AMD SEVを用いてメモリが暗号化されたVMの安全な監視機構

能野 智玄[1]　光来 健一[1]

1. はじめに

近年，ユーザに仮想マシン（VM）を提供するIaaS型クラウドが普及している．クラウドでより重要なデータを扱うようになるにつれて，クラウド内の内部犯によってVMのメモリ上にある機密情報が盗まれるリスクが問題になってきている．この問題に対して，AMD製のCPUではVMのメモリを保護するために，SEVと呼ばれる透過的なメモリ暗号化が提供されている．SEVを用いることによって，クラウドの管理者でさえVMのメモリ上にある機密情報にアクセスすることはできなくなる．

しかし，VM内に侵入されてしまうとSEVによるメモリ暗号化による保護は機能しないため，VM内の機密情報を盗み見られてしまう恐れがある．そのため，侵入検知システム（IDS）を用いてVMの監視を行うことが必要である．VMへの侵入時にIDSが無力化されるのを防ぐために，IDSをVMの外で動作させてVMのメモリを監視するIDSオフロードと呼ばれる手法が用いられている．しかし，SEVを用いてVMのメモリが暗号化されると，IDSオフロードによってVMの監視を行うことができなくなる．

本稿では，SEVを用いてメモリが暗号化されたVMに対してIDSオフロードを実現するシステムSEVmonitorを提案する．

2. SEVmonitor

SEVmonitorは図1のように監視対象VMの内部でメモリデータを取得するためのエージェントを安全に動作させる．エージェントはVM内にインストールされるソフトウェアである．エージェントを導入することにより，VMのメモリが暗号化されていてもオフロードされたIDSがメモリデータを取得してOSデータを解析することができる．

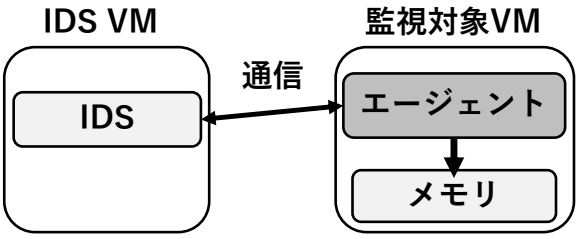


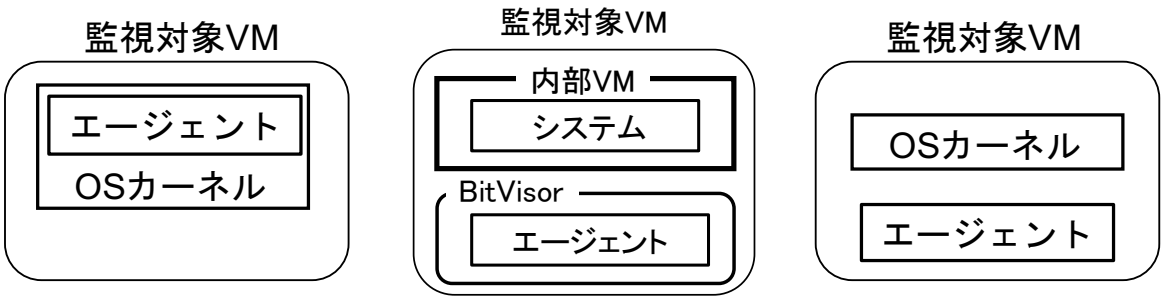
図 1: SEVmonitorのシステム構成

また，SEVmonitorはIDSもSEVを用いてメモリが暗号化されたIDS専用VM内で安全に実行する．これにより，IDSを攻撃することによってIDSが取得した機密情報を盗むことはできない．IDS VMではIDSのみを動作させるため，監視対象VMよりは侵入を防ぐのが容易である．

監視対象VM内に配置されるエージェントは，攻撃者がVMに侵入したとしても無効化されず，安全に動作し続ける必要がある．エージェントの配置は現在のところOS内，監視対象システム外，OS管理外の3つを考えている．図2(a)のように配置されるOS内のエージェントはOSの豊富な機能を利用できるが，OSが攻撃されると無効化される恐れがある．監視対象システム外のエージェントは図2(b)に示すように，BitVisorを用いて監視対象VM内にVMを作成してシステムを閉じ込めることにより，その外側で安全に動作させることができるが，ネストした仮想化のオーバヘッドが大きくなる．図2(c)のように配置されるOS管理外のエージェントはOS起動前に実行することで無効化されにくくすることができるが，OSの機能を用いずに実装するのが難しい．それぞれの配置にはトレードオフがあるため，状況に応じて使い分ける．

