハイパーバイザとユーザ VM を動作させる．クラウド VM およびユーザ VM はどちらも Intel VT-x を用いて完全仮想化で動作させることを想定している．

4.1 メモリ監視

クラウド管理 VM からユーザ VM のメモリ上のデータにアクセスするために，V-Met は図 4 のようにアドレス変換を行う．まず，ユーザ VM 内のページテーブルを用いて対象データの仮想アドレスをユーザ VM の物理アドレスに変換する．ユーザ VM 内のページテーブルを用いてアドレス変換を行うために，V-Met はページディレクトリのアドレスを取得する必要がある．ページディレクトリのアドレスは仮想 CPU の CR3 レジスタに格納されているが，この CR3 レジスタは仮想化システム内のハイパーバイザが管理している．しかし信頼できないハイパーバイザから取得した情報を利用することはできない．

そこで，V-Met では図 5 のようにハイパーバイザに依存せずに CR3 レジスタの値を取得する．そのために，ユーザ VM が CR3 レジスタの値を変更しようとした時にクラウドハイパーバイザに対して VM Exit を発生させる．従来は，CR3 レジスタへの書き込みが行われても VM Exit は発生しなかった．そこで，コントロールレジスタの読み書きで VM Exit が

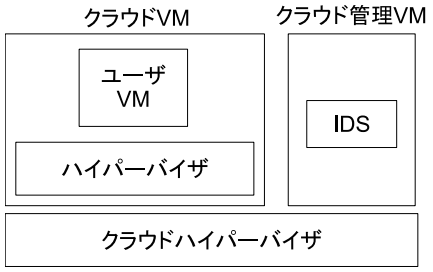

図 3　V-Met の実際のシステム構成

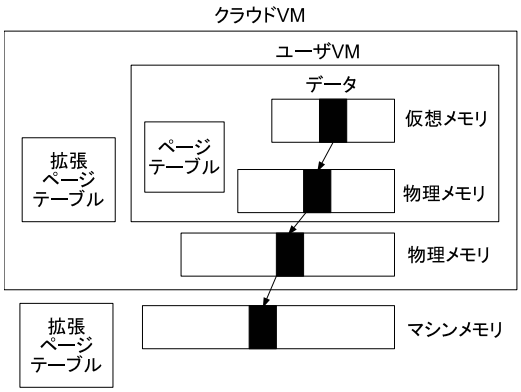

図 4　ゲスト VM にアクセスするためのアドレス変換

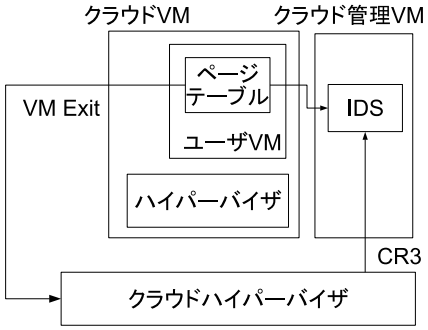

図 5　CR3 の取得

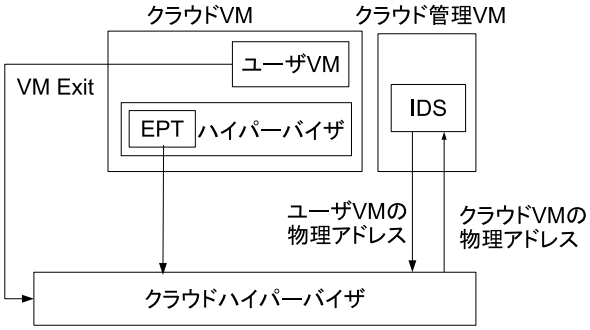

図 6　EPT を用いたアドレス変換

発生するように，クラウドハイパーバイザにおいてクラウド VM の仮想 CPU 内にある VMCS の VM 実行制御フィールドに設定を行うようにした．クラウドハイパーバイザでは CR3 レジスタに対する書き込みかどうかを判定し，そうであれば書き込み元のレジスタの値を保存する．クラウド管理 VM 上の IDS は新たに追加したハイパーコールを用いてクラウドハイパーバイザに保存されている最新の CR3 レジスタの値を取得し，ページテーブルをたどってアドレス変換を行う．

