2007 年度
セキュア OS の導入に関する課題の試行結果
報告書

2008 年 6 月

特定非営利活動法人 日本ネットワークセキュリティ協会

技術部会　セキュア OS 普及促進ワーキンググループ

目　次

1. はじめに

1.1 経緯と目的

様々なタイプのセキュア基盤（ＯＳ）を利用したソリューションモデル考案、及び、当該技術の普及促進を図り、産業界のインフラにまで発展したインターネットを少しでも安全に利用できるよう社会貢献することを目的として活動を行う。

1.2 WG 活動の概要

現在、大規模な構築が始まった医療情報ネットワークの構築にＰＫＩ+ＶＰＮが用いられるようになった背景には、ＯＳＩの上位層での暗号化対策に脆弱性が多く含まれるという資料が、厚生労働省をとりまく機関から公表されたことがきっかけとなった。

セキュアＯＳも同様に下位層での最も基礎的な対策であるが、国内では抵抗感が相変わらず根強くあるため、これらを少しでも払拭できるような効果的な普及啓蒙活動ができないかと当WGメンバーで数回に渡り議論をした。

この結果、後述するように、ＪＮＳＡのＤＭＺ設置サーバに対してセキュアＯＳを適用し、その過程も含めて広く公表していくことで多くの方にセキュアＯＳの本質を理解をしていただけるよう活動を展開することになった。

平成 19 年度のメンバーは以下の通りである。

リーダー	澤田　栄浩（株式会社 JTS）
メンバー	楠木　秀明（日本 CA 株式会社）
	栗原　実（株式会社富士通ソーシアルサイエンスラボラトリ）
	河本　高文（東芝ソリューションズ株式会社）
	田口 裕也（株式会社 JTS）
	武田　健太郎（株式会社 NTT データ）
	冨田　高樹（みずほ情報総研株式会社）
	原田　季栄（株式会社 NTT データ）
	半田　哲夫（NTT データ先端技術株式会社）
	三田　聖彦（インフォコム株式会社）
	やすだ　なお（JNSA／株式会社ディアイティ／サイバー大学）
協力	坂本　慶（株式会社ディアイティ）

1.3 本報告書の構成

本報告書は今回の実験の内容と結果を、読者が自らの環境にセキュア OS を導入する際の参考とできるよう、以下の構成のもとで記述している。

第 1 章　はじめに
　セキュア OS 普及促進 WG の設立の経緯と活動内容の概略について説明する。
第 2 章　セキュア OS を利用したソリューションモデルの検討
　今回の実験の対象としたシステムや OS をどのように選択したかについて、その検討の経過を説明する。
第 3 章　TOMOYO Linux の導入実験
　導入計画の立案からセキュア OS としての強制モードの運用開始までの手順について、実際に実験で行った内容の説明を行う。
第 4 章　サーバの運用実験
　サーバの運用時における実施事項やトラブルの発生状況について、実験結果をもとに説明する。
第 5 章　今後のセキュア OS の普及に向けた知見
　今回の実験で明らかになったことをもとに、今後セキュア OS を導入する際に考慮すべき事項や、今回の実験では確認できなかった事項について整理する。
第 6 章　まとめ
　今回の活動内容の総括を行う。

1.4 想定する読者

本報告書は、以下の読者に活用してもらう想定のもとで作成している。

(1) セキュア OS の導入を検討している方

組織のセキュリティ管理者やシステム担当者等、組織で運用しているサーバ等の情報システムにセキュア OS の導入を検討している方が、導入可能性の判断や支援の参考として使うことができるよう、導入時に検討すべき事項や留意点について整理している。

(2) セキュア OS（特に TOMOYO Linux）に関心のある方

セキュア OS ではどのようなことができるのか、そうした機能を使うにはどのような設定や操作を行えばよいかについて、TOMOYO Linux を実際のサーバ運用に用いる場合を例にわかりやすさを重視して説明している。

1.5 本書で使われる用語・語句の定義

セキュア OS においては通常の OS にはない概念を用いることもあり、設定等の場面において聞きなれない用語が用いられることがある。こうした用語について、本報告書では以下のような意味[1]の用語として使用している。

- 「学習モード」: TOMOYO Linux の動作モードのひとつ。要求された処理を行うのに必要なアクセス許可を自動的に収集する。
- 「確認モード」: TOMOYO Linux の動作モードのひとつ。要求された処理を行うのに必要なアクセス許可があるかどうかを確認する。
- 「強制アクセス制御」: あるシステムポリシーを、そのコンピュータシステム内のユーザやプログラムに対して強制できる機能のこと。対してこうした機能をもたないアクセス制御のことは、「任意アクセス制御」と呼ぶ。
- 「強制モード」: TOMOYO Linux の動作モードのひとつ。要求された処理を行うのに必要なアクセス許可がなければ拒否する。
- 「最少特権」: コンピュータシステム内の主体（ユーザやプログラム）の持つ強力な権限を役割や用途に応じて分割し、個々の権限は必要最小限にするという考え方。最小特権とも書く。
- 「セキュア OS」: 強制アクセス制御機能や最少特権機能を中核とした、セキュリティに配慮した OS のこと。

[1] セキュア OS 関連の用語定義については、以下の文献の内容を参考にした。
内閣官房情報セキュリティセンター: 電子政府におけるセキュリティに配慮した OS を活用した情報システム等に関する調査研究, http://www.nisc.go.jp/inquiry/pdf/secure_os_2004.pdf
内閣官房情報セキュリティセンター: 電子政府で利用する情報システムへのセキュリティ機能を強化したＯＳの適用可能性等に関する調査研究, http://www.nisc.go.jp/inquiry/pdf/secure_os_applicabilty.pdf

