AMD SEV-SNPによるネストしたVMの保護

瀧口 和樹[1]　光来 健一[1]

1. はじめに

ユーザに仮想マシン (VM) を提供するクラウドが様々な用途に活用されている．それに伴い，機密性の高い情報がクラウドで扱われるようになり，クラウドの内部犯などから機密情報を盗まれる危険性が増している．そのため，AMD EPYC プロセッサでは Secure Encrypted Virtualization (SEV) と呼ばれる VM のセキュリティ機構が提供されている．SEV は VM のメモリを透過的に暗号化し，VM の内部でのみ復号可能にする．そのため，VM 外部のハイパーバイザ等によってメモリ内部の機密情報が盗聴されるのを防ぐことができる．第 2 世代以降では SEV-Encrypted State (SEV-ES) と呼ばれる拡張が提供されており，メモリに加えてレジスタの状態も暗号化することができる．第 3 世代以降では SEV-Secure Nested Paging (SEV-SNP) と呼ばれる拡張が提供されており，さらにセキュリティが強化されている．Amazon Web Services，Google Cloud，Microsoft Azure などで SEV を適用した Confidential VM が提供されている．

一方，クラウドにおいてネストした仮想化を用いた様々なシステムが提案されている．ネストした仮想化は VM 内で VM を動作させる技術であり，本稿ではクラウドが提供する外側の VM を L1 VM，その中で動作する VM を L2 VM と呼ぶ．例えば，クラウドの L1 VM をホストとして用いることにより仮想クラウドを提供することができる．ネストした仮想化を用いるシステムに L2 VM にも SEV と SEV-ES を適用することを可能にするために，我々は Nested SEV と Nested SEV-ES を提案してきた [1, 2]．

本稿では，SEV-SNP を適用した L1 VM の中で SEV-SNP を適用した L2 VM を動作させることを可能にする Nested SEV-SNP を提案する．

[1] 九州工業大学

2. Nested SEV-SNP

SEV は VM のメモリの暗号化を行うための機能である．暗号鍵の生成と管理や，VM 用の UEFI といったファームウェアや，ディスクの秘密鍵を VM が実行する前に暗号化する必要があるため，これらをプロセッサの内部に存在する AMD セキュアプロセッサ (AMD-SP) が行う．ハイパーバイザが書き換えても検知できるように DH 鍵交換や HMAC などが用いられる．VM の実行中にはメモリコントローラがメモリの暗号化を行う．SEV-ES はレジスタも暗号化する．メモリやレジスタを暗号化しただけでは VM のメモリの内容の破壊や，内容の入れ替えが可能であるため，暗号化されたメモリの中身を取得する攻撃やコードインジェクション攻撃が提案されている．

SEV-SNP ではメモリとレジスタの暗号化に加えて，メモリの整合性の保護などを行うために Reverse Map Table (RMP) と呼ばれるテーブルが新たに導入された．RMP は物理マシンに一つ，エントリは物理ページ単位で存在し，ゲストの物理アドレスや VM の暗号鍵を識別する ASID，validate 済みかどうかのフラグなどが格納されている．RMP 自体は物理メモリ上に存在し，読み出すことは可能であるが，書き換えることは専用の命令か AMD-SP を介すことでのみ可能である．RMP のエントリを書き換えると validate 済みのフラグは 0 に設定され，ゲストがアクセスすると例外が発生する．ゲストが専用の命令を使うとフラグを 1 に設定することができ，ハイパーバイザが RMP を書き換えても検知できる仕組みとなっている．RMP のエントリにはゲスト物理アドレスが存在し，アドレスが一致しない場合動作が停止するため VM のメモリを入れ替えることはできない．また，VM の暗号鍵を識別する ASID も含んでいるため，同一のゲスト物理アドレスのメモリを VM 間で入れ替えることも不可能である．

我々は Linux 6.1 と QEMU 7.1 に SEV-SNP 対応パッチを当てたもの [3] に対して，暗号鍵が同一の構成と暗号鍵

が異なる構成の 2 種類を実装した．暗号鍵が異なる構成では，L0 KVM と L0 QEMU の仮想 AMD-SP 実装を拡張し，SEV-SNP に対応を行った．暗号鍵が同一の構成では，L1 ハイパーバイザから L2 VM のメモリに自由にアクセスできないため，Nested SEV や Nested SEV-ES であった利点が存在しない．これは，SEV-SNP ではゲスト物理アドレスが一致しなければアクセスできないためである．そのうえ，L2 VM 間でのメモリの入れ替えが可能であるため，セキュリティに課題がある．

