AMD SEVとeBPFを用いた安全で高速なリモートVM監視

上杉　貫太[1]　光来　健一[1]

1. はじめに

クラウド上の仮想マシン（VM）を利用するユーザが増えるにつれ，VM 内の機密情報がクラウドの内部犯によって盗聴されるリスクが問題となっている．そのため，AMD 製の CPU は VM のメモリを暗号化することで VM 外からの盗聴を防ぐ SEV と呼ばれる機能を提供している．SEV の暗号鍵は CPU によって管理されるため，クラウドの内部犯であってもメモリデータを復号できず機密情報の漏洩を防ぐことができる．一方，SEV で VM のメモリを暗号化していても，VM 内でのメモリアクセスに対しては復号が透過的に行われメモリを自由に読み書きすることができる．このように，SEV は VM 内の侵入者に対しては無力であるため，侵入検知システム（IDS）と併用する必要がある．しかし，VM が SEV で保護されていると VM の外にオフロードした IDS は VM のメモリ上の OS データを監視できなくなる．

そこで，VM の外にオフロードした IDS が VM 内で安全に動作するエージェントと通信することでメモリデータを取得し，OS データの監視を行う SEVmonitor [1] が提案されている．SEVmonitor を用いることにより，SEV で暗号化された VM の侵入検知を VM 外部から行うことができる．しかし，監視に用いるメモリデータは IDS が必要とした時に取得されるため，プロセスリストなどのようにポインタを用いる OS データの場合はポインタをたどるたびにエージェントとの通信が必要となり，監視性能が低下する．

本稿では，SEV で保護された VM に eBPF プログラムを送り込み，OS データを先読みして一括で取得するシステム eBPFmonitor を提案する．

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

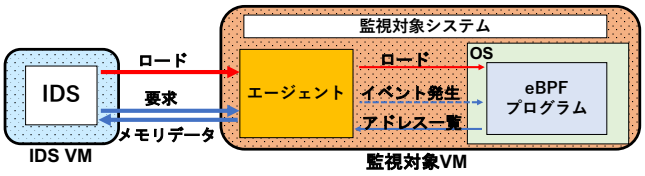

図 1　eBPFmonitor のシステム構成

2. eBPFmonitor

eBPFmonitor のシステム構成を図 1 に示す．eBPFmonitor は SEVmonitor と同様に，監視対象 VM 内で監視対象システムをコンテナに隔離し，その外側で安全にエージェントを動作させる．IDS はエージェント経由で監視対象 VM のメモリデータをページ単位で取得する．それに加えて，監視対象 VM 内で eBPF プログラムを実行することにより，監視に必要なメモリデータを一括で取得することができる．eBPF プログラムはポインタを用いる OS データを先読みし，一連の OS データを収集する．対象となる OS データの例としては，プロセスやカーネルモジュールのリスト，ネットワークソケットを管理するハッシュ表などが考えられる．このように，IDS からの 1 回の要求でエージェントが OS データを一括で返送することにより，通信のオーバヘッドを削減し，VM の監視を高速化する．

eBPFmonitor は OS データの収集に eBPF プログラムを用いることにより，安全性と柔軟性を両立させている．eBPF は性能などを監視するために用いられている Linux の機構であり，ネットワークやシステム動作のトレーシング，システムのセキュリティ対策などに使用されている．eBPF プログラムは OS 内にロードされ，ネットワークイベントやシステムコール，関数エントリ，カーネルのトレースポイントなどのイベントが発生した時に実行される．eBPF プログラムは OS の変更やカーネルモジュールの追加なしに OS 内に動的にロードして実行することができる．

