社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2005－ＳＬＤＭ－119 (27)
2005／3／18

番犬型 IDS: システム・コールを用いた 侵入検知モジュールの設計と実装

宮本 元　中川 徹　北川 一

豊田工業大学 大学院工学研究科 〒468-8511 名古屋市天白区久方 2-12-1
E-mail: {sd03226, nakagawa, kitagawa} @toyota-ti.ac.jp

あらまし 現在，ソフトウェアで実装された OS 協調型のネットワーク侵入検出システム（IDS）が開発されているが，OS 感染後の被害拡大が大問題となっている．そこで本提案では，侵入検知と警報を IP 接続機器と独立の専用ハードウェア PIDM（Programmable Intrusion Detecting Module）で行い，IP 接続機器単体と相互間の信頼性を確保した．また，処理の中心となるシステム・コールの異常検出処理を FPGA で行うことにより，低コストにて低消費電力を実現し，IP 接続機器上の負荷を最低限のものにした．

キーワード 番犬型 IDS，PIDM，侵入検知，FPGA，ハードウェア／ソフトウェア協調設計

Watch Dog IDS: Design and Implementation of a Intrusion Detection Module using Anomaly Intervals of System Calls

Hajime MIYAMOTO　Toru NAKAGAWA　and　Hajime KITAGAWA

Graduate School of Engineering, Toyota Technological Institute

2-12-1 Hisakata, Tempaku-ku Nagoya 468-8511, Japan
E-mail: {sd03226, nakagawa, kitagawa} @toyota-ti.ac.jp

Abstract Presently, all of the network intrusion detection system (IDS) is based on software implementation and co-operated with some operating system. However, an expansion of the infection is a big problem after the operating system has been infected by worms. In this proposal, we design an intrusion detection module named PIDM (Programmable Intrusion Detecting Module). This module is implemented on an FPGA. PIDM can detect anomaly intervals of system calls and warn for users, so that the dependability of itself and each other IP connected equipments is guranteed. Furthermore, we achieved low electric power consumption of PIDM and low CPU load on IP connected equipments.

Keyword Watch Dog IDS, PIDM, Intrusion Detection, FPGA, Hardware/Software Co-design

1. はじめに

近年，注目されている情報セキュリティ技術の一つにネットワーク上での侵入行為の警報や，証拠の保存を行う侵入検知システム（以下，IDS）がある．侵入検知の方式は，現在，監視対象データと既知の攻撃手法を収めたデータベースとのマッチングを行う不正検出型が主流である．一方，監視対象の異常な振舞いを探す異常検出型は，増加の懸念される未知の攻撃手法に対応可能なため注目されている．しかし，これまで正常／異常の定義の困難さから，効果的な実現方式は見つかっていなかった．

ようやく最近になって，正常／異常の定義が容易な事から，プログラムにおけるシステム・コールの実行パターンを監視対象とする手法が，注目を集めている[1][2][3]．検知精度の向上や，データベース・サイズの低減を狙った研究がなされてきたが，異常検出処理を行う OS の負荷は大きいものとなっている[4][5][6]．このため，IP 接続機器における CPU の交換や増設でこの問題に対応しようとすると，消費電力の大幅な増加を招き，ランニング・コストが増大するという問題が生じることになる．

また，従来，IDS のほとんどが IP 接続機器上のソフトウェアで実現されており，検知結果の通知を行うものの，感染後，その機器はデータの改ざんを受ける可能性がある．従って，IP 接続機器自体と通知された検知結果の信頼性が保証できないことが問題となってい

た．また，感染後にIP接続機器が他のIP接続機器への攻撃を行う可能性もある事から，IP接続機器相互間の信頼性が確保できない事も大きな問題となっていた．

そこで，本設計では，これらの問題を解決するため，これまでIP接続機器内のソフトウェアで実現されてきたIDS処理の中から，核である異常検出処理と警告発生処理を切り出し，IP接続機器のCPUとは独立のFPGA上で実現することを提案した．そして，IP接続機器の監視を独立のハードウェアで行うこのIDSを番犬型IDSと名づけた．このFPGAを搭載したハードウェアで侵入検知を行うことにより，IP接続機器単体と相互間の信頼性確保，低消費電力化，IP接続機器上のOS負荷削減を目的とする．