2. セキュア OS を利用したソリューションモデルの検討

2.1 実施すべき活動内容の検討

国内におけるセキュア OS の状況は、ここ 3, 4 年の間に劇的に変わった。当初は、ごく限られた分野の専門家しか知らなかったものが、雑誌の特集、講演会、イベントなどにも大きく取り上げられるようになり、今やセキュア OS という言葉、概念などは既に広く知られるところになったと言って過言はないであろう。一方、情報漏洩事件などの増加もあり、セキュリティに関するニーズや自覚は着実に高まっている。セキュア OS は社会の IT 化、セキュリティに関するニーズの高まりを受けて、活用されることが期待されたが、現状はそれとは大きくかけ離れている。

本 WG では、セキュア OS が利用できるにもかかわらず活用されていないというこの状況について、実際に運用しているサーバにセキュリティ強化 OS を導入することにより、その原因を調べ、またその結果得られた結果や知見を広く公開することにより、今後のセキュア OS の普及、促進を行うことを目指し活動をすることとした。

2.2 対象システムの選定

対象システムの選定にあたっては、以下の条件を設けた。

- インターネットに接続されている情報システムであって、実証実験を通じてセキュア OS の有効性を客観的に検証可能であること
- 実証実験の中立性・客観性の確保の観点から、使用する OS のベンダ等と密接な利害関係にあるような情報システムでないこと
- 情報システムの運用主体において実証実験の活動の主旨についての理解が得られ、かつその協力を得られること

WG メンバーによる検討の結果、JNSA 自体の事務局の web サーバを本実験の対象システムとすることにした。

2.3 対象セキュア OS の選定

現在利用可能なセキュア OS として、オープンソースソフトウェアでは SELinux, AppArmor, TOMOYO Linux、プロプライエタリソフトウェアでは HiZARD, PitBull, SecuveTOS, SHieldWare などが存在している。

今回の実験では、以下の理由から TOMOYO Linux を対象システムに導入することとし

た。

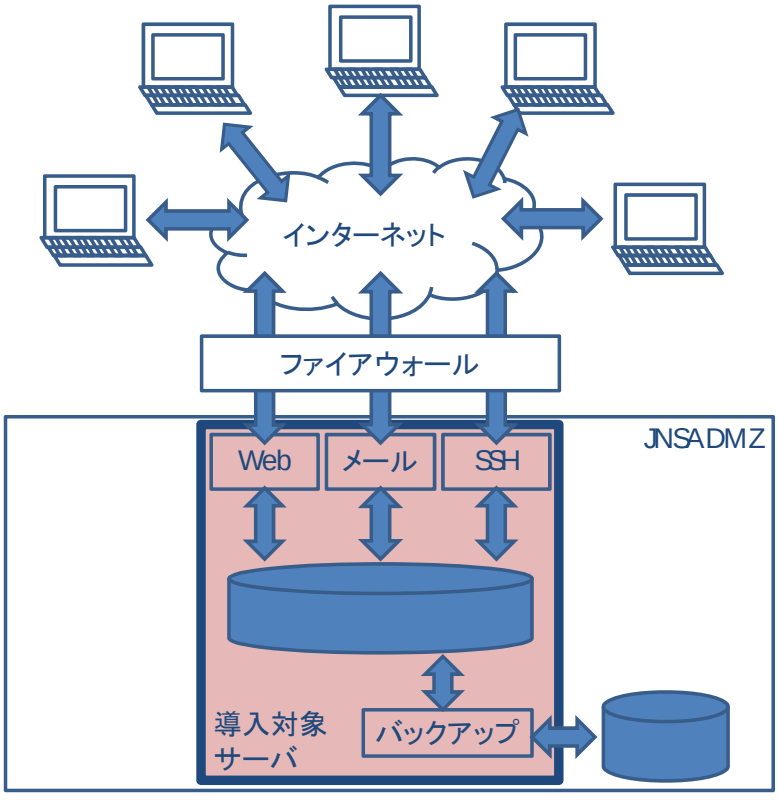
- 対象システムに搭載されている RedHat Enterprise Linux 4（RHEL4）に対応しており、アプリケーションの入れ替え等の手間を省くことができる。
- セキュア OS の導入・設定・運用という流れを、自動学習という機能を用いることでサーバ管理者自らが実施することができるので、導入時の負荷を軽減させることができる上に、導入実験としての今回の目的にも適している。
- 開発が日本で行われている上、開発メンバより本実験への支援が可能との回答を得ており、導入・運用における疑問点や障害の解消が容易となることが期待できる。

2.4 実験環境の概要

対象システムは JNSA 内部で運用されている 1U のサーバで、CPU は広く用いられている Intel i386 アーキテクチャである。ファイアウォールを経由してインターネットに接続されている。

外部に向けて提供している主なサービスは Web とメールである。サーバの管理作業には SSH ログインを使用しており、一般ユーザでのログインのみを認めている。管理者権限が必要な操作は su コマンドで権限を取得の上で実施している。

このほか特筆すべき構成として、バックアッププログラムが定期的に動作している。動的な Web コンテンツを格納しているデータベースの内部データを USB 接続のディスクに保存するように設定されている。



