平成 22 年度 卒業論文概要			
所　　属	機械情報工学科	指導教員	光來 健一
学生番号	07237053	学生氏名	中村 孝介
論文題目	KVM における仮想マシンを用いた IDS オフロードの実現		

1　はじめに

インターネットに接続された計算機への攻撃は年々増加している。攻撃者の侵入を検知する手段として侵入者検知システム (IDS) が用いられているが、攻撃者は IDS に検知されるのを防ぐために侵入後に IDS の無効化を試みる。このような IDS 自身への攻撃に対処するために、近年、仮想マシン (VM) を用いて IDS をオフロードするという手法が提案されている。IDS のオフロードとは、IDS を監視対象 VM の外で動作させ、別の VM からオフロード元の VM を監視する手法である。オフロードを行うことにより、監視対象の VM に侵入されても、IDS は別の VM で動作しているので攻撃されにくくなり、IDS のセキュリティを向上させることができる。

これまで、IDS のオフロードの研究は仮想化ソフトウェアの一つである Xen を使って行われてきた [1][2]。しかし、最近普及してきた KVM を使った IDS のオフロードはまだ研究されていない。KVM は急速に普及しつつあるため、IDS のオフロードを実現できるようにすることがセキュリティを向上させるために必要である。しかし、KVM のアーキテクチャは Xen とは大きく異なるため、Xen におけるオフロード手法を適用できるかどうかが不明であった。

本研究では KVM を用いて IDS のオフロードを実現するシステム KVMonitor を提案し、オフロードを行った際のリソース管理の問題を解決する。

2　Xen における IDS オフロード

Xen は図 1 のように特権をもった VM であるドメイン 0 と通常の VM であるドメイン U で構成されている。Xen を用いて IDS のオフロードを行う場合、IDS はドメイン 0 にオフロードされ、ドメイン U を監視する。ドメイン 0 は全てのドメインの管理を行う権限を持っているため、ドメイン 0 で動作する IDS はドメイン U を監視することができる。

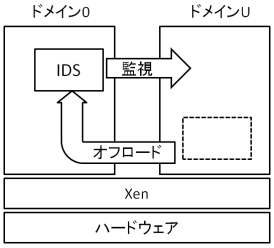

図 1　Xen のアーキテクチャ

オフロードした IDS は以下のようにしてドメイン U のリソースを監視する。ディスクについては、ドメイン 0 でドメイン U のディスクイメージをマウントすることにより、ドメイン U の仮想ディスクを監視することができる。マウントされた仮想ディスクに対して Tripwire などの IDS を用いてファイルの整合性のチェックを行う。メモリについては、ドメイン 0 からドメイン U の仮想アドレスを指定して目的の物理メモリをマップする機能を用いてドメイン U のメモリにアクセスする。マップしたメモリに対して、XenAccess などを用いてドメイン U 内の OS のプロセスやモジュールの一覧などを監視する。

IDS をオフロードすると VM の性能分離が難しくなるため、オフロードを考慮した CPU 割り当てのための Resource Cage[2] が提案されている。

3　KVMonitor

本研究では KVM を用いた IDS オフロードを実現するシステム KVMonitor を提案する。KVM は Linux カーネル自体を仮想化ソフトウェアとして利用する。これにより、Linux カーネルの豊富な機能を利用して VM の管理を行うことができる。図 2 のように、VM はホスト OS の 1 つのプロセスとして作成され、その管理には QEMU という CPU エミュレータを使用する。KVM では監視対象 VM の IDS をホスト OS にオフロードし、監視を行う。

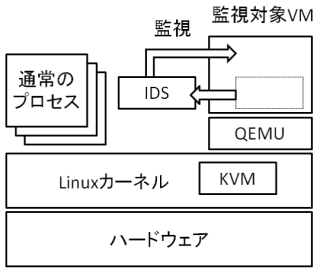

図 2　KVM のアーキテクチャ

3.1　ディスクの監視

IDS が VM のディスクを監視できるようにするために、VM のディスクをホスト OS にマウントする。Xen と違い、KVM はデフォルトで qcow2 フォーマットのディスクイメージを使用する。qcow2 はディスクに割り当てるサイズではなく、実際に使用するサイズでディスクイメージを作成するのでディスクスペースの節約ができる。qcow2 形式のディスクイメージは直接マウントすることができないので、qemu-nbd というツールを用いてマウントできる形式に動的に変換する。qemu-nbd は qcow2 形式のディスクイメージを単純なブロックデバイスとして見せるツールである。さらに、kpartx コマンドを使って、パーティション毎のブロックデバイスを作成し、このブロックデバイスをマウントすることによりゲスト OS のファ

イルシステムにアクセスする。

3.2 メモリの監視

IDS が VM 内部のメモリ上のデータを監視できるようにするために、VM のメモリを共有することでアクセスできるようにする。Xen では VM のメモリをマップして共有するという機構が提供されているが、KVM では VM を管理している QEMU しか VM のメモリにアクセスすることができない。そこで、監視対象 VM の起動時にオプションを指定することでファイルを物理メモリとして使わせるようにする。このファイルを置くディレクトリは hugetlbfs にし、Linux の hugepages 機構を使用することで性能への影響を減らす。

