Vol. 41 No. 6 情報処理学会論文誌 June 2000

Cell/B.E. のSPE Isolationモードを用いた監視システム

光来 健一永田 卓也[†,†]

1. はじめに

近年、ネットワークを経由して個人の計算機が攻撃されるという事件が多発している。各ユーザーはAntiVirusなどの監視ソフトウェアを利用してそれらの攻撃に備えるのが一般的である。しかし、監視ソフトウェアはOSの機能を利用して監視を行っており、攻撃によってOSが改竄されてしまった場合、監視ソフトウェアの実行結果を信用することは出来なくなる。監視ソフトウェアの信頼性を向上させるためにはOSが正しく動作していることを保証すべきであるが、OSはシステム全体を管理しているため、OS自体が改竄されていないことを保証するのは難しい。

本研究では、Cell/B.E. の持つSPE Isolationモードに着目し、SPEからOSカーネルを安全に監視するシステムを提案する。SPE Isolationモードを用いることにより、SPE上で正しい監視プログラムが動作することを保証することができる。さらに、セキュリティプロキシから監視プログラムに定期的にハートビートを送ることで、監視プログラムが動作しているかどうかをチェックする。PS3 Linux上に本システムを実装し、従来手法で監視プログラムを動かした場合と性能を比較した。

2. Cell/B.E.

Cell/B.E. はIBM、ソニー、東芝が共同開発したヘテロジニアス型マルチコアプロセッサであり、PLAYSTATION3やCELL REGZA等に使用されている。このプロセッサは制御系プロセッサコアであるPPEと、演算系プロセッサコアであるSPEによって構成され、それぞれのコアとハードウェアはEIBと呼ばれるバスで接続されている。図1にCell/B.E. の物理構成を示す。

Cell/B.E. において、システム全体の管理を行うOSはPPE上で動作し、OSカーネルのプログラムやデータはメインメモリ上に置かれる。一方、SPEには内部にLocal Store（LS）と呼ばれるメモリ領域が存在し、DMA転送を用いてメインメモリからプログラムをロードしたり、演算に必要なデータを取得したりする。

監視プログラムはOSが動作しているPPE上で動作させるのが一般的であるが、攻撃を受けてOSが改竄されてしまうと、監視プログラムも正常に動作しなくなる。監視プログラムをSPE上で動作させれば、OSが動作しているPPEからハードウェア的に分離されているため安全性は向上する。しかし、PPEはSPEを制御する機能を持つため、SPEで動いている監視プログラムを改竄したり、実行を停止させたりすることができてしまう。

3. カーネルメモリ監視システム

本研究ではSPE Isolationモードを用いることで、監視プログラムをSPE上で安全に動作させることができるシステムを提案する。さらに、Cell/B.E. 搭載マシンの外部に置いたセキュリティプロキシからSPE上の監視プログラムに定期的にハートビートを送ることによって、監視プログラムが動作していることをチェックする。監視プログラムの例として、メインメモリ上のOSカーネルの整合性をチェックするプログラムを対象とする。図2にシステム構成図を示す。

3.1 SPE Isolationモードによる安全な実行

SPE Isolationモードとは、SPEの持つLS領域に対し、外部からのアクセスを禁止する機能である。LS領域に対して外部からアクセス不可になるため、LS内部にロードされたプログラムを実行中に改竄することはできない。その上、セキュリティプロキシとの間でハートビートを行うためにLS上に格納する必要がある暗号鍵を盗まれてしまうこともない。

また、SPE Isolationモードで動かすプログラムはコンパイル時にハードウェアが持つ鍵によって暗号化されており、LSにロードする際にSecure Loaderによって復号化される。Secure Loaderはプログラムの復号化と同時に整合性のチェックを行うため、攻撃者が改竄したプログラムは実行できない。

