仮想マシン環境におけるセキュア VM 間通信機構の開発

片山　吉章　　松本　利夫

三菱電機（株）　情報技術総合研究所

1. はじめに

仮想マシン環境において，各仮想マシン（VM: Virtual Machine）は独立性が高く，あるゲスト OS 上の障害やウィルスなどの影響は，他のゲスト OS に影響しないと言われている[1]．しかしながら，従来，仮想マシン間の通信には仮想ネットワークを利用するため，これを介してあるゲスト OS から他のゲスト OS に対し，Denial of Service（DoS）攻撃を仕掛けることが可能であるという問題点があった．そこで，我々は上記問題を解決するため，仮想マシン環境においてアクセス制御可能かつ DoS 攻撃回避可能なセキュア VM 間通信機構をオープンソースの仮想化ソフトウェアである Xen[2]上に構築した．本稿では，本機構の設計と評価結果について述べる．

2. セキュア VM 間通信機能

2.1 特徴

セキュア VM 間通信（SecureIVC: Secure Inter-VM Communication）機構の特徴を以下に示す．

(1) 受信側 VM に負荷を与えないアクセス制御機能

受信側 VM が予め許可した VM からのデータ通信のみ可能なアクセス制御機能を持つ．さらに，通信時のアクセス制御に関する処理を受信側 VM では行わないようにすることにより，DoS 攻撃がなされた場合でも受信側 VM への負荷をなくすことが可能である．

(2) VM をまたがるユーザプロセス間の高速通信

Xen ではゲスト OS 上のユーザプロセス間の通信手段としては TCP/IP を介した仮想ネットワークドライバを経由する方法しか実装されていない．本機構はデバイスドライバの 1 つとして実装可能なため，VM をまたがるユーザプロセス間で TCP/IP を介さない高速な通信が可能となる．

2.2 機能

SecureIVC 機構は，Xen 上で動作する Linux のデバイスドライバとして実装し，アプリケーションに対し，表 1に示す機能を提供する．

表 1　機能一覧

関数名	機能説明
open	SecureIVC デバイスをオープンする．以後，本機能で得られたファイルディスクリプタを使って通信を行う．
read	送信側アプリケーションからのデータを受信する．
write	指定した VM 上のアプリケーションに対してデータを送信する．
ioctl	他の VM に対し，自身の VM への送信許可を設定する．不許可の VM からの送信は拒否される．
close	SecureIVC デバイスをクローズする．

3. 設計

本機構を用いた場合の全体ソフトウェア構成を図 1に示す．以下に，今回開発した SecureIVC ドライバおよびアクセス権管理テーブル（図 1 の網掛け部分）について説明する．

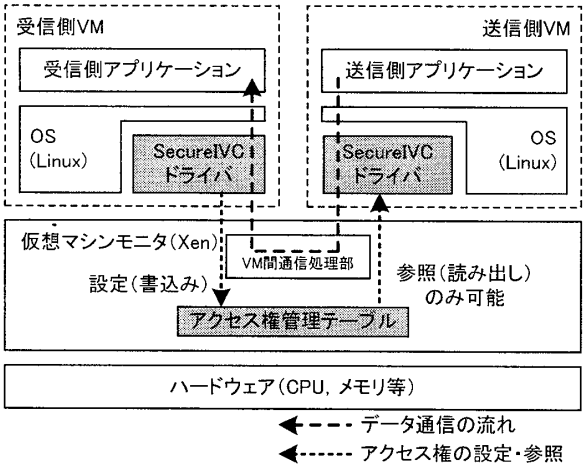

図 1　全体ソフトウェア構成

SecureIVC ドライバ：アプリケーションからの要求に基づき，仮想マシンモニタの提供する機能（Xen では共有メモリとイベントチャネル）を用いて VM 間でデータの送受信を行うものであり，次の 2 つの特徴を持つ．(1)オープン時に確保したアクセス権管理テーブルに対し，アプリケーションからの ioctl 要求によりどの VM からのデータ送信を許可するか設定する機能を持つ．

A Mechanism of Secure Inter-VM Communication for Virtual Machine Environment
Yoshiaki KATAYAMA and Toshio MATSUMOTO
Information Technology R&D Center, Mitsubishi Electric Corp.