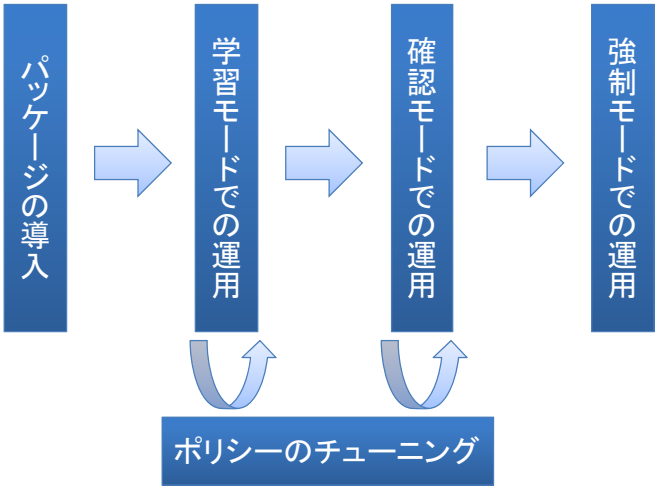
3. TOMOYO Linux の導入実験

3.1 導入へ向けた検討

TOMOYO Linux を用いたセキュリティ強化を実現するにあたり、まずは対象サーバのディストリビューションとバージョン、パッケージ構成と稼動サービス、アップデート等の運用方法、といった情報をまとめた。今回の導入実験では、新規に運用を開始するサーバではなく、すでに稼動しているサーバに TOMOYO Linux を適用し、セキュリティ強化を実現する。TOMOYO Linux 開発チームと導入環境について議論したところ、導入にあたり特に問題となる事項は挙がらなかった。

実現するセキュリティレベルについては、導入前に綿密に設計するのではなく、TOMOYO Linux の導入を進めながら、ポリシーの学習結果を見た上で検討する、という手法を取ることとした。これは、セキュリティ強化を段階的に実現していくという、「積み上げ式のセキュリティ強化」が可能である、という TOMOYO Linux の特徴を活かした導入方法である。

TOMOYO Linux の導入の大まかな流れを以下の図に示す。



3.2 パッケージの導入と学習モードでの運用の開始

TOMOYO Linux カーネルとツールを導入し、システムを再起動して、学習モードでの稼動を開始した。カーネルとツールはいずれも RPM パッケージが TOMOYO Linux プロジェクトから提供されており、導入作業自体は標準的な Linux 管理者であれば簡単に実施できる。

学習モードでは、TOMOYO Linux のアクセス制御は実施されず、見かけ上通常の Linux とまったく同様に動作する。通常の Linux と異なることは、システム中で発生するアクセ

スが全てポリシーとして記録されていく、という点にある。実際の環境で学習モードでサーバを運用したのちポリシーを見れば、どのようなプロセスがサーバ内で動作しているかが明らかになり、それを基準にどのようなセキュリティ強化を実施するかを検討することができる。

2 週間ほど運用したのち、学習されたポリシーを TOMOYO Linux 開発チームと共に精査したところ、冗長な学習を防ぐための初期設定がなされていない状態で学習が進んでいることが判明した。導入操作を振り返ってみると、TOMOYO Linux のツールパッケージの誤ったバージョンを選んだ[2]ことで、初期設定が正しく作成されていないことが判明した。

冗長な学習が実施されてはいたが、学習結果として得られる情報に不足が生じるわけではなく、肥大化したポリシーによるメモリ消費も 6MB 程度に収まっていた。このため、すぐに問題が発生することはないと判断し、引き続き学習モードでの運用を継続した。

3.3 セキュリティ強化対象の検討

学習されたポリシーを参照してサーバでどのようなプロセスが起動されているかを確認し、TOMOYO Linux 開発チームと共同でセキュリティ強化の対象を検討した。今回は外向きサービスを提供している Apache を最初のセキュリティ強化対象としてポリシーを作成することを決定した。また、TOMOYO Linux はファイル操作だけでなく、ネットワークやシグナル、ケイパビリティなど多様な資源を制御できるが、今回は最も基本的かつセキュリティ強化の効果が大きいファイル操作に限ってセキュリティ強化を実施することとした。

3.4 ポリシーのチューニング

学習モードでの運用の結果蓄積されていたポリシーを精査した結果、Web コンテンツや CGI 関連のポリシーをチューニングする必要があることが判明した。Apache の設定ファイルの読み込みなどは「決まりきった動作」であり、これまでに学習したポリシーだけで充分である。しかし、Web コンテンツや CGI の動作は多岐に渡り、適宜ワイルドカードを用いるなどのチューニングが必要となるのである。

静的な Web コンテンツについては、参照するディレクトリ以下の絶対パスをワイルドカードを用いてパターン化する作業を実施した。Apache が参照するドキュメントルート以下のファイルを再帰的に指定することが必要だが、TOMOYO Linux のワイルドカードには「このディレクトリ以下」という再帰的な指定は存在せず、ディレクトリ階層の深さごとに指定する点に注意が必要である。実際の指定では TOMOYO Linux のアクセス制御を記

[2] このあとリリースされた TOMOYO Linux バージョン 1.5.1 では、初期設定作成時にツールが正常に動作するかどうかのセルフチェック機構が追加された。

述するドメイン別ポリシーに直接パス名を記述することもできたが、今後のメンテナンス性を考慮し、例外ポリシーに複数のパス名をグループ化するマクロを定義しておき、ドメイン別ポリシーにはマクロ参照のみを記述することとした。

TOMOYO Linux の「全てのプログラムの実行でドメイン（＝アクセス制御の単位）を自動的に分ける」という性質から、CGI は Apache とは別のドメインで動作する。今回導入対象としたサーバでは、Web コンテンツを検索する CGI と、セミナーなどの参加を受け付けメールを送信する CGI が動作していたが、それぞれのドメインが学習結果として自動的に生成されていた。それぞれの CGI のドメインで要求されているリクエストから、ファイル名に揺らぎが生じる範囲をピックアップし、ワイルドカードを用いてパターン化する作業を実施した。