2. 番犬型 IDS

今回試作した番犬型IDSは，システム・コールの実行検出と検査情報の生成および入力をソフトウェア部で行い，異常検出と警告を侵入検知IPコアPIDM（Programmable Intrusion Detecting Module）が実装されたハードウェア部で行っている．試作システムの概要を図1に示す．この処理系の設計には，ハードウェア／ソフトウェア協調設計の手法を用いており，システム全体の処理を予めC言語で記述した後，処理速度などを考慮し，その処理をハードウェア部とソフトウェア部に分割して実現する．ハードウェア部に割り当てられた処理を行う部分のプログラムは，PIDMシミュレータとして，後にハードウェア部の試験や評価に用いる．検査情報送出部とPIDMシミュレータは，sp_watcherというプログラムで実現されている．以下，各処理部の解説を行う．

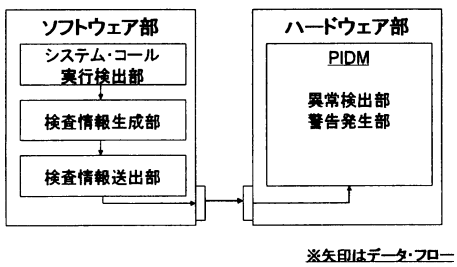

図 1: 試作した番犬型IDSの概要．

2.1. ソフトウェア部

ソフトウェア部では，システム・コール実行情報の抽出，検査情報の生成およびハードウェア部への送出を行う．これらの処理は，すべてIP接続機器上のCPUが実行する．システム・コールの実行情報の抽出は，カーネル内で，ソフトウェア割り込みが行われた際にその情報を取得することで実現される．また，この情報の他に，実行中のプロセス情報を取得し，これらの情報を関連付ける．この検査情報は，カーネル・メモリ上の循環バッファに格納され，その後，この循環バッファにアクセスできる特殊なシステム・コールの実行によりユーザ空間にコピーされる．sp_watcherは，USBドライバを通じてこの情報をハードウェア部へ送出する．

2.2. ハードウェア部

ハードウェア部では，PIDMを実装したFPGAが処理を行い，ソフトウェア部から入力された情報をもとに，異常検出を行う．PIDMは異常検出アルゴリズムによる侵入の検出と，異常検出後は後述の柔軟な警告発生を行う．

PIDMに実装された異常検出アルゴリズムは，コンピュータ・ウイルスの一種であるワームと一般的なサーバ・プロセスとの振舞いの違いから異常を検出していく．ワームは，一般的にセキュリティ・ホールの攻撃，ワーム自身のコピーと実行，感染活動などを行う．このため図2にあるように，クライアントからの要求の応答処理を繰り返すサーバ・プロセスに比べ，システム・コールの実行回数が多い．また，実行間隔がネットワーク通信速度に依存するサーバ・プロセスに比べ，実行間隔が短くなる．そのため，PIDMは，システム・コール実行回数と実行間隔から得られる実行速度に閾値を設け異常プロセスの検出を行っていく．図2のデータを元に，閾値は，実行回数が65534，実行速度が1秒間当たり実行回数8191とした．

このアルゴリズムは，他の異常検出手法に比べ単純であるが，処理速度の向上や，FPGAに搭載可能な規模にIPコアを収めることを目指すことに加え，異常検出の決定的なアルゴリズムが無いなどの理由から，PIDMの第1版ではこの方式を採用した．