次に，ユーザ VM 用の拡張ページテーブル（EPT）を用いて，ユーザ VM の物理アドレスをクラウド VM の物理アドレスに変換する．そのために，仮想化システム内のハイパーバイザが管理している EPT のアドレスを取得する必要がある．V-Met では，図 6 のように，ユーザ VM が CR3 レジスタへの書き込みによって VM Exit が発生した時に，ユーザ VM の仮想 CPU 内にある VMCS から EPT のアドレスを取得する．IDS がハイパーコールを呼び出した時に，ハイパーコール内でアドレス変換を行う．

信頼できない仮想化システム上で動作するユーザ VM 内のページテーブルは，CloudVisor [13] のメモリ隔離技術を用いて保護する．CloudVisor は仮想化システムからユーザ VM のメモリへのアクセスを制限するため，ハイパーバイザや管理 VM がユーザ VM のページテーブルを改ざんすることはできない．同様に，ハイパーバイザ内の EPT も CloudVisor のメモリ所有者追跡技術を用いて保護する．CloudVisor はユーザ VM のメモリだけを EPT に登録可能としているため，ハイパーバイザが EPT を改ざんするのは難しい．

最後に，クラウド VM 用の EPT を用いて，クラウド VM の物理アドレスをマシンアドレスに変換する．この EPT はクラウドハイパーバイザ内にあり，クラウド VM のメモリページをマップするハイパーコールを実行する際にこのアドレス変換が自動的に行われる．

4.2　ネットワーク監視

IDS はクラウド管理 VM に作られたクラウド VM 用の仮想ネットワークインタフェース（vif）を通して，ユーザ VM のネットワークパケットを取得する．この vif からはクラウド VM が送受信するすべてのパケットが取得される．ユーザ VM の MAC アドレスはネットワーク上で一意に決まるので，MAC アドレスでフィルタリングすることで，特定のユーザ VM のパケットだけを取得する．

パケットのフィルタリングには ebtables を用いた．ebtables は Ethernet フレームをフィルタし，フィルタしたパケットをユーザ空間で扱うことがことを可能にする．V-Met は，パケットが vif からネットワークブリッジに送られてきた時は送信元 MAC アドレスを基にパケットを振り分ける．逆に，ネットワークブリッジから vif に送られてきた時は宛先 MAC アドレスを基にパケットを振り分ける．V-Met は MAC アドレスごとに OpenVPN を用いて tap デバイスを作成し，振り分けたパケットを tap デバイスに書き込む．マルチキャストやブロードキャストのパケットの

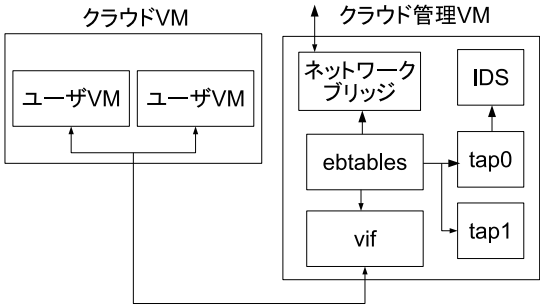

図 7　ネットワーク監視の流れ

場合には，作成したすべての tap デバイスにパケットを書き込む．IDS は tap デバイスからパケットをキャプチャすることで，従来通りのネットワーク監視を行うことができる．

4.3　ディスク監視

クラウド管理 VM はユーザ VM のディスクイメージが置かれたディレクトリを NFS マウントし，そのディスクイメージをループバック・マウントする．その際には，クラウド管理 VM の OS 内のファイルシステムを用いる．IDS はディスクイメージがマウントされたディレクトリを参照することで，ユーザ VM のファイルシステムにアクセスする．一方，仮想化システム内でも同様に NFS マウントを行い，マウントしたディレクトリ上のディスクイメージを用いてユーザ VM を起動する．

