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	17237026	学生氏名	古賀 吉道
論文題目	Intel SGX と SMM を組み合わせた侵入検知システムの安全な実行		

1　はじめに

近年，インターネットに接続されたシステムへの攻撃が数多く報告されている．攻撃の糸口となるシステムの脆弱性を完全に取り除くのは困難であるため，侵入検知システム（IDS）を用いてシステムを監視し，システムが攻撃を受けた際には管理者に報告する必要がある．しかし，システムの状態を監視して異常を検知するホストベース IDS は監視対象ホスト上で動作するため，安全に実行するのは難しい．例えば，システムが攻撃を受けた後はそのシステムから正しい情報を取得できるとは限らないため、攻撃を正確に検知できない恐れがある．また，IDS が攻撃を受けると無力化されてしまい，それ以降の攻撃を検知できなくなる．これまでに汎用 CPU の機能を用いて IDS を安全に実行する様々な手法が提案されてきたが，安全性や性能の面で十分なものではなかった．

本研究では，Intel 製 CPU のセキュリティ機能である SGX とシステムマネジメントモード（SMM）を組み合わせることで，より安全に IDS を実行することが可能なシステム SSdetector を提案する．

2　汎用 CPU の機能を用いた IDS の安全な実行

ホストベース IDS を安全に実行するための要件として，以下の二つが挙げられる．第一に，IDS は監視対象システムの機能を用いずに攻撃を検知することができる必要がある．システムが攻撃を受けた後は IDS がそのシステムから正しい情報を取得できる保証はないため，攻撃を検知できなくなる恐れがある．第二に，攻撃者がシステムに侵入したとしても IDS は攻撃を受けないようにすることが必要である．IDS を無力化されるとそれ以降の攻撃が検知できなくなる．

このような要件を満たすために，これまでに汎用 CPU の機能を利用して IDS を安全に実行する様々な手法が提案されてきた。例えば，Intel 製 CPU の動作モードの 1 つである SMM を用いた手法が挙げられる．SMM はハードウェアに近い BIOS 内でシステムからのアクセスができない独立した実行環境を提供するため，安全に IDS を実行することができる．IDS はメモリ上にあるシステムのデータを解析することで攻撃を検知する．しかし，SMM でのプログラム実行は低速であり，実行中はシステムが停止するという欠点がある．SMM でのプログラムの実行を最小限に抑えるためにメモリデータを外部ホストに送信する手法も提案されている [1] が，外部ホストで動く IDS の安全性が課題となる．

近年，Intel 製 CPU に搭載されている SGX によって提供されるエンクレイヴ内で IDS を安全に実行する手法も提案されている．エンクレイヴは CPU によってメモリが暗号化された保護領域であり，エンクレイヴの外部から IDS の盗聴や改ざんを行うことはできない．しかし，エンクレイヴはアプリケーション内に作成されるため，システムのメモリデータを安全に取得するのは難しい．そのため，システムを仮想マシン（VM）上で実行してその下で動作するハイパーバイザ経由で監視する必要があり [2]，仮想化を用いないシステムには適用ができない．その上，比較的，大きなソフトウェアであるハイパーバイザを信頼しなければならない．

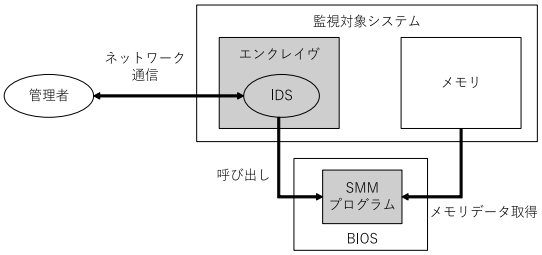

図 1　SSdetector のシステム構成

3　SSdetector

本研究では，SGX と SMM を組み合わせることにより，IDS をより安全に実行できるようにするシステム SSdetector を提案する．SSdetector は図 1 のように SGX のエンクレイヴ内で IDS を実行し，SMM で動作するプログラムがシステムのメモリデータの取得のみを行う．そのため，SMM による実行速度の低下を最小限に抑えることができる．また，システムを仮想化する必要もない．SSdetector では CPU および，ハイパーバイザよりはるかに攻撃が難しい BIOS 内の SMM プログラムのみを信頼する．一方で，IDS を実行する OS などの実行環境は信頼しない．なお，システム管理者は定期的に IDS と通信して，IDS の正常動作や検知結果の確認を行う．

3.1　エンクレイヴからのメモリデータの取得

SSdetector では，エンクレイヴ内の IDS は OS のメモリデータを解析することによりシステムの監視を行う．IDS が OS データを必要とした際には，図 2 のように SGX の機能である OCALL を用いることで，エンクレイヴの外部で動作する SSdetector ランタイムを安全に呼び出す．ランタイムを呼び出すのは SGX の仕様上，エンクレイヴ内から直接 SMM プログラムを呼び出すことができないためである．呼び出したランタイムは特定の I/O ポートに書き込むことによって，最高の優先度を持った外部割込みである SMI を発生させる．発生させた SMI により，SMM プログラム内の SMI ハンドラが呼び出され，指定したメモリデータがバッファに格納される．エンクレイヴ内では取得したメモリデータをキャッシュする