TOMOYO Linux の学習モードを用いることで、システム中でどのようなプログラムが動作しているかをポリシーを見ることで簡単に把握できるという副作用が得られた。CGI のポリシーをチューニングする際、通常の CGI ではあまり行わないアクセスを要求していることをポリシーの精査時に発見する[3]など、システムの挙動を把握するという意味でも有用であった。

ポリシーのチューニング作業はシステムやアプリケーションの再起動なしに行えるが、当初 TOMOYO Linux を導入したときに初期設定が生成されていなかった問題を解消するため、チューニングが一通り完了した時点でシステムの再起動を実施した。再起動後の TOMOYO Linux のメモリ消費は、冗長なポリシーがなくなったことにより 2MB 程度[4]にまで減少した。

3.5 確認モードでの運用

前述したチューニング作業が完了したのち、システム全体を学習モードから確認モードと呼ばれるモードに切り替えた。確認モードでは学習モードと同じくアクセス制御は実施されず、見かけの動作は通常の Linux と変わらない。学習モードとの違いは、ポリシーに違反するアクセスが発生した場合にそのアクセスがポリシーに追加されるのではなくログが出力されるという点である。

確認モードへの移行には特にシステムやアプリケーションの再起動は必要なく、実際に移行しても特に問題は発生しなかった。

確認モードで運用中に出力されたログのうち、Apache のログ以外をフィルタリングした

[3] この CGI は一般公開されていない自作 CGI で、CGI や Web サーバが通常アクセスしない場所にデータを書き込むことができる仕様になっていた。セキュリティ的な問題はないが、書き込みを行うことを TOMOYO Linux が検出してくれたことになる。学習モードを十分にかける効果を知ることとなった。

[4] TOMOYO Linux 開発チームによると、今回はシステム全体の挙動を学習させているが、特定のアプリケーションのみのポリシーに絞ることで、通常は 1MB 以内のメモリ消費に抑えることができるとのことである。

ところ、学習モードでの運用時には実行されていない新たな CGI が動作していることが判明し、ポリシーに追加した。

3.6 強制モードへの移行

2 週間ほどの確認モードでの運用ののち、実際のアクセス制御を実施する強制モードへ移行した。強制モードへの移行もほかのモード切替と同様、システムやアプリケーションの再起動は不要である。また、TOMOYO Linux はドメインごとに独立に制御モードを持つことができ、今回は Apache と Apache から起動される CGI のドメインを強制モードとし、その他のドメインは確認モードのまま運用を継続した。

4. サーバの運用実験

4.1 運用にあたっての検討事項

運用を開始する際には、セキュア OS による障害が発生した場合に早急に対応できる体制が課題となる。この課題を解決するために、TOMOYO Linux の出力するログを読み、Apache 関連のログが出力されると管理者にメールで通知するスクリプトを作成した。作成したスクリプトは強制モードでの運用開始と同時に cron ジョブとして登録した。これにより、Apache がポリシーに存在しないアクセスを要求し、そのアクセスが TOMOYO Linux により拒否されたことが 1 時間以内に管理者に通知される。

Apache のアップデートや Web コンテンツのディレクトリ構成の更新など、Web サーバ関連の更新時の対応についても留意が必要である。Web サーバのパッケージをアップデートする際には、TOMOYO Linux プロジェクトページに掲載されているパッケージのアップデート手順に従って行うこととした。Web コンテンツのディレクトリ構成の変更については、ポリシーを修正するための手順書 TOMOYO Linux 開発チームと共同で作成した。

4.2 実験中のログの状況

強制モードでの運用開始後 2 週間で、8 回のログメールが送信された。ログが送信されるたびに TOMOYO Linux 開発チームとともに中身を精査し、クラッキングによるアクセスでないことを確認の上、要求されたアクセスをポリシーに追加していった。実際に追加したアクセスの中身は、学習中には存在しなかった Web コンテンツへの読み込みアクセスや、PHP スクリプトによる特定のディレクトリへの書き込みアクセスであった。ポリシーの追加操作自体はこれまで行ってきたポリシーのチューニング作業と同様で特に難しい点はなかった。

4.3 パフォーマンスの変化

TOMOYO Linux 導入前と導入後では、体感できるほどのパフォーマンスの変化は生じなかった。TOMOYO Linux 開発チームによると、学習モードでの運用中はメモリ領域の確保など負荷の大きな処理が多いため、比較的大きなパフォーマンスの低下がありうるが、強制モードでの運用に移行すれば、体感できるほどの影響が生じることはまれである、とのことである。

強制モードでの安定運用時に TOMOYO Linux が使用するメモリは 2MB 程度であった。

今回の実験ではシステム全体のポリシーを学習させたため、かなり大きなポリシーをメモリ上に保持しているが、最近のサーバのメモリ搭載量ならば問題にならないほどのメモリ消費である。TOMOYO Linux 開発チームによると、保護する必要のないアプリケーションのポリシーを削除すればさらにメモリ消費は削減でき、通常 1MB を超えない程度に抑えることができる、とのことである。

4.4 トラブルの状況

学習や確認モードでの運用では通常の Linux と見かけ上の挙動がまったく同じなので、障害が発生することはそもそもまれである。今回の実験でも、学習と確認モードの環境では特に障害は発生しなかった。また、強制モードでの運用中でも、システムが提供するサービスが停止するような重大な障害は発生していない。

軽微な障害としては、強制モードでの運用中にいくつか許可すべきアクセスが拒否された事例が挙げられる。拒否されたアクセスの内容は、学習モード下では存在しなかったディレクトリへの読み込みアクセスや、新たに導入された CGI の設定ファイルへの書き込みアクセスなどであった。これらのアクセスを要求したクライアント側には、多くの場合 403 Forbidden という HTTP のエラーコードが返されている。