そのため，VM の監視に必要な OS データを IDS ごとに異なった eBPF プログラムを用いて収集することで，OS データの柔軟かつ効率的な一括取得を行うことができる．また，eBPF プログラムのロード時に検証器によって OS の実行に影響する命令や無限ループが検知されるとロードに失敗する．そのため，関数やグローバル変数へのアクセスが制限されるだけのカーネルモジュールより安全に OS データの取得を行うことができる．

eBPFmonitor では，IDS が VM の監視を開始する前にあらかじめ eBPF プログラムを監視対象 VM にロードしておく．IDS からエージェントを介して監視対象 VM の OS に eBPF プログラムをロードし，その際に eBPF プログラムを呼び出すためのイベントを OS に設定する．その後，IDS が OS データを必要とした時には IDS からエージェントに OS データの一括取得の要求を送信し，要求を受信したエージェントは eBPF プログラムを実行するためのイベントを発生させる．実行された eBPF プログラムはポインタをたどり OS データが格納されているアドレスを収集する．エージェントは収集されたアドレスに対するメモリデータを取得し，順次 IDS に返送する．IDS は受信したメモリデータをキャッシュに保存して利用する．

3. 実験

eBPFmonitor を用いた OS データの一括取得にかかる時間を調べた．IDS と監視対象の VM を SEV で保護し, 監視対象 VM のユーザ空間にエージェントを配置した．IDS は起動時にエージェントとのネットワーク接続を確立し，エージェント経由で eBPF プログラムをロードする．その後も接続を維持し，定期的に OS データを取得して監視を行う．そのため，エージェントに OS データの一括取得要求を送信してから，IDS がメモリデータを取得し，プロセス情報を出力するまでの時間を測定した．比較として，ポインタをたどるたびに通信を行う SEVmonitor でも取得時間の測定を行った．

eBPFmonitor と SEVmonitor を用いて監視対象 VM のプロセス一覧とカーネルモジュール一覧を取得する時間を測定した．測定結果を図 2 に示す．実行している全てのプロセスを取得する場合, eBPFmonitor と SEVmonitor ではエージェントの分だけプロセス数が異なるため，取得するプロセスを 100 個に統一して取得時間の比較を行った．図 2 より，eBPFmonitor は SEVmonitor より 2.5 倍高速にプロセス一覧を取得できることが分かった．SEVmonitor において IDS とエージェントは 101 回の往復通信を行ったのに対し，eBPFmonitor の通信は要求が 1 回，応答が 101 回となった．そのため，通信のオーバヘッドが削減され取得時間が減少した．一方, カーネルモジュール一覧の取得はどちらの手法でも監視対象 VM で実行中のカーネルモジュールは変わらないため，全カーネルモジュールの情報を取得する時間を測定した．図 2 より, eBPFmonitor は SEVmonitor より 2.6 倍高速にカーネルモジュール一覧を取得できることが分かった．プロセス一覧の取得時と比べて通信データ量が少ないため，eBPFmonitor と SEVmonitor の取得時間が減少したと考えられる．

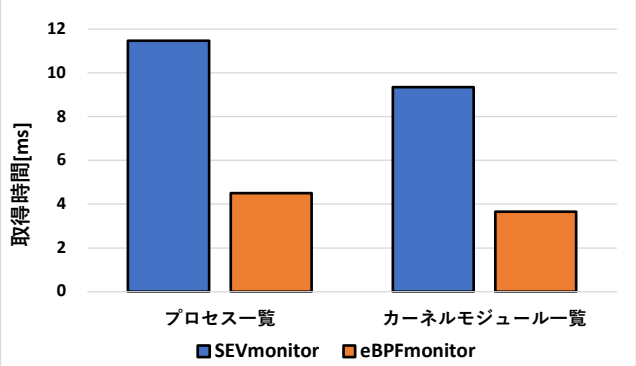

図 2 OS データの取得時間

4. まとめ

本稿では，SEV で保護された VM に送り込んだ eBPF プログラムを用いて OS データを先読みして一括で取得するシステム eBPFmonitor を提案した．実験結果より，IDS から監視対象 VM への 1 回の要求で，必要とするすべてのメモリデータを一括で取得することで, 取得時間を大幅に高速化できることが分かった．今後の課題は，IDS が必要とするデータのみを一括で取得できるようにし，通信データ量の削減によるさらなる高速化を行うことである．また，エージェントを OS 内やハイパーバイザ内で動作させられるようにすることでエージェントの保護を強化することも検討している．

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（05501）による成果を含む．

参考文献

[1] 能野智玄，光来健一：AMD SEV で保護された VM の隔離エージェントを用いた安全な監視，*CSS 2022* (2022).