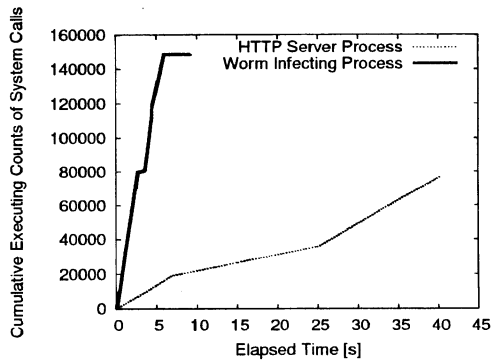

図 2: ワームに感染したプロセスとHTTPサーバ・プロセスのシステム・コール累積実行回数．

PIDM は，入力データを管理するデータ・テーブルを持っている．このデータ・テーブルは異常検出に必要な 3 つの情報（プロセス ID，システム・コール実行回数，リストへの最終アクセスからの時間）を，ユニークなプロセス ID ごとに管理する．データ・テーブルは，LRU（Least Recently Used）アルゴリズムによるデータの置き換えと，アクセスされていないデータの消去により管理される．

データ・テーブルが収めるプロセス ID の数，すなわちデータ・リスト数は，検知精度を低下させないため，十分な大きさであることが望ましい．しかし,データ・リスト数を増やすと，ハードウェア規模が著しく増大するため，適当な大きさに設定する必要がある．そこで，PIDM シミュレータを用い，データ・テーブルが管理するプロセス ID のリストの数を変化させ，ヒット率の測定を行った．ヒット率は，以下の式で算出される．

$$\text{ヒット率} = \frac{\text{データ・リスト上でのアクセス回数}}{\text{データ・リストへの全アクセス数}}$$

評価には，実際の処理で得られた，システム・コールを実行したプロセス ID のトレース・データを入力データとして用いる．これを PIDM シミュレータに入力してヒット率を測定した．リスト数は，1，2，4，8，16，32 の 6 種類に変化させた．

評価結果は，CPU 使用率 100%の状態で入力データを取得した場合，リスト数 1 のとき約 58%，2 のとき約 95%，4 のとき約 98%となり，それ以降は約 99%となった．以上の結果より，PIDM で採用するリスト数は 4 としている．

PIDM は，上記アルゴリズムにより異常な振舞いのプロセス ID を検出した後，警告を発生する．警告は，LED を用いたプロセス ID の表示と，ブザーによる警告音の発生で通知される．また，これに伴って IP 接続機器の電源を制御して再起動を行うなど，柔軟な警告の発生が可能である．これにより，感染後の IP 接続機器上での通知よりも信頼性の向上を図っている．

PIDM を FPGA（Xilinx XC2S100）へ書き込んだ際の使用ゲート数は 22,493，使用スライス数は 620 となっている．

3. 評価実験

3.1. 処理効率の評価

異常検出部にハードウェアで実現した PIDM と，ソフトウェアで実現した PIDM シミュレータを用いたそれぞれの場合の処理効率の比較を行った．比較項目は，CPU 使用率と処理時間である．CPU 使用率の評価は，sp_watcher の CPU 使用率を 0.6 秒間隔で 1000 回測定した．また，処理時間は，PIDM では VHDL シミュレータで確認し，PIDM シミュレータでは gettimeofday システム・コールを使い測定した．

CPU 使用率の評価結果を図 3に示す．PIDM を用いた場合，CPU 使用率が 20%以上になる場合が減少していることが確認できる．また，平均 CPU 使用率と最大 CPU 使用率は，PIDM が 2.3%と 34.3%であり，PIDM シミュレータが 3.6%と 51.9%であった．測定対象である sp_watcher は，PIDM を用いた場合は検査情報送出処理を行い，PIDM シミュレータを用いた場合は異常検出処理を行う．このため，異常検出処理よりも，検査情報送出処理のオーバーヘッドの方が小さいことが分かる．

次に，処理時間は，PIDM が最長で 0.869 μs，最短で 0.788 μs，PIDM シミュレータが平均 0.957 μs となり，PIDM の処理がより高速なことが確認された．

