令和 元年度 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	16237024	学生氏名	河村 拓実
論文題目	Intel SGX 向けコンテナを用いた既存 IDS の安全なオフロード		

1　はじめに

近年，仮想マシン（VM）を提供する IaaS 型クラウドが普及している．クラウドはインターネット経由で攻撃を受けやすいため，侵入検知システム（IDS）で VM を監視する必要がある．IDS を VM 内で実行すると VM に侵入した攻撃者に無効化される恐れがあるため，IDS を VM の外で動かす IDS オフロードと呼ばれる手法が用いられている．これにより，VM に侵入した攻撃者は IDS を攻撃できなくなるが，クラウドの内部犯やクラウド外部の攻撃者から攻撃を受ける恐れがあり，オフロードした IDS の安全性はまだ十分に確保できていない．そこで，CPU のセキュリティ機構である Intel SGX を用いた監視システム SGmonitor[1] が提案されている．SGmonitor は，オフロードした IDS を SGX の保護領域（エンクレイヴ）の中で実行することで IDS を攻撃から守る．しかし，IDS の開発には OS レベルのプログラミングが必要となり，既存の IDS の多くを動かすことができない．

本研究では，Intel SGX 向けコンテナの SCONE[2] を用いることで，既存の IDS をエンクレイヴ内にオフロードして実行可能にするシステム SCwatcher を提案する．

2　クラウドにおける安全な IDS オフロード

IDS オフロードは，図 1 に示すように監視対象 VM の外で IDS を動かす手法である．VM 内に侵入されたとしても IDS は VM の外にあるため，侵入者に無効化される恐れはない．監視対象 VM 内で IDS を動かす従来の手法と異なり，オフロードした IDS は監視対象 VM のメモリを解析し，OS が管理しているデータを取得する．例えば，監視対象 VM のネットワーク情報を取得することで不正な通信を検知することができる. しかし，IDS オフロードを行ってもまだ IDS が攻撃を受ける可能性がある．オフロードした IDS を運用するクラウド内に内部犯がいる可能性があるためである．また，クラウド外部の攻撃者から IDS を攻撃される恐れもある．IDS が攻撃を受けると取得した VM の機密情報を盗まれる危険性がある．

この問題を解決するために，オフロードした IDS を Intel SGX を用いて保護し，VM を安全に監視するシステム SGmonitor[1] が提案されている．Intel SGX は Intel 製 CPU が提供する CPU 支援型のセキュリティ機構である．SGmonitor は IDS を SGX アプリケーションとして作成し，SGX を用いて作成されるエンクレイヴと呼ばれる保護領域の中で IDS を実行する．これにより IDS の改竄や VM から取得した情報の漏洩を防ぐことができる．

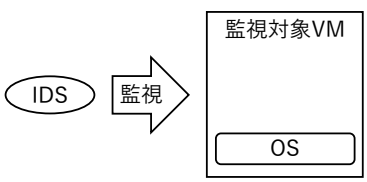

図 1　IDS オフロード

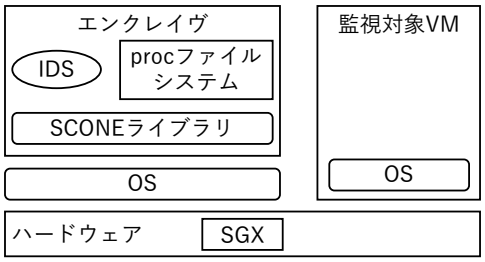

図 2　SCwatcher のシステム構成

しかし，SGmonitor で動作する IDS の開発には OS レベルのプログラミングが必要となるため，一般の開発者には難しい．また，エンクレイヴ内ではシステムの情報を取得するために OS が提供するインタフェースを利用することができないため，既存の IDS の多くを動かすことができない．そこで，ライブラリ OS を用いて既存の IDS をエンクレイヴ内で動かすことを目指したシステム GLvisor[3] が提案されている．GLvisor ではライブラリ OS が IDS に OS のインタフェースを提供するが，既存の IDS を動かすことはできていない．また，複雑なライブラリ OS を動かすことにより攻撃を受ける可能性も高くなる．

3　SCwatcher

本研究では，Intel SGX 向けコンテナである SCONE[2] を用いてエンクレイヴ内で既存の IDS を安全に実行するシステム SCwatcher を提案する．SCONE は Intel SGX を用いて既存のアプリケーションを安全に実行するためのコンテナ環境である．SCONE のライブラリがエンクレイヴ内のアプリケーションに標準 C ライブラリのインタフェースを提供し，必要に応じてエンクレイヴの外の OS を呼び出す．図 2 に SCwatcher のシステム構成を示す．SCwatcher は SCONE を用いて既存の IDS と proc ファイルシステムをエンクレイヴ内で動かす．proc ファイルシステムは従来，OS の機能であり，システムに関する情報を IDS に提供する．SCwatcher では，監視対象 VM 内のシステムの情報を IDS に透過的に提供することにより，既存の IDS をエンクレイヴ内にオフロードして動作させることを可能にする．

3.1　エンクレイヴ内の proc ファイルシステム

SCwatcher は監視対象 VM のメモリ上の OS データを取得して，proc ファイルシステムが提供する疑似ファイルを作成する．疑似ファイルはディスク上に格納される通常ファイ