IDS VM内のIDSは監視対象VM内のエージェントと仮想ネットワークまたは共有メモリを用いて通信を行い，メモリデータを取得する．VM間の通信はクラウドの内部犯によって盗聴される恐れがあるため，送受信するデータは暗号化する．

[1] 九州工業大学
Kyushu Institute of Technology



(a) OS 内　(b) 監視対象システム外　(c) OS 管理外

図 2: エージェントの配置

3. 実験

OS 内のエージェントと監視対象システム外のエージェントを用いて，SEVmonitor の監視性能を調べた．まず，監視対象 VM から OS のバージョン情報を取得する IDS を実行した．この IDS は Linux カーネルの linux_banner 変数に格納されている文字列を取得して表示する．そのために，エージェントに対して要求を 1 回送信し，4KB のメモリデータを取得した．図 3 に示すように，TCP 通信を用いた場合はエージェントの配置によらず性能はほぼ同じになった．一方，共有メモリを用いると 1 ms 高速になった．TCP 通信を用いた場合でも仮想ネットワーク経由での通信となるため，共有メモリを用いた場合と比べてそれほど大きな差にはならなかった．

次に，監視対象 VM から OS のプロセス一覧を取得する IDS を実行した．この IDS は Linux カーネルの init_task 変数からプロセスリストをたどり，プロセスの ID と名前を取得して表示する．そのために，エージェントに対して要求を 119 回送信し，合計で 476KB のメモリデータを取得した．図 4 に示すように，OS 内のエージェントは共有メモリを用いると 40%高速になった．一方，監視対象システム外のエージェントが TCP 通信を用いる場合には 57%遅くなった．

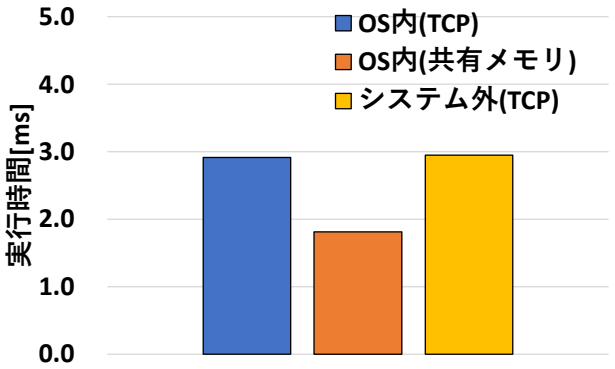


図 3: OS のバージョン情報取得

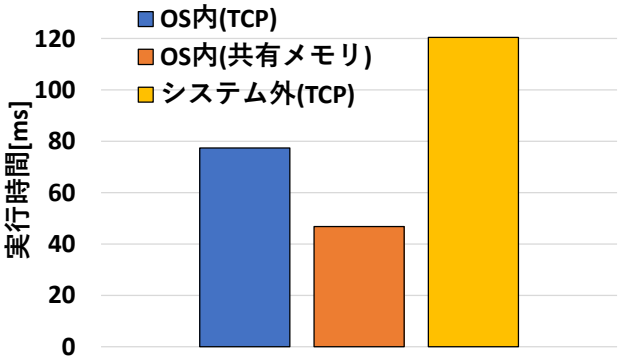


図 4: プロセス一覧の取得

4. まとめ

本稿では，SEV を用いてメモリが暗号化された VM に対して安全な IDS オフロードを実現するシステム SEVmonitor を提案した．今後の課題は，監視対象システム外のエージェントを用いる場合において共有メモリの実装を完成させることや，連続で通信した際のオーバヘッドを削減することである．また，OS 管理外のエージェントの実装方法についても検討を行う予定である．

謝辞

本研究の一部は，JST，CREST，JPMJCR21M4 の支援を受けたものである．また，本研究成果の一部は，国立研究開発法人情報通信研究機構の委託研究により得られたものである．