令和 2 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	17237049	学生氏名	能野 智玄
論文題目	AMD SEV を用いて暗号化された仮想マシンの監視機構		

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．それに伴い，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増大している．この問題に対して，AMD 製 CPU では VM のメモリデータを保護するために SEV と呼ばれる透過的なメモリ暗号化機構が提供されており，内部犯であっても VM のメモリを盗聴することができなくなっている．一方，VM 内に侵入されると SEV によるメモリ暗号化では機密情報が保護できないため，侵入検知システム（IDS）を用いて VM を監視する必要がある．侵入後に IDS が無力化されるのを防ぐには，IDS を VM の外で動作させてメモリ上の OS データを監視する IDS オフロード手法が有効である．しかし，VM の外にオフロードされた IDS は SEV を用いて暗号化された VM のメモリを監視することができない．

本研究では，SEV を用いてメモリが暗号化された VM の内部でエージェントを安全に動作させることにより IDS オフロードを実現するシステム SEVmonitor を提案する．

2　AMD SEV を用いて暗号化された VM の監視

クラウド内部には悪意のある管理者などの内部犯がいる可能性が指摘されている．内部犯に VM のメモリを盗聴されると VM 内に格納されているユーザの機密情報が盗まれる恐れがある．このような問題を解決するために，AMD SEV と呼ばれる CPU のセキュリティ機構が提供されるようになっており，VM のメモリを透過的に暗号化することができる．AMD SEV は CPU の中で VM ごとに異なる暗号鍵を用いてメモリ暗号化を行うため，VM の管理を行うことができる内部犯であっても VM のメモリを盗聴することはできない．

一方，VM のメモリを暗号化していてもネットワーク経由で VM 内に侵入されると，メモリ上の機密情報が盗聴される可能性がある．SEV は VM 外部からのアクセスに対してしかメモリを保護できないためである．そこで，従来と同様に IDS を用いて VM への侵入を検知する必要がある．しかし，VM に侵入された際に IDS が無力化されるとそれ以降の攻撃が検知できなくなる．このような攻撃を防ぐために，VM の外で IDS を動作させる IDS オフロードと呼ばれる手法が提案されており，VM 内に侵入されてもオフロードされた IDS は攻撃を検知し続けることができる．

しかし，SEV を適用した VM に対して IDS オフロードを行うには問題が二つある．第一に，VM のメモリが暗号化されているため，図 1 のようにオフロードされた IDS は VM のメモリ上にある OS データを解析して侵入を検知することができない．SEV からは IDS と攻撃者を区別することができないため，IDS にも暗号化された VM のメモリを復号する手段がない．第二に，オフロードされた IDS が VM を監視できたとしても，IDS 経由で機密情報が漏洩する恐れがある．IDS は VM から機密情報を含む可能性のあるデータを取得するため，クラウドの内部犯は IDS を攻撃することにより機密情報の一部を取得できる．

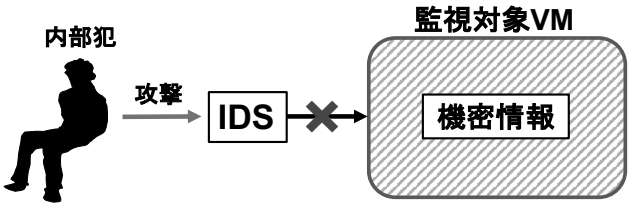


図 1: SEV で暗号化された VM に対する IDS オフロード

3　SEVmonitor

本研究では，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 を提案する．IDS オフロードを行う際の一つ目の問題を解決するために，SEVmonitor は図 2 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かす．エージェントは VM 内にインストールされる小さなソフトウェアであり，取得した VM のメモリデータを IDS に送信する．また，二つ目の問題を解決するために，SEVmonitor はオフロードした IDS も SEV を用いて暗号化された別の VM 内で安全に実行する．これにより，クラウドの内部犯であっても IDS を攻撃して機密情報を取得することができなくなる．

3.1　エージェントの配置

監視対象 VM 内に配置されるエージェントは VM に侵入された際に無効化されないようにする必要がある．そのため，どのようにエージェントを配置するかが重要になるが，エージェントの配置には様々なトレードオフがある．まず，図 3 (a) のように OS カーネル内にエージェントを配置することが考えられる．この配置は OS の豊富な機能を用いてエージェント

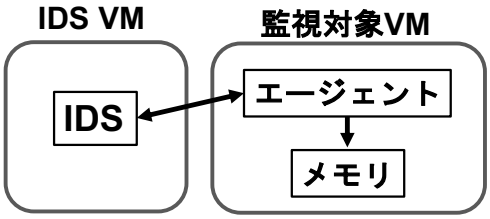
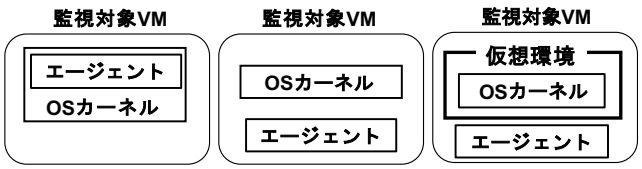