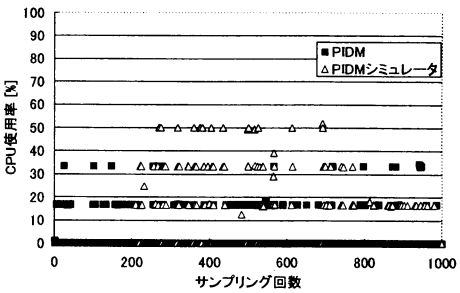

図 3: 異常検出アルゴリズムの実現方式の違いによる，CPU 使用率の違い．

3.2. ワーム感染の検知

ワームが感染活動中の隔離されたネットワーク環境下で，PIDM による検知評価を行った．

評価の結果，PIDM は，ワームのコピーや起動などを行っているプロセスのプロセス ID を検出した．ワームに感染したプロセスの起動から終了まで，PIDM へのデータの平均送出間隔は 29.3 μs であった．USB インターフェイスの通信速度は 5.68 Mbps であることから，最大となる 6 バイトのデータの送出には，1 つあたり 8.45 μs かかる．PIDM の処理時間は最長で 0.869 μs であるため，全体として，このシステムは実用に耐えうると考えられる．

ただし，システム・コールの高速な実行による循環バッファの溢れも確認されたため，検知制度の向上を図る際は，通信速度の向上や検査情報取得のための時間短縮が必要となる．

4. おわりに

本設計では，侵入検知 IP コア “PIDM” を実装し，異常検出処理を FPGA で行う IDS を試作した．これに

より，システム・コールを監視対象とした異常検出処理が外付けのハードウェア・モジュールで実現可能であることを示した．また，異常検出処理と警告発生処理を外付けのハードウェア・モジュールで行うことで，IP 接続機器単体と相互間の信頼性を確保し，IP 接続機器の低消費電力化と OS 負荷の削減を実現した．

今後，検知精度は高いが，長い処理時間がかかる異常検出アルゴリズムを実装した際，負荷のさらなる削減が期待できる．その上，定型化した異常検出処理を ASIC で実現することで，さらなる低消費電力化と処理の高速化が期待できる．

今後の展開としては，①より多くのシステムに対応するため学習機能を備えた異常検出アルゴリズムの開発，②同アルゴリズムを FPGA 上で自動的に再構築するシステムの開発，③感染したホストへのハードウェア信号によるホストの停止や再起動機構，④PIDM を接続した IP 接続機器間の検知情報の共有などを考えている．

情報セキュリティ技術の中でも，侵入行為の検知や警報，証拠保存を行えるより信頼性が高い侵入検知は，今後ますます重要になってくる．とはいえ，現在のようなソフトウェアでの実現だけでは不十分であり，上記機能を実現した LSI をすべての IP 接続機器へ搭載することによって，ネットワーク社会の安全性がさらに確保されると考えている．

文　献

[1] S. Forrest, S. A. Hofmeyr, A. Somayaji and T. A. Longstaff, “A Sense of Self for Unix Processes,” In Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, 1996.

[2] S. Forrest, S. A. Hofmeyr and A. Somayaji, “Computer Immunology,” Communications of the ACM, Vol. 40, No. 10, pp. 88-96, 1997.

[3] S. A. Hofmeyr, S. Forrest and A. Somayaji, “Intrusion Detection Using Sequences of System Calls,” Journal of Computer Security, Vol. 6, No. 3, pp. 151-180, 1998.

[4] David Wagner and Drew Dean, “Intrusion Detection via Static Analysis,” In Proceedings of the 2001 IEEE Symposium on Security and Privacy, pp. 156-168, 2001.

[5] 阿部 洋丈，大山 恵弘，岡 瑞起，加藤 和彦，“静的解析にもとづく侵入検知システムの最適化，”第 15 回コンピュータ・システムシンポジウム，pp. 7-16，2003.

[6] Mizuki Oka, Hirotake Abe, Yoshihiro Oyama, and Kazuhiko Kato, “Intrusion Detection System Based on Static Analysis and Dynamic Detection,” In Proceedings of Forum on Information Technology (FIT 2003), 2003.