5. 今後のセキュア OS の普及に向けた知見

5.1 実験において実現された事項

今回の実験では Web サーバに特化したセキュリティ強化を実現した。これにより、Web サーバとそこから起動される CGI の動作が通常発生する範囲内に制限されている。たとえ Web サーバや CGI にセキュリティホールが発見され、悪意のユーザに攻撃を仕掛けられたとしても、ポリシーに記述した範囲外の動作が行えない状態を実現できている。

アクセス拒否が発生すれば必ず通知メールが送信されるように設定したことで、ポリシーに存在しない動作が要求されていないことが確認できる。通知が送られてこないことから、「クラッキングなどの想定外の動作が要求されていないこと」を確認することができ、サーバ管理者として安心感を得ることができている。

TOMOYO Linux によるセキュリティ強化の副作用として、導入初期段階の学習モードでの運用結果から、システムでどのようなプロセスが動作しているかを改めて把握することができた。学習結果を参照しながら、不必要なサービスを止める、アンインストールするなど、サーバ構成のスリム化を達成できた。

5.2 セキュア OS 化におけるポイント

セキュア OS を導入する際にポイントとなるのは、どこまで厳密にセキュリティ強化を実現するのかを見極める点にある。システム全体のポリシーを作りこみ強制モードで運用するのがセキュリティの観点から理想であるのは明らかであるが、導入・運用のコストから考えると現実的ではない。外部にサービスを公開しているサーバプログラムの動作を制限したり、特定の管理操作だけを実施できるようにログインシェルの動作を制限したり、というように、用途を絞ってセキュア OS のアクセス制御機能を利用するのが効果的である。

一般的な公開サーバであれば、インターネットに対してサービスを提供しているアプリケーションをセキュア OS で保護するだけで充分な効果を得られるであろう。悪意のあるユーザは、インターネット経由でアプリケーションの脆弱性を突いて乗っ取り等の攻撃を仕掛けるからである。たとえセキュア OS で保護しているアプリケーションが乗っ取られたとしても、被害はセキュア OS が許可した範囲内に限定することができる。

TOMOYO Linux の場合、とりあえずカーネルを導入して学習モードでの運用を開始するだけならば何も動作に影響を与えず、積み上げ式のセキュリティ強化が可能となっているため、まずは外部公開しているサービスを対象にセキュリティ強化を実現し、余裕が出てくれば更に別のサービスのセキュリティ強化を実現する、ということが可能である。また、セキュリティ強化の対象としないアプリケーションについても、学習モードで動作を学習

させることで、システムの挙動の把握、監視といった用途としても有用である。

5.3 予想されるトラブルへの対処

セキュア OS を導入した際に発生する代表的なトラブルは、本来許可しなければならないアクセスを拒否してしまう、というものである。重大なものではないが、今回の導入実験でも強制モードへの移行後に何度か、本来許可すべきアクセスを拒否してしまっている。セキュア OS 導入時に充分な試験を実施する、というのが予防的な対処であるが、これ以外にも、アクセス拒否が発生した時にすぐに対処できる体制を作っておく、というのが重要なポイントである。

今回の実験では、アクセス拒否が発生してから 1 時間以内に管理者にメールが送信される仕組みを作っており、この通知を受け取った管理者はただちに拒否されたアクセスの中身を精査し、必要な対処（ポリシーの追加、クラッキング被害の報告）を取ることができる。セキュア OS を導入したシステムの運用では、アクセス違反が発生した時の対応を整備することが重要である。

5.4 今回の実験で未確認の事項

セキュア OS は、今回の実験で実施したアクセス制御だけではなく、管理者権限の分割や、管理操作の詳細なロギングといった用途にも使用できる。これらの用途でセキュア OS を使用する際の導入手順や想定されるトラブルについては未確認である。

6. まとめ

6.1 実験を振り返って

（やすだ なお・JNSA／ディアイティ／サイバー大学）

「セキュア OS を使ってみたい」というのは、サーバ管理をしてみた方であれば、何割かの方々が考えてみたことではないだろうか。私ももう現役ではないが、昔からある種の憧れも含めて、実際に動かしてみたいと考えていた。以前は Trusted OS と呼ばれ、軍事や基幹産業、金融系で使われることが想定され、高価であり簡単に触れられるものではなかった。いわば遠巻きにしてつぶらな瞳で眺め、想像をめぐらせていたのが実情であった。また、当時は強制アクセス制御(Mandatory access control)についての解説はあっても、実際の実装がなく、なかなか直接触れることができなかったこともある。色々なアルゴリズムの違いを実感することも難しかった。そのうち、Solaris 10 が一部 Trusted Solaris の機能を標準装備するようになり、米国の NSA（National Security Agency）が中心となって開発された SELinux (Security-Enhanced Linux) が公開され、MAC として TE や RBAC が実装された。無料でエッセンスを試してみることができるようになり、ずっと身近な存在になると共に、このあたりから「セキュア OS」と呼ばれるようになってきたようだが、やはりまだポリシーと呼ばれるデータを正しく設定しておかないと、今まで普通の Linux サーバで動いていたサーバソフトやアプリケーションソフトを動かすのは結構準備が要るというハードルがあった。