図 2: SEVmonitor のシステム構成



(a) 配置 1　(b) 配置 2　(c) 配置 3

図 3: エージェントの様々な配置

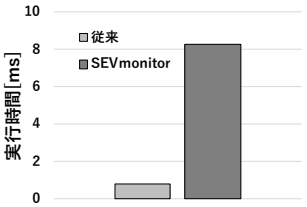


図 4: VM 内のバージョン情報の取得時間

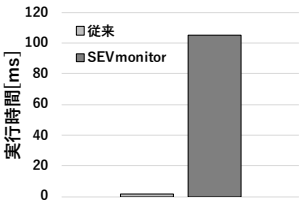


図 5: VM 内のプロセス情報の取得時間

の実装を行えるという利点があるが，OS カーネル内に脆弱性があるとエージェントが無効化される恐れがある．図 3 (b) のように，OS が起動する前にエージェントを実行することにより OS から隠すことも考えられる．この配置は OS カーネルが攻撃を受けたとしてもエージェントが見つかりにくいという利点があるが，OS の機能を用いずにエージェントを実装するのが難しい．そこで，図 3 (c) のように VM 内にさらに仮想環境を作成してその中に監視対象システムを閉じ込め，その外側にエージェントを配置することが考えられる．この配置には OS カーネルが攻撃されたとしてもエージェントを無効化されることがないという利点があるが，二重に仮想化を行うオーバヘッドが大きい．

本研究では，実装が容易であることから図 3 (a) のように OS カーネル内にエージェントを設置する．そのため, OS カーネルには脆弱性がないことを仮定する．エージェントはカーネルスレッドとして実行され，IDS が必要とするメモリデータに仮想アドレスのままでアクセスすることができる．そのため，従来の IDS オフロードのように物理アドレスへのアドレス変換をソフトウェアで行う必要がない．

3.2 IDS とエージェントの安全な通信

オフロードされた IDS と監視対象 VM 内のエージェントは VM 間に作成される仮想ネットワークを用いて通信を行う．エージェントは OS カーネル内の専用の関数を用いて IDS からのネットワーク接続を待ち受ける．エージェントの応答性を高めるために，接続要求が来るまで完全に停止するのではなく，定期的に接続要求をチェックする．そして，IDS からの接続要求が来ていればネットワーク接続を確立する．IDS が監視対象 VM のメモリ上にある OS データを取得する際には，OS データの仮想アドレスをエージェントに送信する．エージェントは接続要求と同様に，定期的にデータ要求をチェックし，要求が来ていれば受信処理を行う．そして，受信した仮想アドレスに対応するメモリデータを取得し，IDS に返送する．

仮想ネットワークはクラウドの内部犯が盗聴できるため，SEVmonitor は仮想ネットワークを流れるデータを暗号化することで機密情報の漏洩を防ぐ．IDS は仮想アドレスを暗号化してエージェントに送信し，エージェントは受信したデータを復号して仮想アドレスを取り出す．暗号化に用いる AES は 16 バイト単位でしか暗号化が行えないため，8 バイトの仮想アドレスに 8 バイトの余分なデータを付加して暗号化する．また，エージェントは 4KB 単位でメモリデータを暗号化して IDS に送信し，IDS は受信したデータを復号してメモリデータに含まれる OS データを取得する．IDS は SEV によって保護された VM 内で実行されるため，復号した OS データや復号鍵をクラウドの内部犯に盗聴されることはない．

3.3 OS データの解析

SEVmonitor は LLView [1] を用いて VM のメモリ上の OS データを解析する．LLView は Linux カーネルのソースコードを用いて透過的に VM 内の OS データにアクセスするためのフレームワークである．IDS をコンパイルして生成された中間表現に対して，メモリからデータを読み込むために用いられるロード命令を変換する．それにより，エージェントとの通信によって取得したメモリデータを読み込ませるようにする．性能を向上させるために，取得したメモリデータは IDS がキャッシュする．

4 実験

SEVmonitor を用いて VM の暗号化されたメモリから OS データを取得できることの確認および，取得時間の測定を行った．比較のために，KVMonitor [2] を用いてメモリが暗号化されていない VM から OS データを取得する時間も測定した．実験に用いたマシンの CPU は AMD EPYC 7262，メモリは 128GB であった．仮想化ソフトウェアには KVM を用い，IDS VM と監視対象 VM にはそれぞれ仮想 CPU を 2 個，メモリを 2GB 割り当てた．OS は両方とも Linux 5.4 を用いた．

まず，VM 内の OS のバージョン情報が取得できることを確認した．バージョン情報の取得にかかった時間は図 4 のようになり，従来手法の 10 倍に増加した．これはエージェントとの通信にかかるオーバヘッドが原因と考えられる．次に，VM 内のプロセス情報の一覧を取得できることを確認した．プロセス情報の取得にかかった時間は図 5 のようになり，従来手法の 56 倍となった．これはエージェントとの通信回数が 119 回行われたためだと考えられる．

5 まとめ

本研究では, AMD SEV を用いてメモリが暗号化された VM の内部でエージェントを安全に動作させることにより IDS オフロードを実現する SEVmonitor を提案した．SEVmonitor では，仮想ネットワークを用いて暗号化されたデータをやりとりすることで，エージェント経由で安全にメモリデータを取得する．今後の課題は，共有メモリを用いて通信のオーバヘッドを削減することや，他のエージェント配置を実装することである．

参考文献

[1] Y. Ozaki et al. Detecting System Failures with GPUs and LLVM. APSys 2019.

[2] K. Nakamura et al. Efficient VM Introspection in KVM and Performance Comparison with Xen. PRDC 2014.