L_034

Hi-sapへのSELinuxの適用と評価
An Adoption of SELinux against Hi-sap and its Evaluation

福田 亮平† 原 大輔† 中山 泰一†
Ryohei Fukuda Daisuke Hara Yasuichi Nakayama

1. はじめに

近年、インターネットの普及により個人でウェブサイトを開設する人が増加している。それらのサイトを設置する場所として共有型ホスティングサービスに人気が集まっている。

しかし既存のウェブサーバには、共有型ホスティングサービスのような大規模環境ではサーバ組み込みのモジュールとして提供されるプログラムをセキュアかつ高速に利用できないという問題がある。

このような問題に対して Hi-sap [1] というウェブサーバシステムを提案して来た。Hi-sap はサーバに格納された多数のサイト群をサイトやコンテンツなどのパーティションに分割し、パーティション毎に異なるユーザ権限でサーバプロセス（以後 worker プロセス）を実行する。これにより組み込みモジュールをセキュアかつ高速に利用することが可能となっている。

しかし、Hi-sap ではソフトウェアや組み込みモジュールのバグを利用したプロセスの権限昇格などは防ぐ事が出来ない。

その問題を解決する方法として、セキュア OS の利用が挙げられる。以下では、セキュア OS の一つである SELinux [2] 及び ウェブサーバシステム Hi-sap について詳しく述べる。

1.1 SELinux

本研究ではセキュア OS の一つである SELinux を用いる。セキュア OS とは、強制アクセス制御と最小特権の機能を有する OS と定義されている。

SELinux では 最小特権の機能をドメインと呼ばれる強力なサンドボックスで実現しており、各ユーザ（root ユーザも含む）に対して、利用可能なドメインを割り振ることで強制的なアクセス制御を行う。以上の方法により、SELinux は強固なセキュリティを実現している。

1.2 Hi-sap

Hi-sap は、共有型ホスティングサービスなどの大規模環境で、組み込みモジュールをセキュアかつ高速に利用できることを目的としている。セキュアに利用するために、サーバに格納されたサイト群をパーティションに分割し、専用のユーザ権限で worker プロセスを実行する。

また、高速に利用するために、*Content Access Scheduler* という、ウェブサーバに特化したスケジューラを提案している。これは、worker プロセスの生成タイミングをクライアントからリクエストがあった時とし、高負荷に陥った場合には、動作中の worker プロセスを選択アルゴリズムにしたがって、適宜終了させるというものである。

† 電気通信大学 情報工学科
Department of Computer Science,
The University of Electro-Communications

しかし、Hi-sap では、プロセスを攻撃者に乗っ取られた場合に、権限の昇格を防ぐ事は出来ないため、管理者権限へ昇格され、OS 全体を乗っ取られてしまう危険性がある。

1.3 本研究の目的

本研究ではセキュア OS である SELinux を Hi-sap に適用することにより、バグを利用した権限の昇格を防止し、ウェブサーバをよりセキュアなものとする。

2. 設計

Hi-sap に SELinux を適用するに当たって、以下の要件を満たすようにする。

- 各パーティションのリソースに、それぞれ独自のファイルコンテキストを付与する。
- 各パーティションを実行する worker プロセスを、それぞれ独自のドメインで動作させる。
- サーバプロセスが他のドメインへ遷移することを禁止する。

まず、各パーティションのリソースに対して、独自のファイルコンテキストと呼ばれるラベルを付与する。この時、ファイルコンテキストは他のパーティションのリソースへ付与するファイルコンテキストや、すでに Linux に存在しているリソースに付与されたファイルコンテキストとは重複させない。

次に、各パーティションを実行する worker プロセスが独自のドメイン下で動作するようにし、そのドメインに対して、自身のパーティション専用のファイルコンテキストを付与されたリソースに対するアクセス権限を設定する。

最後に、worker プロセスが、他の worker プロセスのドメインへ遷移しないようにする。

以上のように設計することによって、実行中の worker プロセスが他の権限へ移る事は無いため、結果的にバグを利用した権限昇格を防止することが出来、サーバ内部がよりセキュアな状態に保たれることになる。

3. 実装

Hi-sap において、各 worker は /vhosts/"Worker ID"/ というディレクトリにインストールされている。本実装では、 worker として用いた Apache の起動・停止時を行う /vhosts/" Worker ID "/bin/ apachectl に 全ての worker で同一のセキュリティコンテキストを付与した。一方それ以外の部分に関しては、 worker が起動中に使用するため、 worker 毎に異なるセキュリティコンテキストを付与した。

また、各 worker プロセスがアクセスできる範囲は、一部の例外を除いて自身がインストールされているディレクトリ以下に制限を行うとともに、CGI 実行時のドメインを worker プロセスのドメイン内部に作り、/vhosts/"Worker ID"/cgi-bin 内に 格納されている CGI の不具合による被害を cgi-bin 内部のみに止めるようにした。

