平成 27 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	12237079	学生氏名	山本 裕明
論文題目	GPU を用いた安全な OS 監視システム		

1　はじめに

近年，ネットワークを経由した計算機への攻撃が増加している．従来，計算機内部にセキュリティソフトウェアをインストールして攻撃を検知するという対策がとられてきた．しかし，カーネルルートキットと呼ばれるプログラムをインストールして，OS 自体を書き換えてしまう攻撃が増えてきた．その結果，OS 上で動作する従来のセキュリティソフトウェアでは攻撃を検知できなくなる．そのため，OS の状態を常に監視するシステムが必要となっているが，OS 内部で監視システムを動作させると OS への攻撃時に無力化されてしまう可能性がある．これまで，仮想マシン（VM）を用いた OS 監視システムや CPU のセキュリティ機能を用いた OS 監視システム，専用ハードウェアを用いた OS 監視システムなどが提案されてきた．しかし，高セキュリティ，低コスト，高性能のすべてを満たす OS 監視システムを実現するのは難しかった．

本研究では，OS が動作する CPU やメインメモリから物理的に隔離された汎用の GPU 上で OS 監視システムを並列実行させることを可能にする GPUsec を提案する．

2　OS 監視システム

OS 監視システムは定期的に OS のメモリを検査して，OS の整合性のチェックを行う．OS のコード領域は起動した後に変更されることはないため，コード領域のハッシュ値を計算して事前に計算した値と照合することで，改ざんを検知することができる．また，システムコールテーブルや割り込みテーブルなどは OS の起動時に設定された後に変更されることはないため，コード領域と同様に改ざんを検知できる．一方，OS のデータ領域は実行中に変更されるため改ざんの検知は難しいが，データの整合性を検査したりすることで改ざんを検知することができる．

OS 内部で監視システムを動作させるのが最も容易な監視方法であるが，OS が改ざんされるとすぐに監視システムを停止されてしまう恐れがある．そのため，VM を用いて安全に監視システムを動作させる手法が提案されてきた．この手法は監視対象システムを VM 内で動作させ，VM の外から OS のメモリにアクセスすることで改ざんの検知を行う．しかし，ソフトウェアで実現される VM には様々な脆弱性が報告されており，VM 内部から外側の OS 監視システムを攻撃される可能性がある．また，監視対象システムを VM 内で動作させるため，システムの性能が低下するという問題もある．

専用ハードウェアを用いて OS 監視システムを動作させる手法も提案されている．この手法では PCI バス経由で OS のメモリにアクセスし，リモートホストに送信して改ざんの検知を行う．専用ハードウェアを用意しなければならないため，そのコストが問題となる．そこで，汎用的な CPU のセキュリティ機能を用いて安全に OS 監視システムを動作させる手法も提案されている．この手法は CPU の特殊な実行モードで監視システム動作させるため，実行性能が低いという問題がある．また，監視を行う間は OS を停止させなければならないため，システム全体への影響が大きい．

3　GPUsec

本研究では，OS 監視システムを GPU 上で動作させることを可能にする GPUsec を提案する．GPU は OS が動作する CPU やメインメモリから物理的に隔離されており，ソフトウェアの脆弱性を利用して OS 側から GPU 上の監視システムを攻撃することはできない．また，GPU は多くの計算機に標準的に搭載されている汎用品であり，一部の非常に高性能な GPU を除いてそのコストは低い．その上，GPU は多数の演算コアを有しており，OS 監視システムの並列化を行うことで高い性能を実現することができる．

GPUsec のシステム構成は図 1 のようになる．GPU 上の OS 監視システムはシステムの起動時に実行を開始され，常に GPU を占有して動作し続ける．この時点ではまだシステムは攻撃を受けていないことを仮定する．定期的に OS の監視を行うために，外部の監視ホストから監視対象ホストの GPU 上の OS 監視システムに監視コマンドを送信する．監視コマンドを受信した OS 監視システムは，メインメモリからデータを取得してハッシュ値の計算やデータ構造の解析などを行う．その結果を監視ホストに送信すると，監視ホストでは OS の改ざん検知を行い，必要に応じて管理者への通知を行う．

3.1　GPU からのメインメモリ全体へのアクセス

GPU 上の OS 監視システムはマップトメモリと呼ばれる機能を用いてメインメモリ全体へのアクセスを行う．マップトメモリは GPU のメモリアドレス空間上にメインメモリをマップすることで，GPU からメインメモリを直接アクセスできるようにする機能である．メインメモリ上のデータを GPU に転送するには DMA を使う方が高速であるが，OS が攻撃を受けると正しく DMA 転送が行われることを保証できなくなる．

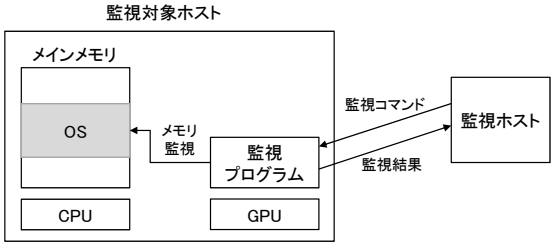

図 1　GPUsec のシステム構成

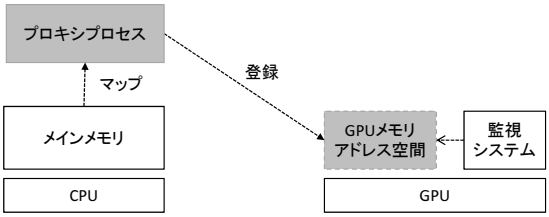

図 2　マップトメモリを用いたメインメモリへのアクセス

マップトメモリを用いることにより，OS が攻撃を受けた後でも GPU は指定したメインメモリ上のデータを安全に取得することができる．

マップトメモリを利用できるようにするために，まず図 2 のように，GPUsec が提供する専用のデバイスファイル経由でメインメモリ全体を OS 上のプロキシプロセスにマップする．このデバイスファイルはメモリを使用中にすることなくプロセスにマップすることを可能にする．メモリをプロセスにマップすると通常は使用中になってしまうため，空きメモリであっても使用中になる．その結果，システムの空きメモリがなくなり，動作に支障をきたすことになるため，GPUsec ではシステムの動作に影響を及ぼさないようにメモリのマップを行う．