4.4　Transcall の移植

我々は Transcall [14] を V-Met 上に移植した．既存の Transcall は VM の仮想 CPU の状態を取得するハイパーコールを呼び出して CR3 レジスタの値を取得していた．そこで，V-Met においてユーザ VM の CR3 レジスタの値を取得できるようにするために，4.1 節のようにクラウドハイパーバイザが保存した CR3 レジスタの値を取得するハイパーコールを呼び出すように変更した．また，既存の Transcall では，ページテーブルエントリに格納された物理ページフレーム番号に対応するページをマップしながら VM のページテーブルをたどっていた．しかし，ユーザ VM のページテーブルエントリにはユーザ VM の物理ページフレーム番号が格納されているため，ハイパーコールを呼び出してクラウド VM の物理アドレスに変換してから対応するページをマップするように変更した．

5　実験

V-Met を用いてオフロードした IDS の監視性能を調べるための実験を行った. 実験には，Intel Xeon E3-1270v3 の CPU，16GB のメモリ，1.8TB の HDD，ギガビットイーサネットを搭載したマシンを用いた. このマシンでは，V-Met を実装した Xen 4.4 を動作させ，クラウド管理 VM 上では Linux 3.13.0 を動作させた．ホスト VM では，既存の Xen 4.4 を動作させ，管理 VM では Linux 3.13.0 を動作させた. また，ユーザ VM では Linux 3.13.0 を動作させた. 比較のために，ネストした仮想化を用いない従来システムでも実験を行い，V-Met におけるホスト VM 内の仮想化システムと同じものを動作させた．NFS サーバには，Intel Xeon X5675 の CPU，32GB のメモリ，3.75TB の RAID5 ディスク，ギガビットイーサネットを搭載したマシンを用いた．これらのマシンはギガビットスイッチで接続した．

5.1　アドレス変換のオーバヘッド

V-Met においてアドレス変換のために追加したハイパーコールのオーバヘッドを測定した．その結果，ユーザ VM 用の EPT を用いてユーザ VM の物理アドレスをクラウド VM の物理アドレスに変換するハイパーコールの呼び出しには 1.3μs かかることが分かった．一方，ユーザ VM の CR3 の値を取得するハイパーコールの呼び出しには 0.96μs かかることが分かった．従来システムではユーザ VM の CR3 の値を取得するためのハイパーコール呼び出しに 15μs かかっていた．これは CR3 の値だけでなく仮想 CPU の状態を取得する汎用のハイパーコールを用いているためである．

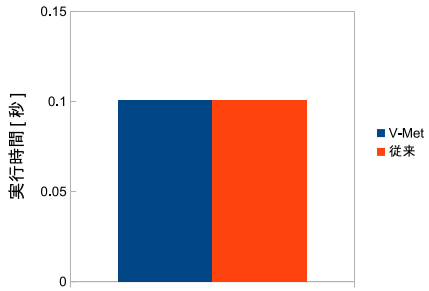

図 8　Shadow proc ファイルシステムの構築時間の比較

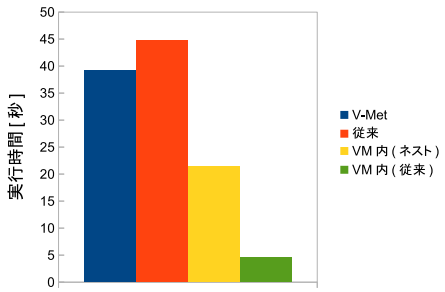

図 9　chkrootkit の実行時間

5.2　Shadow proc ファイルシステムの構築時間

Transcall はあらかじめユーザ VM 内のメモリからプロセスなどの OS の情報を取得して Shadow proc ファイルシステムを構築する．そこで，Shadow proc ファイルシステムの構築にかかる時間を V-Met と従来システムとで比較した．結果は図 8 のようになり，V-Met によるオーバヘッドはないことが分かった．V-Met では，ユーザ VM 用の EPT を用いたアドレス変換が 3，089 回必要になるが，292 回必要な CR3 の値取得のオーバヘッドが大幅に削減される．