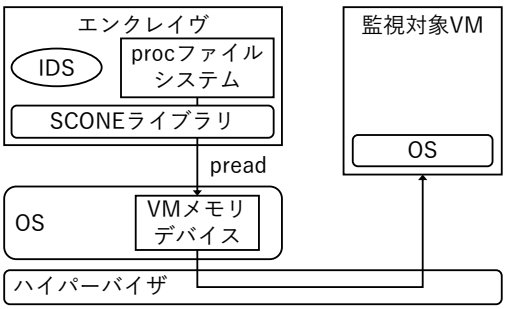

図 3　VM 内の OS データの取得

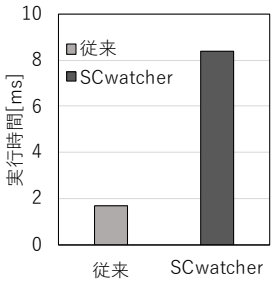

図 4　netstat の実行時間

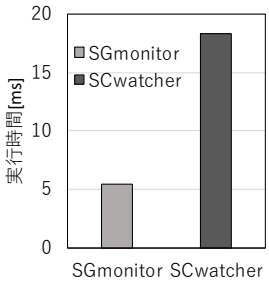

図 5　プロセス一覧の取得時間

ルとは異なり，アクセスした時に動的に内容が作成される特殊なファイルである．例えば，疑似ファイルの/proc/net/tcp や/proc/net/udp には TCP 通信や UDP 通信についての通信先の IP アドレスやポート番号，ステートなどの情報が格納される．proc ファイルシステムは OS のソースコードを利用して開発し，LLView フレームワーク [1] を用いてコンパイルする．LLView は proc ファイルシステムが OS のメモリにアクセスしようとした時に VM のメモリから OS データを取得するようにプログラム変換を行う．

エンクレイヴ内の proc ファイルシステムは SGX の機能を使ってエンクレイヴ外のハイパーバイザを呼び出すことで，監視対象 VM のメモリデータを取得することが可能である．しかし，ソースコードが公開されていない SCONE にこのような機能を追加することは難しい．そこで，SCwatcher では図 3 のようにエンクレイヴ外の OS 内に VM メモリデバイスを用意し，SCONE の機能を用いてこのデバイス経由でハイパーバイザを呼び出す．SCONE はエンクレイヴ内で発行されたシステムコールをエンクレイヴ外の OS に実行させることができるため，pread システムコールを用いて VM メモリデバイスにアクセスする．pread の引数として指定するファイルオフセットにはアクセスする OS データの仮想アドレスを対応させる．

VM メモリデバイスはハイパーバイザを呼び出して，VM のメモリから指定された OS データを取得し，pread システムコールで指定されたバッファを介して proc ファイルシステムへ返す．VM 内の情報の漏洩を防ぐために，取得した OS データはハイパーバイザ内で暗号化し，エンクレイヴ内の proc ファイルシステムで復号する．復号した OS データはハッシュ表を用いて proc ファイルシステム内にキャッシュする．これにより，同じ OS データを必要とした時に再度ハイパーバイザを呼び出す必要がなくなる．

3.2　IDS から proc ファイルシステムへのアクセス

SCwatcher の proc ファイルシステムは IDS のコンパイル時に IDS 本体とリンクされる．しかし，そのままでは IDS が proc ファイルシステムにアクセスしようとすると，SCONE の機能を用いてエンクレイヴ外の OS を呼び出してしまう．そこで，SCwatcher では LLView を拡張して IDS のコンパイル時に，fopen などの標準ファイル関数を専用の関数に置き換え，必要に応じてエンクレイヴ内の proc ファイルシステムにアクセスさせる．これらの関数は proc ファイルシステムの疑似ファイルに対して標準ファイル関数の処理をエミュレートする．IDS が疑似ファイルをオープンする時に，proc ファイルシステムは VM メモリデバイスにアクセスして監視対象 VM の OS データを取得し，疑似ファイルを作成する．疑似ファイルを読み込む際には，オープン時に作成した疑似ファイルの内容を返す．

4　実験

SCwatcher を用いて既存の netstat コマンドを実行し，実行時間を測定した．netstat は proc ファイルシステムから取得した情報を基にネットワークの接続状態を表示するコマンドであり，IDS から呼び出して使われることが多い．比較のために，エンクレイヴを用いずに従来のオフロード実行を行った場合についても測定した．実験に使用したマシンの CPU は Intel Core i7-8700，メモリは 16GB であった．仮想化システムには Intel SGX の仮想化をサポートした Xen-SGX 4.7 を使用した．

実験の結果，SCwatcher でも netstat コマンドを用いて正常に VM 内のネットワーク状態を取得できることが分かった．SCwatcher における netstat の実行時間は，図 4 に示すように従来手法の 5.0 倍になった．この原因を調べるために，エンクレイヴ内で監視対象 VM のプロセス一覧を取得するのにかかる時間を先行研究の SGmonitor と比較した．図 5 に示すように，プロセス一覧の取得時間は SCwatcher の方が 3.4 倍長かった．このことから，SCONE を用いてシステムコールを実行することによるオーバヘッドが大きいと考えられる．

5　まとめ

本研究では，Intel SGX 向けコンテナの SCONE を用いることで，既存の IDS をエンクレイヴ内にオフロードして安全に実行可能にするシステム SCwatcher を提案した．エンクレイヴ内の proc ファイルシステムが監視対象 VM のメモリから OS データを取得することにより，従来のインタフェースを用いて IDS に VM 内のシステムの情報を提供する．今後の課題は，SCONE のオーバヘッドを削減すること，および様々な IDS を実行できるようにすることである．

参考文献

[1] 中野智晴，光来健一．SGX を用いた VM のメモリとディスクの安全な監視．CSS 2019.

[2] S. Arnautov et al. SCONE: Secure Linux Containers with Intel SGX. OSDI 2016.

[3] 篠原悠介．Intel SGX とライブラリ OS を用いた IDS オフロード．九州工業大学卒業論文，2019.