ネストしたVMをAMD SEVで保護することによる安全な通信の追跡・制御

安東 尚哉[1]　光来 健一[1]

1. はじめに

近年，不正アクセスや設定ミスによるパブリッククラウドからのパーソナルデータの漏洩が問題となっている．その原因の一つとして，クラウドが年々，複雑なサービスを提供するようになっていることが挙げられる．マイクロサービスやマルチクラウドを用いて複数のサービスを連携させることが多くなっているため，パーソナルデータがクラウド内，さらにはクラウド間で様々なサービスに転送されるようになっている．

クラウドが内部でどのようにデータを扱っているかの詳細についてはユーザに公開されていないため，パーソナルデータがどのクラウドサービスに転送されているのかもわからないことが多い．そのため，ユーザは自分のパーソナルデータが意図しないサービスに転送されたり，漏洩したりしていてもそのことを把握することができない．この問題を解決するためにはユーザが自分のデータの流れを追跡・制御することができるプライバシ制御機構がクラウド内に必要となる．このような機能があれば，自分のデータの流通範囲や保存先を把握することができ，データの流れを制御することで情報漏洩を未然に防ぐことができる．しかし，ユーザからはクラウドが提供するプライバシ制御機構を完全には信頼することができない．

本稿では，AMD SEV を用いてクラウド内のデータ流を安全に追跡・制御するシステム SEV-tracker を提案する．

2. SEV-tracker

SEV-tracker のシステム構成は図 1 のようになる．クラウド内でプライバシ制御機構が正しく動作することを保証するために，SEV-tracker はネストした仮想化を用いて，ユーザが送り込んだハイパーバイザをクラウドの VM 内で実行する．このような構成にすることにより，クラウド VM がネストした仮想化と SEV をサポートしていれば，既存のクラウドに適用することができる．送り込んだユーザ・ハイパーバイザ上でユーザ VM を作成し，ユーザ VM 内でクラウドサービスを動作させる．これにより，クラウドサービスの通信やディスクアクセスなどをユーザ・ハイパーバイザ内のプライバシ制御機構が捕捉して追跡・制御を行うことができる．

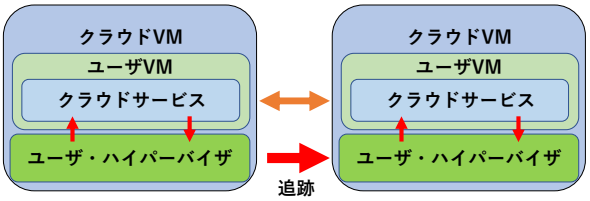

図 1　SEV-tracker のシステム構成

SEV-tracker は SEV を用いて相互保護を実現する．SEV は AMD 製 CPU が提供する VM のメモリ暗号化機能であり，VM がメモリにデータを書き込む際に暗号化を行い，読み込む際に復号化を行う．メモリ暗号化のための鍵は AMD セキュアプロセッサで生成・管理されるため，ハイパーバイザから VM を保護することができる．ユーザ・ハイパーバイザが動作するクラウド VM のメモリを SEV で保護することにより，クラウドから攻撃されたとしてもデータの改竄や窃取を防ぐことができる．また，クラウドサービスが動作するユーザ VM のメモリも SEV で保護することで，ユーザ・ハイパーバイザからの攻撃を防ぐことができる．

SEV-tracker はネストした仮想化を用いるため，クラウドサービスの実行オーバヘッドが大きくなる．このオーバヘッドを削減するために，SEV-tracker はユーザ・ハイパーバイザとしてできるだけ軽量なハイパーバイザを用いる．通常，ハイパーバイザは複数の VM をサポートしてい

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

るが，ユーザ・ハイパーバイザは一つのユーザ VM のみをサポートすれば十分である．複数の VM を動作させるのに必要な機能を省くことによって，ハイパーバイザを軽量化し，オーバヘッドを削減する．また，ハイパーバイザはデバイスを仮想化して VM に提供するが，ユーザ・ハイパーバイザはデータの追跡・制御を行う必要のないデバイスは仮想化する必要がない．パススルー機能を用いてクラウド VM の仮想デバイスをユーザ VM にそのまま見せることで軽量化を実現する．