JNSA でも Web サーバと Mail サーバを中心にしてサーバの運用管理を行っているが、よくある組織と同じように、この方面の専従者を確保できているわけではなく、予防システムについてセキュア OS を利用してみたいという希望は大分以前から持っていた。ただ、実際にサービスをしているサーバなので、長期間の停止は難しく、できるだけ切れ目のないサービスを続けながらの移行という課題を抱えていた。これは、普通に良くある小規模のサーバをセキュア OS 化したいときと同じ悩みであるといえるだろう。JNSA のサーバは大変こじんまりとしたごく普通の良くあるサーバーだが、この環境でうまくいくようであれば、他でも同じように実稼動しているサーバをセキュア OS 化するためのひとつの見本となるのではないか、という JNSA 的な活動の一環としても意義があるだろうと思われた。

このような経緯で、JNSA の Web サーバをセキュア OS 化するための作業を、技術部会のセキュアＯＳ普及促進 WG を中心として検討していただき、TOMOYO Linux の開発者である NTT データの原田氏、半田氏、武田氏をメンバーに加え、JNSA の Web サーバのセキュア OS 化を具体的に進めることになった。設定の詳細については、他の項をご覧になっていただきたいが、思ったよりスムーズに行ったように感じている。もともと JNSA の Web サーバは Red Hat Linux が動作していたので、TOMOYO Linux を追加インストールするだけで、学習モードでのチェックができるようになった。あとは強制モードに移行するまでに、実際に動作しているプロセスの一覧やアクセスファイルをチェックし、問題点をつぶ

す作業を行った。今まで動作しているサーバなので、実際にどのようなプログラムが動作しているのかを完全に把握することは案外難しい。本来はきちんと文書化されていなければならないのであろうが、ごく一般的な問題例としても課題を認識したかった、ということもあり、敢えて問題点を再認識する方法を試行してみた。この結果、忘れていた設定や動作を思い出させてくれる、という、再ドキュメント化ツールとしても TOMOYO Linux は役に立つという副作用を認識することができた。

今回の TOMOYO Linux によるセキュア OS 化を実施してみて感じたのは、「案外簡単だった」ということである。それなりの効果も出てきているようだが、予定外のプロセス実行をレポートしてくれるということが、普段の運用管理面でも思った以上に役に立つことも確認できた。この報告書を読まれた皆さんもチャレンジしてみていただきたい。

6.2 TOMOYO Linux を提供してみて

（原田 季栄・NTT データ）

実験を振り返ってみると、特に問題らしい問題には遭遇せず、作業自体は当初予定していた以上にスムーズに完了した。提供元としては一安心したわけであるが、「稼働中の実サーバに導入することによりセキュア OS 普及促進上の問題点を洗い出し、その結果を白日のもとにさらすことにより日本中の管理者に参考にしてもらおう」と意気込んで実験に参加した立場からは、正直やや物足りない結果となったのも事実である。

発生した「問題」については主に 2 件となる。ひとつは誤って異なるバージョンをインストールしてしまったことによる初期化ミスで、対処としては単純にインストール作業をやり直しただけだが、開発元として想定していない事態（オペレーション）だったため、何が起こっているかの判断、および「システムが現在どのような状態になっているか」の解明には少なからぬ時間を要した。現在のバージョンではインストールスクリプトのエラー処理を追加しており、本実験により改善された機能となる。もうひとつは保護開始後のポリシー違反で、TOMOYO Linux によるサーバの保護を有効にしてまもないある日、メールにて「Apache からポリシーに定義していない CGI が起動され、怪しい動きをしている」通知を受信した。「すわ早速 JNSA サーバにクラッキングか？」と関係者は大いに色めきたったが、その後の調査により JNSA 安田様が書かれた「多目的用途に使える CGI」であることが判明している。対処としては、違反ログの内容をもとに用途とアクセス範囲に問題ないことを安田様に確認いただいてから必要なアクセス許可をポリシーに追加している（ポリシー違反の検知は自動的に行えても、その違反が本当に違反かどうかはサーバの管理者でなければ行えない）。このように「サーバ管理者自身がその存在を忘れているような場合」でも、その振る舞いをもれなく忠実にトレースするのは、まさに TOMOYO Linux の特長である学習機能による効果であり、今回の実験ははからずも TOMOYO Linux が管理者の誤操作や内部犯による情報の持ち出し等にも効力を持つことを証明した形となったと考えている。

TOMOYO Linux の開発は、「セキュア OS が普及しないのは使いにくいから」という仮説に基づき、「使いこなせて安全な Linux」を実現しようとして始めたものだった。取り組みを始めた 2003 年当時には SELinux を含め「パッチを当ててインストールする」という作業が必要であり、またそもそも「セキュア OS」に関する日本語の情報がほとんど存在していなかったわけだが、現在その状況は劇的に改善されている。SELinux は Linux 標準カーネルに取り込まれており、それ以外にも TOMOYO Linux や AppArmor など従来のシステム管理者でも十分に使える実装があり、雑誌や Web など情報も読み切れないほど豊富に存在している。つまり、セキュア OS の普及を阻害する要因は、もはや「使いにくさ」ではなく、それ以外にあることは明白である。今回の実験について、提供側としての視点からいうと、最大のネックは「サーバ管理者の稼働の確保」にあった。実験には約半年間の期間を要したが、実際に行った作業は、ほとんど一週間程度でしかなく、運用にはいってからも特に問題も生じていない。作業にあたり質問も特になく、大きなトラブルも皆無だった。