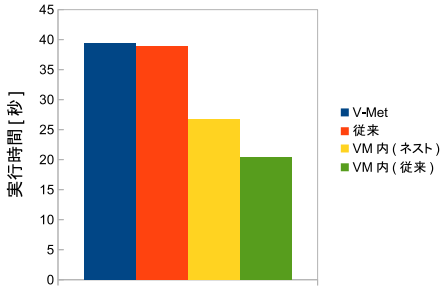

図 10　Tripwire の実行時間

5.3　オフロードした IDS の実行時間

V-Met と Transcall を用いてオフロードした chkrootkit を実行し，ユーザ VM の検査にかかる時間を測定した．chkrootkit はインストールされたルートキットを検知する IDS である．比較のために，従来システムにおいてオフロードした chkrootkit を実行した場合の時間も測定した．また，ネストした仮想化を用いたシステムと従来システムにおいてオフロードせずにユーザ VM 内で chkrootkit を実行した時にかかる時間も測定した．

実験結果は図 9 のようになり，オフロードした chkrootkit の実行時間は V-Met のほうが従来システムより短くなった．chkrootkit の中で実行される chkproc というプログラムの実行時間のばらつきが大きいことが一因であると考えられる．一方，オフロードしない場合には実行時間が大幅に長くなった．これは chkproc が Shadow proc ファイルシステムに大量にアクセスを行い、Shadow proc ファイルシステムが用いている FUSE のオーバヘッドが顕著に表れたためである。また、システムコールをトラップすることによるオーバヘッドが大きくなることも原因である。

同様に，Tripwire をオフロードして実行した場合の実行時間を測定し，オフロードしない場合とも比較した．Tripwire はファイルシステムの整合性を検査する IDS である．

実験結果は図 10 のようになり，V-Met によるオフロードのオーバヘッドはわずかであった．オフロードしない場合と比較すると，transcall がシステムコールをトラップするオーバヘッドの分だけ性能が低下した．

5.4　ネットワーク監視のオーバヘッド

V-Met を用いてネットワーク監視を行うことによるオーバヘッドを調べるために，iperf を用いてネットワーク性能を測定した．iperf サーバをユーザ VM 内で動作させ，iperf クライアントを外部ホストで動

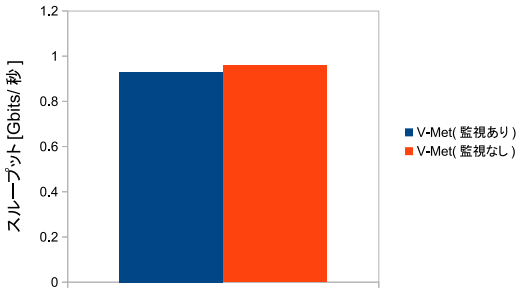

図 11　iperf の実行時間

作させた．実験結果を図 11 に示す．ネットワーク監視を行った場合にはスループットが 3.3%低下した．

6　関連研究

Self-Service Cloud (SSC) [3] は，信頼できるハイパーバイザを用いて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，IDS を動作させて他の 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った場合，攻撃を受ける可能性がある．また，ハイパーバイザを信頼する必要がある．

RemoteTrans [5] はユーザ 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ユーザ VM を監視できるようにするシステムである．リモートホスト上の IDS は信頼できるハイパーバイザと暗号通信を行い，ユーザ VM のメモリやネットワークパケット，ディスクブロックを取得する．Transcall を用いることにより既存の IDS を動作させることもできる．しかし，IDS をユーザ側で管理する必要があり，IDS を動作させるホストの安全性もユーザが担保する必要がある．

ハードウェアによる安全な実行のサポートを用いることによって，安全に IDS を動作させつつ，一般のクラウド管理者に仮想化システム全体を管理させることが可能である．HyperGuard [9] は CPU のシステムマネジメントモード（SMM）で安全にハイパーバイザのメモリチェックを行う．HyperCheck [12] は SMM を用いてメモリやレジスタの内容をリモートホストに送り，完全性のチェックを行う．HyperSentry [1] は SMM と IPMI を利用してハイパーバイザ内で安全に IDS を動作させる．しかし，SMM で IDS を実行している間はシステムの他の部分が停止してしまうという問題がある．また，SMM での実行は低速なため，IDS の性能に大きな影響を及ぼす．Flicker [8] は Intel TXT や AMD SVM を用いて安全に IDS を実行するが，SMM と同様の問題を抱えている．