† 九州工業大学

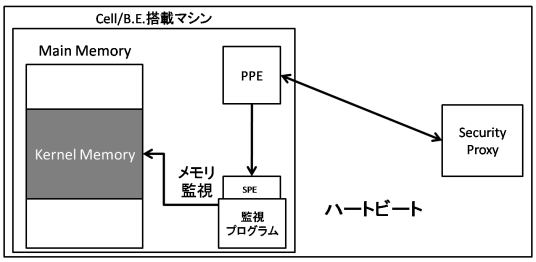

図 1 監視システム全体図

3.2 ハートビートによる実行のチェック

セキュリティプロキシは監視プログラムに定期的にハートビートを送り、監視プログラムが動作しているかどうかのチェックを行う。これは、SPE を Isolation モードで動作させたとしても PPE から SPE の実行停止が可能だからである。ハートビートに対して監視プログラムから正しい返答がなかった場合、セキュリティプロキシは監視プログラムが不正に停止させられたと判断し、Cell/B.E. 搭載マシンに出入りするパケットを全て遮断する。

監視プログラムだけがハートビートに正しく返答できるようにするために、セキュリティプロキシは監視プログラムとの間の共通鍵を用いて暗号化したハートビートを送る。

3.3 カーネルメモリの監視

SPE 上の監視プログラムが DMA を用いてメインメモリにアクセスできるようにするために、SPE が持つ Segment Lookaside Buffer (SLB) にカーネルメモリのアドレスを登録する。SLB は仮想アドレスに実効アドレスを対応づけるためのテーブルである。さらに、アクセスするのに特権を必要とするカーネルのメモリ空間に SPE がアクセスできるようにするために、SPE の持つ Memory Flow Controller (MFC) の状態を変更する。

DMA 転送には一定の時間がかかるため、監視プログラムの高速化のために DMA 転送とメモリ内容のチェックを並列に行う。例えば、配列 A に対して DMA 転送を行うと、その DMA 転送が完了するまでは配列 A のデータを信頼することができない。そこで、本システムでは配列を 2 つ用意し、配列 A に対し DMA 転送を行っている間に、DMA 転送が完了している配列 B の内容をチェックする。

4. 実　　験

実験には PlayStation 3 の 80GB モデル、OS に Fedora 9 の Linux 2.6.27.25-78.2.56 を用いた。

4.1 監視プログラムの実行時間

OS カーネルが使用しているメモリ領域をメインメモリから SPE の LS に DMA 転送し、内容をチェックする監視プログラムの実行時間を測定した。”SPE システム”が全体の実行時間であり、”SPE メモリアクセス”はその内の DMA 転送を行うのにかかった時間である。比較のために、OS カーネルのサイズと同じ 12MB のメモリを確保して内容をチェックする PPE プログラムを作成し、実行時間を測定した。”PPE システム”は全体の実行時間、”PPE メモリアクセス”はメモリのアクセスにかかった時間のみである。

表 1．システム実行時間計測結果

	実行時間 (msec)
PPE システム	290
PPE メモリアクセス	72
SPE システム	8
SPE メモリアクセス	1

SPE を用いてカーネルメモリ全体を監視するのにかかる時間は 8msec 程度であり、PPE で同様のシステムを動かすよりも高速に処理でき、十分実用的である事が分かった。

4.2 カーネルメモリの整合性のチェック

OS カーネルのアドレス空間を 0x100000 毎に区切ってメモリの内容の和を求め、マシン起動時にあらかじめ計算しておいた値と比較した。表 2 にアドレス毎の比較結果を示す。時間によって不変であった領域が○、そうでなかった領域が×である。

カーネル内部で不変であった領域はコード領域および読み込み専用領域であり、変化した領域はデータ領域であった。データ領域は刻一刻と変化するので、整合性をチェックする対象にする必要はない。

4.3 ま と め

本研究では Cell/B.E. の SPE Isolation モードを用いて OS カーネルを安全に監視することができるシステムを提案した。SPE Isolation モードによって正しい監視プログラムが実行されることを保証することができる。さらに、監視プログラムが動作しているかどうかチェックするためにセキュリティプロキシを用いた。今後の課題として、用いる暗号の強化、セキュリティプロキシでのチェックの強化や、ハートビートを SPE 単体で処理できるようにすることなどが挙げられる