JNSA 様に限らず、管理を担当されている方々の多くは複数のサーバ、システムを担当されていることが多く、日々の業務で余裕がないのが実情かもしれない。しかし、不幸にして被害（事故、攻撃）を受けるとさらにとほうもない時間がかかり、「それは起こりえないことではない」ということを強調しておきたい。被害が起こってからでは遅いのである。「まだ自分はサーバのセキュリティ強化の必要性を感じていない」という方々にも是非一度 TOMOYO Linux を走らせて、サーバの振る舞いを計測してもらいたい。その上で Web サーバ等の保護したい部分＝ドメインを切り出して保護すればもちろん最上である。その他、組込み関係の用途でも使われていないファイルを抽出して削除するなどの応用が可能であるし、SELinux のポリシーの策定に活用しても良い。今回の実験が、国内におけるセキュア OS の普及促進の一助となることを心から願ってやまない。

付録

（1）実験で使用したポリシー（抜粋）

注）網掛け下線部は前行からの継続行なので注意。

```
<kernel> /usr/sbin/httpd

1 /bin/sh
6 /dev/null
4 /dev/urandom
4 /etc/group
4 /etc/host.conf
4 /etc/hosts
4 /etc/httpd/cert/\*/\*
4 /etc/httpd/conf.d/manual.conf
4 /etc/httpd/conf.d/perl.conf
4 /etc/httpd/conf.d/php.conf
4 /etc/httpd/conf.d/python.conf
4 /etc/httpd/conf.d/ssl.conf
4 /etc/httpd/conf.d/virtualhost.conf
4 /etc/httpd/conf.d/welcome.conf
4 /etc/httpd/conf/httpd.conf
4 /etc/httpd/conf/magic
4 /etc/mime.types
4 /etc/mtab
4 /etc/nsswitch.conf
4 /etc/odbcinst.ini
4 /etc/passwd
4 /etc/php.d/gd.ini
4 /etc/php.d/ldap.ini
4 /etc/php.d/mbstring.ini
4 /etc/php.d/mysql.ini
4 /etc/php.d/odbc.ini
4 /etc/php.ini
4 /etc/protocols
4 /etc/resolv.conf
4 /etc/services
4 /proc/sys/kernel/ngroups_max
4 /usr/lib/httpd/modules/libphp4.so
4 /usr/lib/httpd/modules/mod_access.so
4 /usr/lib/httpd/modules/mod_actions.so
4 /usr/lib/httpd/modules/mod_alias.so
4 /usr/lib/httpd/modules/mod_asis.so
4 /usr/lib/httpd/modules/mod_auth.so
4 /usr/lib/httpd/modules/mod_auth_anon.so
4 /usr/lib/httpd/modules/mod_auth_dbm.so
4 /usr/lib/httpd/modules/mod_auth_digest.so
4 /usr/lib/httpd/modules/mod_auth_ldap.so
4 /usr/lib/httpd/modules/mod_autoindex.so
4 /usr/lib/httpd/modules/mod_cache.so
4 /usr/lib/httpd/modules/mod_cern_meta.so
4 /usr/lib/httpd/modules/mod_cgi.so
4 /usr/lib/httpd/modules/mod_dav.so
```

```
4 /usr/lib/httpd/modules/mod_dav_fs.so
4 /usr/lib/httpd/modules/mod_deflate.so
4 /usr/lib/httpd/modules/mod_dir.so
4 /usr/lib/httpd/modules/mod_disk_cache.so
4 /usr/lib/httpd/modules/mod_env.so
4 /usr/lib/httpd/modules/mod_expires.so
4 /usr/lib/httpd/modules/mod_file_cache.so
4 /usr/lib/httpd/modules/mod_headers.so
4 /usr/lib/httpd/modules/mod_imap.so
4 /usr/lib/httpd/modules/mod_include.so
4 /usr/lib/httpd/modules/mod_info.so
4 /usr/lib/httpd/modules/mod_ldap.so
4 /usr/lib/httpd/modules/mod_log_config.so
4 /usr/lib/httpd/modules/mod_mem_cache.so
4 /usr/lib/httpd/modules/mod_mime.so
4 /usr/lib/httpd/modules/mod_mime_magic.so
4 /usr/lib/httpd/modules/mod_negotiation.so
4 /usr/lib/httpd/modules/mod_perl.so
4 /usr/lib/httpd/modules/mod_proxy.so
4 /usr/lib/httpd/modules/mod_proxy_connect.so
4 /usr/lib/httpd/modules/mod_proxy_ftp.so
4 /usr/lib/httpd/modules/mod_proxy_http.so
4 /usr/lib/httpd/modules/mod_python.so
4 /usr/lib/httpd/modules/mod_rewrite.so
4 /usr/lib/httpd/modules/mod_setenvif.so
4 /usr/lib/httpd/modules/mod_speling.so
4 /usr/lib/httpd/modules/mod_ssl.so
4 /usr/lib/httpd/modules/mod_status.so
4 /usr/lib/httpd/modules/mod_suexec.so
4 /usr/lib/httpd/modules/mod_userdir.so
4 /usr/lib/httpd/modules/mod_usertrack.so
4 /usr/lib/httpd/modules/mod_vhost_alias.so
4 /usr/lib/perl5/5.8.5/AutoLoader.pm
4 /usr/lib/perl5/5.8.5/Carp.pm
4 /usr/lib/perl5/5.8.5/Exporter.pm
4 /usr/lib/perl5/5.8.5/i386-linux-thread-multi/CORE/libperl.so
4 /usr/lib/perl5/5.8.5/i386-linux-thread-multi/Config.pm
4 /usr/lib/perl5/5.8.5/i386-linux-thread-multi/DynaLoader.pm
4 /usr/lib/perl5/5.8.5/strict.pm
4 /usr/lib/perl5/5.8.5/vars.pm
4 /usr/lib/perl5/5.8.5/warnings.pm
4 /usr/lib/perl5/5.8.5/warnings/register.pm
4 /usr/lib/php4/gd.so
4 /usr/lib/php4/ldap.so
4 /usr/lib/php4/mbstring.so
4 /usr/lib/php4/mysql.so
4 /usr/lib/php4/odbc.so