L1 KVM が RMP を L2 VM のために書き換える必要があるため，そのインターフェースの実装を L0 KVM に行い，L1 KVM はそのインターフェースを利用して RMP の書き換えを行うようにした．そのほかにも L0 KVM，L1 KVM，L1 Linux カーネルを変更して適切に RMP の制御を行うようにした．

3. 実験

VM 内の Apache HTTP Server 2.4.48 に bombardier を用いて VM 内から並列にリクエストを送信した．Web サーバのリクエスト処理性能を図 1 に示す．L1 VM には 12 個の仮想 CPU と 16 GiB のメモリを割り当て，L0 で使われていない物理 CPU を割り当てた．L2 VM には 6 個の仮想 CPU と 8 GiB のメモリを割り当て，L1 VM で使われていない仮想 CPU を割り当てた．ネストした仮想化を用いない場合は L1 VM に 6 個の仮想 CPU と 8 GiB のメモリを割り当てた．なお，RMP チェックのオーバーヘッドは SEV-SNP 以外の VM の書き込みアクセスとページテーブルのメモリアクセスにも及ぶため，ベンチマークによっては RMP を有効にしたかどうかで性能に差が出ることがある．図 1 の測定では SEV-SNP の測定以外では RMP を無効にしている．ネストしていない場合でも非 SEV と比べて 1 並列と 1000 並列でそれぞれ 46.2 %，32.0 %と大きく低下しており，SEV-SNP のオーバーヘッドや RMP チェックのオーバーヘッドによるものと考えられる．ネストしている場合，異なる暗号鍵の構成では非 SEV と比べて 1 並列と 1000 並列でそれぞれ 58.7 %，31.0 %と低下している．同一の暗号鍵の構成でも 55.3 %，32.3 %の低下し，異なる暗号鍵の構成と同様だった．

4. まとめ

本稿では，SEV-SNP を適用した L1 VM の中で SEV-SNP を適用した L2 VM を動作させることを可能にする Nested SEV-SNP を提案した．

今後の課題は，SEV-SNP が提供する他の機能への対応や，Nested SEV で対応していた Xen や BitVisor を L1 ハイパーバイザとして使えるようにすることである．

謝辞

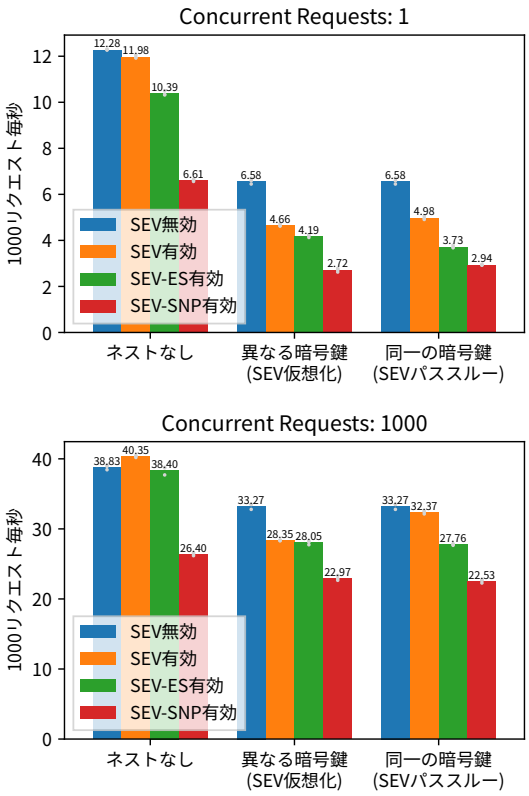

図 1 HTTP サーバのリクエスト処理性能

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] 瀧口和樹，光来健一：Nested SEV：ネストした仮想化への AMD SEV の適用，第 34 回コンピュータシステム・シンポジウム (2022).
[2] 瀧口和樹，光来健一：AMD SEV-ES によるネストした VM の保護，第 159 回 OS 研究会 (2023).
[3] Michael Roth: [PATCH RFC v8 00/56] Add AMD Secure Nested Paging (SEV-SNP) Hypervisor Support, https://lore.kernel.org/lkml/20230220183847.59159-1-michael.roth@amd.com/.