(2)アプリケーションからの write 要求に対し，無条件に送信処理を行うのではなく，アクセス権管理テーブルを参照し，送信先の VM により送信が許可されている場合のみ実際の送信処理を行う．

アクセス権管理テーブル：受信 VM 毎に存在し，当該 VM がどの VM からのデータ送信を許可するかを記述したものである．本テーブル用のメモリ領域は仮想マシンモニタの管理下にあり，受信側 SucureIVC ドライバにより確保される．確保した領域は受信側 SecureIVC ドライバでは読み書き可能であるが，送信側を含む他の VM に対しては，リードオンリーで共有できるようメモリアクセス権を設定する．

次に，データの送信から受信までの処理フローの一例を以下に示す．

(1) データ通信を開始するため，受信側アプリケーションが SecureIVC デバイスをオープンする．
(2) 受信側 SecureIVC ドライバは，オープン要求に基づき，通信データを格納するための共有メモリおよびデータ到着を通知するためのイベントチャネルの確保・設定を行う．さらに，アクセス権管理テーブル用の共有メモリを確保し，他のドメインに対してはリードオンリーで共有できるよう設定を行う．テーブルの内容は安全のため，すべての VM からの送信を拒否するよう設定しておく．
(3) 受信側アプリケーションが ioctl 要求を出し，受信対象の VM に対してのみ，送信許可を行う．
(4) ioctl 要求を受けた受信側 SecureIVC ドライバは，当該 VM に対し送信を許可するようアクセス権管理テーブルを更新する．
(5) 受信側アプリケーションが SecureIVC デバイスに対し read（受信）要求を出す．
(6) 送信側アプリケーションが SecureIVC デバイスをオープンし，write（送信）要求を出す．
(7) write 要求を受けた送信側 SecureIVC ドライバは，送信要求に含まれる送信先を参照し，送信先 VM のアクセス権管理テーブルから送信が許可されているか否かを判定する．判定の結果，許可されていれば実際の送信処理を行う．不許可の場合はアプリケーションへエラー通知し，リターンする．

4. 評価

本章では，従来の TCP/IP によるネットワーク機能と SecureIVC 機構に関し，通信時の CPU 負荷および性能について比較を実施した結果について記述する．

4.1 測定環境

本評価に利用した環境を表 2に示す．

表 2 測定環境

CPU	Intel Pentium4 3.0GHz －シングルコア －ハイパースレッディング無効	
メモリ	全容量	1024M バイト
	ゲスト OS 割当て容量	256M バイト
OS	CentOS 5.0（カーネルは 2.6.18.8-xen）	
VM	Xen 3.2.0	

4.2 測定結果

(1) データ通信時の受信側 VM における CPU 負荷

DoS 攻撃を想定し，ある VM から送信が許可されていない他の VM に対し，4KB のデータを連続的に送信した時の受信側 VM の CPU 負荷（top コマンドの load average 値）をそれぞれ測定した（図 2）．当初の設計どおり，SecureIVC では受信側 VM への負荷はないことがわかる．

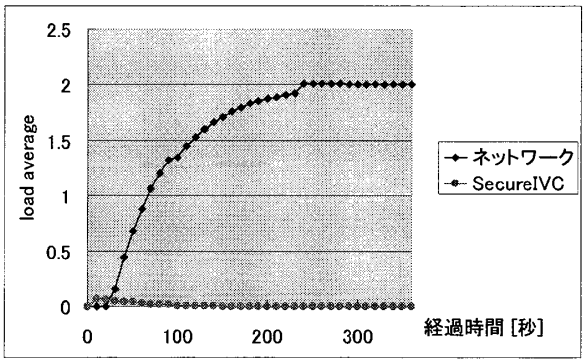

図 2 CPU 負荷測定結果

(2) データ通信性能

ある VM から他の VM に対し，1KB のデータの送受信を 2 万回処理するのにかかる時間を測定した（表 3）．ネットワークより SecureIVC の方が約 3.3 倍速く通信可能であることがわかる．

表 3 通信性能測定結果

ネットワーク	1.85 秒
SecureIVC	0.56 秒

5. おわりに

本 SecureIVC 機構を用いることで，VM 間におけるデータ通信時のアクセス制御にかかる負荷を送信側 VM のみに限定することができ，DoS 攻撃回避可能かつ高速な VM 間データ通信を行うことが可能となる．今後は，本機構を利用し，セキュリティが重視される組込みシステム等への VM 適用を実施していく予定である．

参考文献

[1] すべてわかる仮想化大全 VMware/Virtual Server，日経 BP 社（2006）．
[2] http://www.xen.org