Intel SGXとSMMを用いたIDSの安全な実行機構

古賀　吉道[1]　光来　健一[1]

1. はじめに

近年，情報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場合には管理者に通知する必要がある．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容易ではない．例えば，システムが攻撃を受けた後に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．また，IDSが改ざんされ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を安全に実行する手法が提案されてきたが，安全性や性能などの面で問題があった．

本研究では，Intel CPUのセキュリティ機構である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，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. SSdetector

SSdetectorは図1のように，SGXとSMMを用いてIDSがOSのメモリデータを取得できるようにすることにより，システムの安全な監視を可能にする．SGXはIntel CPUが提供する隔離実行環境であり，アプリケーションのメモリ上にエンクレイヴと呼ばれる保護領域を作成する．SSdetectorはエンクレイヴ内でIDSを実行し，CPUによるメモリの暗号化および整合性検査によりIDSの改ざんや盗聴を防ぐ．ただし，SGXだけでは攻撃者によるIDSの停止を防ぐことはできないため，リモートホストからIDSに定期的にハートビートを送信することによりIDSの動作を確認する．

エンクレイヴ内のIDSがシステムを監視するためにOSデータを必要とした際には，SGXの機構を用いてエンクレイヴの外部で動作するSSdetectorランタイムを呼び出す．そして，割り込みであるSMIを発生させてSMMで動作するプログラムを呼び出す．SMMはBIOSによってのみ使用可能な独立した実行環境を提供するため，攻撃を受けることなく安全にシステムのメモリデータを取得できる．SMMプログラムはメモリ上のOSのページテーブルを探索することで，IDSから渡されたOSデータの仮想アドレスを物理アドレスに変換してメモリにアクセスする．SMMでのプログラム実行は低速であるが，メモリデータの取得のみを行うことにより性能低下を抑えることができる．SSdetectorランタイムだけは攻撃を受ける可能性があるため，SSdetectorはエンクレイヴとSMMプログラム間でやりとりするすべてのデータを暗号化する．

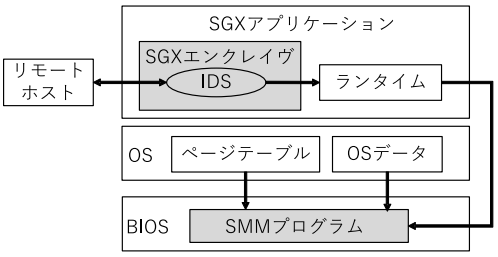

図 1　SSdetectorのシステム構成

SSdetectorではLLViewフレームワーク[1]を用いることで，Linuxのソースコードを利用してOSデータを監視するIDSを開発することができる．LLViewはIDSをコンパイルして生成された中間表現に対してプログラム変換を行い，透過的にSMMプログラム経由で監視対象システムのメモリにアクセスできるようにする．

3. 実験

SSdetectorの有効性を調べるために，OSのバージョン情報を取得するIDSの実行時間を測定した．実機のBIOSを変更するのは難しいため，本実験は仮想マシンを用いて行った．比較として，(1) 暗号化を行わない場合，(2) SGX

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

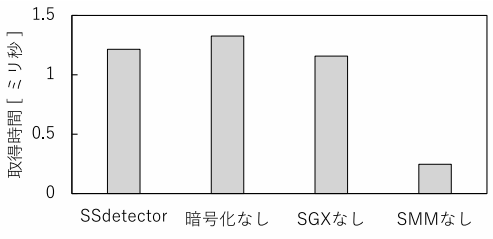

図 2 OS バージョン情報の取得時間の最小値

を用いない場合，(3) SMM を用いない場合についても計測した．

図 2 に OS バージョン情報の取得時間の最小値を示す．この結果より，SMM プログラムを呼び出すことによって約 1ms のオーバヘッドが生じていることが分かった．これは取得時間の 80 %を占める．しかし，SMM を用いない場合には IDS は安全にシステムのメモリデータを取得することができないため，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．

4. まとめ

本研究では 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．今後の課題は，SMM プログラムが取得したメモリデータの整合性検査を行えるようにして取得中の改ざんを検知できるようにすることである．また，暗号鍵を IDS と SMM プログラムの間で安全に共有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す．また，本研究成果の一部は，国立研究開発法人情報通信研究機構の委託研究により得られたものです．

参考文献

[1] Y. Ozaki, S. Kanamoto, H. Yamamoto, and K. Kourai. Detecting System Failures with GPUs and LLVM. In Proceedings of the 10th ACM SIGOPS Asia-Pacific Workshop on Systems, pp. 47–53, 2019.