IDS はこのファイルを mmap システムコールを用いて自分自身のメモリ上にマップすることで、VM のメモリにアクセスする。従来の QEMU は他のプロセスにこのファイルを見られないようにファイルをオープンした後で削除していた。そこで QEMU に修正を加え、ファイルを削除しないようにして、IDS からアクセスできるようにする。

仮想アドレスを使って VM 内部のデータにアクセスできるようにするために、QEMU と通信して仮想アドレスを物理アドレスに変換する。VM 内部の OS カーネルの変数等は仮想アドレスだけが分かっているが、VM から IDS にマップするのは物理メモリであるため物理アドレスでしかアクセスできない。QEMU は仮想アドレスから物理アドレスの変換を行う機能を持っているため、外部から利用を可能にするために xaddr コマンドを追加した。IDS がアドレス変換を行う際には、QEMU Monitor Protocol (QMP) を使って QEMU と通信を行い、このコマンドを実行する。

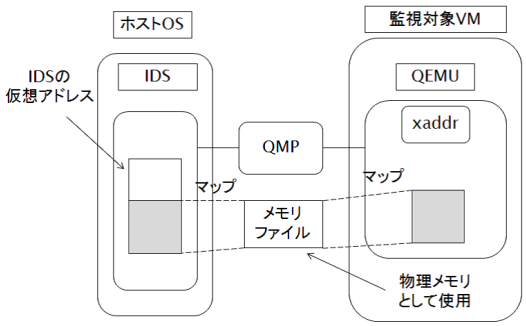

図 3　メモリの監視

3.3 CPU に関する性能分離

IDS をオフロードすることで生じる性能分離の問題を解決するために、Linux が提供している Cgroups 機能を用いる。Cgroups とは、複数のプロセスをグループ化して、そのグループ単位で資源管理を可能にする機能である。この機構を用いて、ホスト OS 上で監視対象 VM と IDS の 2 つのプロセスをグループ化し、このグループに対して CPU 使用率を設定することで CPU に関する性能分離を実現する。この性能分離は KVM において VM が 1 プロセスとして作成されているために容易に行うことができる。Xen では VM はプロセスではなく、Cgroups を使うこともできないため、VM スケジューラを変更することで実現していた。

Cgroups を利用するには cgroup ファイルシステムをマウントし、サブディレクトリを作ることでグループを作成する。このディレクトリにはグループ化するプロセスを指定したり CPU 使用率を設定したりするための設定ファイルが含まれる。これらの設定ファイルにグループ化するプロセスの ID と CPU 使用率の下限値を設定する。ただし、Xen の場合は上限値を設定していたため、CPU が 100% 使われていない時の挙動は異なる。

4 実験

Cgroups で CPU 使用率を指定し、時間毎の CPU 使用率の推移を確認する実験を行った。Cgroups を用いて監視対象 VM に 50% の CPU を割り当て、ホスト OS 側と監視対象 VM 側で CPU を 100% 使うプログラムを実行した。実験結果は図 4 のようになった。この結果から監視対象 VM の CPU 使用率はおおよそ 60% となり、割り当てた 50% の CPU 使用率が保証されていることからホスト OS のプロセスが監視対象 VM の CPU 使用率に影響を与えていないことが確認できた。

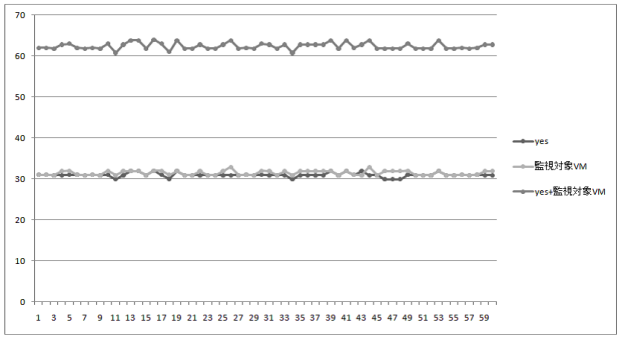

図 4　監視対象 VM の CPU 使用率

5 まとめ

本研究では KVM において IDS オフロードを実現するシステム KVMonitor を提案した。IDS がディスクの監視を行えるようにするために、監視対象 VM のディスクイメージをマウントできる形式に変換してからマウントする。QEMU にファイルを VM のメモリとして使わせ、そのファイルをマップすることでメモリの監視を行う。また、IDS をオフロードすることで CPU に関する性能分離が行えなくなるという問題を Linux の Cgroups を用いることで解決した。今後の課題はネットワークの監視を行えるようにし、メモリに関する性能分離も実現することである。

参考文献

[1] 飯田貴大, 光来健一 :仮想マシンを用いた既存ソフトウェアのオフロード手法 (2009)

[2] 新井昇鎬, 光来健一, 千葉滋 :仮想マシンを用いた IDS オフロードにおける CPU 資源管理 (2009)

[3] Chisnall, D., 渡邉了介訳: 仮想化技術 Xen-概念と内部構造, 毎日コミュニケーションズ (2008).

[4] 平初/森若和雄/鶴野龍一郎/まえだこうへい: KVM 徹底入門 Linux カーネル仮想化基盤構築ガイド, 株式会社翔泳社 (2010).