次に，プロキシプロセスにマップしたメインメモリを GPU に登録する．登録した後でメモリがディスクにスワップアウトされて存在しなくなるのを防ぐために，GPU のデバイスドライバはメモリのロックを行う．この際に，GPUsec が提供するデバイスファイル経由でマップされたメモリについては，ロックを行っても使用中にならないようにする．これは，空きメモリをロックして使用中になってしまうと，上述のようにシステムの動作に影響を及ぼすためである．GPU へのメインメモリの登録は OS 監視システムの起動後に変更することはできないため，OS が攻撃を受けてもアクセス先をを変えられることはない．

3.2　GPU との暗号通信

GPUsec では，外部の監視ホストは OS 上で動作するプロキシプロセス経由で GPU 上の OS 監視システムと通信を行う．これは，GPU 上のプログラムは直接，ネットワーク通信を行うことができないためである．そのため，プロキシプロセスが攻撃を受けて改ざんされると，OS 監視システムからの監視結果を改ざんして監視ホストに送信される可能性がある．もしくは，監視コマンドを OS 監視システムに転送せず，架空の監視結果を監視ホストに送信される可能性もある．このような場合，監視ホストは OS の改ざんを検知できなくなる．

このような監視結果の改ざんを防ぐために，GPUsec は監視ホストと GPU 上の OS 監視システムとの間の通信を暗号化する．監視ホストは監視コマンドを暗号化してプロキシプロセスに送り，プロキシプロセスはそれを GPU に DMA 転送する．GPU 上の OS 監視システムは監視コマンドを復号して，指示された監視を行う．その後，OS 監視システムは監視結果を暗号化してプロキシプロセスに DMA 転送し，プロキシプロセスはそれを監視ホストへ送る．監視ホストは監視結果を復号し，正しく OS 監視システムによって実行されたかの判断および，監視結果の検証を行う．

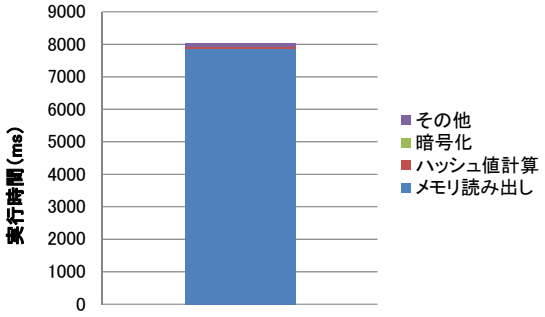

図 3　OS 監視システムの実行時間

暗号通信に用いる暗号鍵は OS 起動時にプロキシプロセスが新たに作成し，GPU 上の OS 監視システムおよび外部の監視ホストに登録する．これにより，OS 監視システムの起動ごとに暗号を変える．また，暗号通信ごとに暗号化，復号化に用いる初期ベクトル (IV) を 1 ずつ増加させることで，暗号化メッセージを保存しておいて後で再利用するリプレイ攻撃を防ぐ．

4　実験

実験には Intel Xeon W3550 の CPU，6GB のメモリ，NVIDIA GeForce GTX 960 の GPU を搭載した PC を用いた．OS として GPUsec 用に修正した Linux 3.16.7 を動作させ，CUDA Toolkit 7.5 を使用した．また，監視ホストには Intel Core i5 4200U の CPU，6GB のメモリを搭載した PC を用い，監視対象ホストとギガビットイーサネットで接続した．

まず，外部の監視ホストから監視対象ホストの OS の改ざんを検知できるかどうかを確認する実験を行った．この実験では，メインメモリ上の OS のコード領域のハッシュ値を計算する OS 監視システムを用いた．OS の改ざんを模擬するために，コンフィグを変更してコンパイルした OS をインストールして再起動し，インストール前後のハッシュ値を比較した．その結果，再起動後にハッシュ値が変化したため，GPUsec を用いて OS の整合性を検査できることが確認できた．

次に，この OS 監視システムの実行にかかる時間を測定した．図 3 に実験結果を示す．1 回の OS の整合性検査にかかる時間は 8 秒程度であることが分かる．その内訳を調べたところ，GPU からメインメモリにアクセスするのに 98% の時間がかかっていることが分かった．今回用いた OS 監視システムは並列化されていないため，並列にメインメモリにアクセスすることによってこの時間は短縮できると考えられる．

5　まとめ

本研究では，CPU やメインメモリから隔離された GPU を用いて OS 監視システムを動作させることを可能にする GPUsec を提案した．GPUsec では GPU 上の OS 監視システムがメインメモリ上の OS を監視し，監視結果を外部の監視ホストに安全に送信することができる．今後の課題は，OS 監視システムを並列化することにより高速化することや，暗号鍵を安全に扱うための仕組みを構築することである．