一部の例外を以下に挙げる

- dispatcher として用いた Apache のログ用ディレクトリに対するファイルの追加許可
- Apache のモジュールを利用するため、モジュールの存在するディレクトリ及びファイルへの読み込みなどの許可
- 各種ライブラリなどの読み込みの許可

worker プロセスに対して、以上のようなアクセス権限を与えることにより、非常に高いセキュリティを実現することが出来ると考えられる。

4. 評価実験

図 1 は、*httperf* ベンチマークバージョン 0.8 を用いて行った、本システムに対する基礎性能評価実験の測定結果である。X 軸はリクエスト頻度、Y 軸はスループットを表している。Hi-sap は SELinux を有効にした状態で動作させ、比較対象は Apache、suEXEC を有効にした Apache、*One-to-one* 方式‡ を用いた。リクエスト対象は phpinfo() とし、一回のリクエストによるデータ転送量は 40 KB 程度である。図 1 が示すように、Hi-sap と

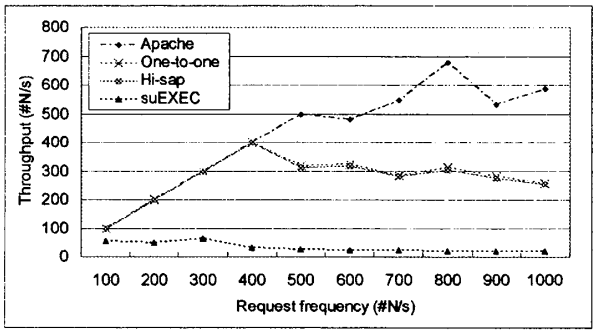

図 1: 基礎性能評価実験

Apache と比較して平均で 28.0%、最大で 56.5%の性能低下が見られた。また、One-to-one と比較して平均 1.0%、最大 2.6%の性能低下が見られた。同じリバースプロキシ型のネットワーク構成を取る Hi-sap と One-to-one の性能差は非常に低いため、Apache との大きな差の原因はリバースプロキシ型を取ったために発生する、通信によるオーバーヘッドであると考えられる。

その一方で、suEXEC を有効にした Apache と比較して平均で 10.2 倍、最大で 14.3 倍の性能を達成しており、本実装の有効性が示されたと言える。

‡ リバースプロキシ型のネットワークを取り、サイト毎に専用のウェブサーバを割り当てる方式

次に実アプリケーション（Weblog）をリクエスト対象として用いて同様の実験を行った。用いた Weblog は Ruby で記述された tDiary バージョン 2.0.2 である。図 2 に測定結果を示す。Hi-sap は Apache と比較して平均

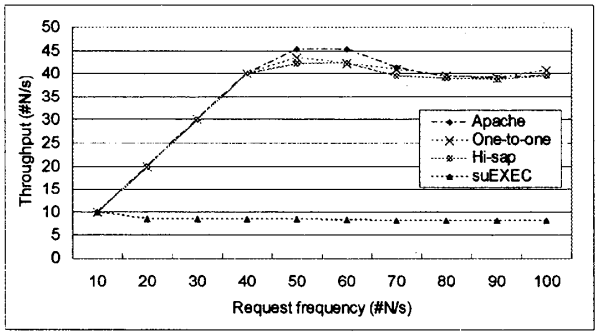

図 2: 基礎性能評価実験：Weblog

で 2.0%、最大で 6.9% の性能低下が見られ、One-to-one とは、平均で 1.1% 最大で 3.7% の性能低下が見られた。phpinfo() を対象にした実験に比べ、Apache との性能差が小さくなったのは、実アプリケーションを用いたために、通信時間に対するリクエストの処理時間が増加したためである。

最後に SELinux が無効時と有効時それぞれの場合に、PHP スクリプトによる /tmp/ に対する書き込み実験を行った。その結果、パーミションが 777 であり、あらゆるユーザに対して読み取り、書き込み、実行が許可された /tmp/ ディレクトリに対して、SELinux が無効時は書き込みが成功し、有効時には失敗した。これは、ドメインによるアクセス制御が既存のユーザによるアクセス制御より、優先されていることを示している。

5. 考察

SELinux を有効にした Hi-sap は実アプリケーションを用いた実験において、Apache や リバースプロキシを用いているという点が共通する One-to-one 方式に比べ性能低下は小さく、高セキュリティを維持しつつも、高い処理性能を達成できた。また、本システムは同種のセキュリティ機構である suEXEC に対して圧倒的に高い性能を示した。

以上から、SELinux を適用した Hi-sap は十分実用に耐え得る性能を示したといえる。

謝辞

本研究は、一部、独立行政法人情報処理推進機構（IPA）「未踏ソフトウェア創造事業」の支援による。

参考文献

[1] 原大輔, 中山泰一:セキュアかつ高性能なウェブサーバの設計と実装, 情報処理学会第 47 回プログラミング・シンポジウム報告集, pp.71–78 (2006).

[2] BILL MCCARTY:SELINUX システム管理－セキュア OS の基礎と運用, オーム社 (2005).