CloudVisor [13] はネストした仮想化を用いてハイパーバイザの下にセキュリティモニタを導入することで，信頼できないクラウドの中で安全に VM を動作させることができる．VM はハイパーバイザと管理 VM から隔離されるため，VM からの情報漏洩を防ぐことができる．しかし，VM を安全に監視する機能は提供されていない．

7　まとめ

本稿では，ネストした仮想化を用いて IDS を仮想化システムの外側で実行するシステム V-Met を提案した．V-Met を用いることで，IDS が仮想化システムの管理者に攻撃されるのを防ぐことができる．また，信頼できない一般のクラウド管理者が従来通りにハイパーバイザを含めた仮想化システム全体の管理を行うことができる．V-Met は従来の仮想化システム全体をクラウド VM 内で動作させ，クラウド VM 内のユーザ VM のメモリ，ディスク，ネットワークの監視を可能にする．我々は既存の IDS を動作させることを可能にする Transcall を V-Met に移植し，いくつかの IDS のオーバヘッドが非常に小さいことを確認した．

今後の課題は，仮想化システム内で複数のユーザ VM が動作している時に，それぞれのユーザ VM を IDS から安全に特定できるようにすることである．現在の実装では，メモリの監視時にはユーザ VM が一つだけ動作していることを仮定しており，ネットワークの監視時にはユーザ VM の MAC アドレスが分かることを仮定している．

参 考 文 献

[1] 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 Conference on Computer and Communications Security*, 2010, pp. 38–49.

[2] Ben-Yehuda, M., Day, M. D., Dubitzky, Z., Factor, M., Har'El, N., Gordon, A., Liguori, A., Wasserman, O., and Yassour, B.-A.: The Turtles Project: Design and Implementation of Nested Virtualization., *Proceedings of Symposium on Operating Systems Design and Implementation*, 2010, pp. 423–436.

[3] Butt, S., Lagar-Cavilla, H. A., Srivastava, A., and Ganapathy, V.: Self-service cloud computing, *In Processings of Conference on Computer and Communications Security*, 2012, pp. 253–264.

[4] Garfinkel, T. and Rosenblum, M.: A Virtual Machine Introspection Based Architecture for Intrusion Detection, 2003, pp. 191–206.

[5] Kourai, K. and Juda, K.: Secure Offloading of Legacy IDSes Using Remote VM Introspection in Semi-trusted Clouds, 2016, pp. 43–50.

[6] Li, C., Raghunathan, A., and Jha, N. K.: Secure Virtual Machine Execution under an Untrusted Management OS, 2010, pp. 172–179.

[7] Li, C., Raghunathan, A., and Jha, N. K.: A Trusted Virtual Machine in an Untrusted Management Environment, 2012, pp. 472–483.

[8] McCune, J. M., Parno, B., Perrig, A., Reiter, M. K., and Isozaki, H.: Flicker: An Execution Infrastructure for TCB Minimization, 2008, pp. 315–328.

[9] Rutkowska, J., Wojtczuk, R., and Tereshkin, A.: HyperGuard, 2008.

[10] Santos, N., Gummadi, K. P., and Rodrigues, R.: Towards Trusted Cloud Computing, 2009.

[11] Tadokoro, H., Kourai, K., and Chiba, S.: Preventing Information Leakage from Virtual Machines' Memory in IaaS Clouds, 2012, pp. 156–166.

[12] Wang, J., Stavrou, A., and Ghosh, A.: HyperCheck: A hardware-assisted integrity monitor, *Proceedings of International Symposium on Recent Advances in Intrusion Detection*, 2010, pp. 158–177.

[13] Zhang, F., Chen, J., Chen, H., and Zang, B.: CloudVisor: Retrofitting Protection of Virtual Machines in Multitenant Cloud with Nested Virtualization, 2011, pp. 203–216.

[14] 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, 2011.