さらに，ユーザ VM 内のクラウドサービスには軽量なライブラリ OS をゲスト OS として使わせる．ライブラリ OS とは OS の機能をライブラリとして提供し，アプリケーションにリンクして利用することができる OS である．アプリケーションをコンパイルする際に必要な機能のみをリンクすることで汎用 OS を使う場合よりも高速に動作し，メモリ使用量を抑えることができる．また，ライブラリ OS は最小限の初期化しか行わないため，クラウドサービスの起動を速くすることもできる．

SEV-tracker はクラウドサービスのすべての通信を監視することにより，データ流の追跡を可能にする．ユーザ VM 内のクラウドサービスが送受信を行うたびに，ユーザ・ハイパーバイザ内のプライバシ制御機構がパケットを捕捉して解析し，通信情報を記録する．ユーザがユーザ・ハイパーバイザに要求を送ると，その上で実行されているクラウドサービスの通信履歴を返す．さらに通信先でもユーザ・ハイパーバイザが動作している場合には，プライバシ制御機構から再帰的に要求を送ってそのクラウドサービスの通信履歴を取得する．この時，クラウドサービスの通信と同じ経路が使われるため，すべてのユーザ・ハイパーバイザと通信することができる．

3. 実験

SEV-tracker におけるクラウドサービスの性能を調べるために，iperf3 を用いて通信性能を測定した．KVM 上にクラウド VM を作成し，その中でユーザ・ハイパーバイザとして BitVisor を動作させた．また，BitVisor 上にユーザ VM を作成し，その中で Linux を動作させた．本実験では SEV を用いずに VM を作成して実験を行った．比較として，ネストを行わずにクラウド VM 内で Linux を動作させた場合についても測定した．測定には 10GbE のネットワークを使用し，クラウド VM の仮想 NIC には e1000 を使用した．

対象システムが外部ホストにデータを送信した場合と外部ホストからデータを受信した場合の測定結果を図 2 に示す．BitVisor を用いてネストした場合，ネストを行わなかった場合に比べて性能が 41〜56%に低下することが確認できた．これは，ネストしたことによる iperf3 の実行オーバヘッドのせいだと考えられる．また，BitVisor で ippass

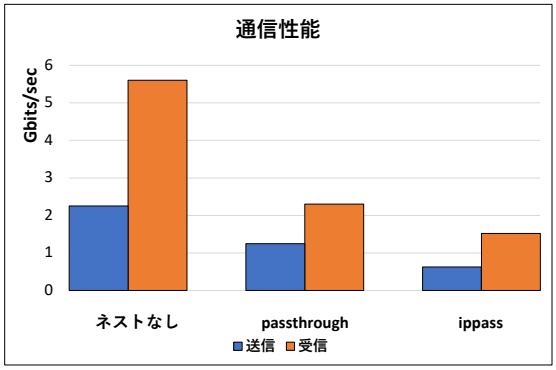

図 2 クラウドサービスの通信性能

モジュールを使用してパケットを捕捉すると，さらに性能が 51〜66%に低下することも確認できた．ippass モジュールを用いると NIC の仮想化を行うため，ネットワーク処理のオーバヘッドが大きくなってしまうことが原因であると考えられる．

今回は BitVisor はクラウド VM の e1000 を使用したが，BitVisor に virtio-net ドライバを実装すれば性能の改善が見込めると考えている．

4. まとめ

本稿では，AMD SEV を用いてユーザがクラウド内のデータ流を安全に追跡・制御することを可能にするシステム SEV-tracker を提案した．今後の課題は，様々な Unikraft アプリケーションを動作させてデータ流の追跡を行えるようにすることである．まずは，Unikraft 上で通信を行って通信履歴を正しく取得できることを確認する．また，データ流の追跡だけでなく制御も行えるようにする予定である．さらに，SEV を用いて BitVisor を起動できるようにし，BitVisor 上で SEV を用いて Unikraft を起動できるようにすることも必要である．

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．