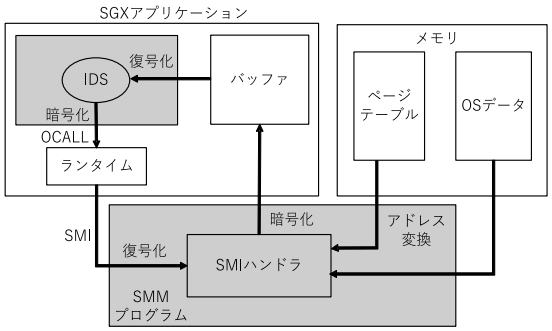

図 2　IDS からのメモリデータの取得

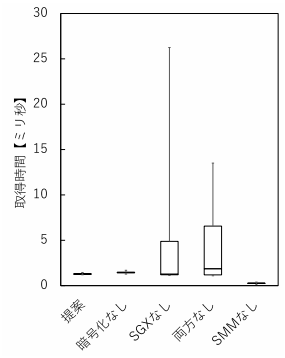

図 3　OS データの取得時間

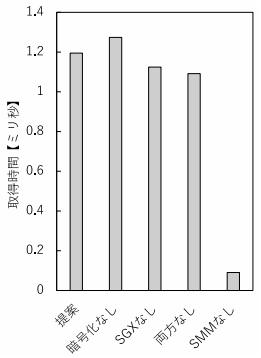

図 4　取得時間の最小値

ことにより，SMM プログラムの呼び出し回数を削減する．

メモリデータを取得する際にランタイム等に情報が漏洩するのを防ぐために，SSdetector はエンクレイヴと SMM プログラム間でやりとりするデータの暗号化を行う．エンクレイヴ内の IDS が SMM プログラムを呼び出す際には，OS データのアドレスとバッファのアドレスを暗号化する．それを SMM プログラムが復号してアドレスを取り出す．一方，SMM プログラムから IDS にメモリデータを返す際には，渡されたバッファに暗号化したメモリデータを格納する．その後，エンクレイヴ内の IDS が復号してメモリデータを取り出す．

3.2　SMM プログラムでのメモリアクセス

SMM プログラムがメモリにアクセスする際には，IDS から渡された仮想アドレスを物理アドレスに変換する必要がある．IDS には監視対象システムの OS データやメモリデータを受け取るバッファの仮想アドレスしか分からないのに対し，SMM では物理アドレスでしかメモリにアクセスすることができないためである．そこで，SMM プログラムはまず，CPU の CR3 レジスタの値を取得して OS メモリ上のページテーブルを特定する．ページテーブルはシステムがアドレス変換を行うために用いる表である．SMM プログラムでも同様にして，OS のページテーブルを用いて OS データの仮想アドレスを変換し，SGX アプリケーションのページテーブルを用いてバッファの仮想アドレスを変換する．その後，OS のメモリデータを読み出し，暗号化してバッファに書き込む．

SMM プログラムが監視対象システムのメモリ全体にアクセスできるようにするために，SSdetector では従来の BIOS の後継である UEFI BIOS を用いる．UEFI BIOS は 64 ビットモードで動作し，4GB を超えるメモリにもアクセス可能である．ただし，SMM でアクセス可能なメモリ領域は制限されているため，SSdetector ではメモリ全体にアクセスできるようにこの制限を解除する．

4　実験

SSdetector の SMM プログラムをオープンソースの UEFI BIOS である TianoCore に実装した．また，Linux のバージョン情報が格納された OS データを取得する IDS を Intel SGX SDK を用いて実装した．SSdetector を用いて，この IDS が OS データを取得するのにかかる時間を計測した．比較として，暗号化を行わない場合および，SGX を用いない場合，SMM を用いない場合についても計測した．SMM を用いない場合には，OS に組み込んだデバイスドライバからメモリデータを取得した．実験に使用したマシンの CPU は Intel Core i7-9700 であり，メモリは 16GB であった．実機の BIOS を変更すると起動しなくなる恐れがあるため，本実験では SGX 仮想化をサポートした KVM を用いて SSdetector を VM 内で動作させた．この VM には仮想 CPU を 1 個，メモリを 2GB 割り当てた．

50 回ずつ計測した OS データの取得時間を図 3 に示す．取得時間のばらつきが大きい場合があったため，箱ひげ図で最大値，最小値，四分位数を示している．SMM を用い，かつ SGX を用いない場合はばらつきが大きくなっており，最大値と第 3 四分位数が非常に大きくなってことがわかる．図 4 に OS データの取得時間の最小値を示す．実験結果より，SMM プログラムを呼び出すことによって 1.1ms のオーバヘッドが生じていることが分かった．これは SSdetector を用いて OS データを取得する時間の 92.4% を占める，しかし，SMM を用いない場合には安全にメモリデータを取得することはできないため，安全性の確保と引き換えに必要となるオーバヘッドである．一方，SGX を用いることによって生じるオーバヘッドは 0.07ms であり，OS データを取得する時間全体の 5.9% であった．

5　まとめ

本研究では，Intel SGX と SMM を組み合わせることで，IDS をより安全に実行可能にするシステム SSdetector を提案した．SSdetector では，SGX のエンクレイヴ内で IDS を実行し，SMM プログラムを用いてシステムのメモリデータの取得のみを行う．エンクレイヴと SMM プログラム間でメモリデータを暗号化することで，取得したメモリデータからの情報の漏洩を防ぐ．今後の課題は，SMM が取得した OS データの整合性検査も行えるようにして改ざんを検知できるようにすることである．また，様々な OS データを取得する IDS を動かして性能評価を行うことも必要である．

参考文献

[1] J. Wang et al. HyperCheck: A Hardware-assisted Integrity Monitor. RAID 2010.

[2] 中野ら．SGX を用いた VM のメモリとディスクの安全な監視．CSS 2019.