4 /usr/lib/python2.3/UserDict.py
4 /usr/lib/python2.3/UserDict.pyc
4 /usr/lib/python2.3/bdb.py
4 /usr/lib/python2.3/bdb.pyc
4 /usr/lib/python2.3/cmd.py
4 /usr/lib/python2.3/cmd.pyc
4 /usr/lib/python2.3/codecs.py
4 /usr/lib/python2.3/codecs.pyc
4 /usr/lib/python2.3/copy_reg.py
4 /usr/lib/python2.3/copy_reg.pyc
```

```
4 /usr/lib/python2.3/encodings/__init__.py
4 /usr/lib/python2.3/encodings/__init__.pyc
4 /usr/lib/python2.3/encodings/aliases.py
4 /usr/lib/python2.3/encodings/aliases.pyc
4 /usr/lib/python2.3/encodings/ascii.py
4 /usr/lib/python2.3/encodings/ascii.pyc
4 /usr/lib/python2.3/lib-dynload/cStringIO.so
4 /usr/lib/python2.3/lib-dynload/strop.so
4 /usr/lib/python2.3/lib-dynload/syslogmodule.so
4 /usr/lib/python2.3/lib-dynload/timemodule.so
4 /usr/lib/python2.3/linecache.py
4 /usr/lib/python2.3/linecache.pyc
4 /usr/lib/python2.3/os.py
4 /usr/lib/python2.3/os.pyc
4 /usr/lib/python2.3/pdb.py
4 /usr/lib/python2.3/pdb.pyc
4 /usr/lib/python2.3/posixpath.py
4 /usr/lib/python2.3/posixpath.pyc
4 /usr/lib/python2.3/pprint.py
4 /usr/lib/python2.3/pprint.pyc
4 /usr/lib/python2.3/re.py
4 /usr/lib/python2.3/re.pyc
4 /usr/lib/python2.3/repr.py
4 /usr/lib/python2.3/repr.pyc
4 /usr/lib/python2.3/site-packages/japanese.pth
4 /usr/lib/python2.3/site-packages/japanese/__init__.py
4 /usr/lib/python2.3/site-packages/japanese/__init__.pyc
4 /usr/lib/python2.3/site-packages/japanese/aliases/__init__.py
4 /usr/lib/python2.3/site-packages/japanese/aliases/__init__.pyc
4 /usr/lib/python2.3/site-packages/mod_python/__init__.py
4 /usr/lib/python2.3/site-packages/mod_python/__init__.pyc
4 /usr/lib/python2.3/site-packages/mod_python/apache.py
4 /usr/lib/python2.3/site-packages/mod_python/apache.pyc
4 /usr/lib/python2.3/site-packages/pygtk.pth
4 /usr/lib/python2.3/site.py
4 /usr/lib/python2.3/site.pyc
4 /usr/lib/python2.3/sre.py
4 /usr/lib/python2.3/sre.pyc
4 /usr/lib/python2.3/sre_compile.py
4 /usr/lib/python2.3/sre_compile.pyc
4 /usr/lib/python2.3/sre_constants.py
4 /usr/lib/python2.3/sre_constants.pyc
4 /usr/lib/python2.3/sre_parse.py
4 /usr/lib/python2.3/sre_parse.pyc
4 /usr/lib/python2.3/stat.py
4 /usr/lib/python2.3/stat.pyc
4 /usr/lib/python2.3/string.py
4 /usr/lib/python2.3/string.pyc
4 /usr/lib/python2.3/traceback.py
4 /usr/lib/python2.3/traceback.pyc
4 /usr/lib/python2.3/types.py
4 /usr/lib/python2.3/types.pyc
4 /usr/lib/python2.3/warnings.py
4 /usr/lib/python2.3/warnings.pyc
4 /usr/share/file/magic.mime
4 /usr/share/mysql/charsets/Index.xml
2 /var/log/httpd/access_log
```

```
2 /var/log/httpd/error_log
2 /var/log/httpd/non_access_log
2 /var/log/httpd/non_error_log
2 /var/log/httpd/ssl_access_log
2 /var/log/httpd/ssl_error_log
2 /var/log/httpd/ssl_request_log
2 /var/log/httpd/wg_access_log
2 /var/log/httpd/wg_error_log
2 /var/run/httpd.pid
1 /var/sample/contents/cgi-bin/namazu.cgi
1 /var/sample/contents/sslcgi/DUMPshow.cgi
1 /var/sample/contents/sslcgi/ToMail.cgi
6 /var/sample/web/vol2/html/templates_c/\*
6 @WEB_CACHE
4 @WEB_CONTENTS
allow_create /var/run/httpd.pid
allow_create /var/sample/web/vol2/html/templates_c/\*
allow_create @WEB_CACHE
allow_link @WEB_CACHE @WEB_CACHE
allow_rename /var/sample/web/vol2/html/templates_c/\* /var/sample/web/vol2/html/templates_c/\*
allow_rename @WEB_CACHE @WEB_CACHE
allow_symlink @WEB_CACHE
allow_truncate /var/run/httpd.pid
allow_truncate /var/sample/web/vol2/html/templates_c/\*
allow_truncate @WEB_CACHE
allow_unlink /var/run/httpd.pid
allow_unlink @WEB_CACHE
use_profile 3

<kernel> /usr/sbin/httpd /bin/sh

6 /dev/tty
4 /etc/mtab
4 /etc/nsswitch.conf
4 /etc/passwd
1 /usr/sbin/sendmail.postfix
use_profile 3

<kernel> /usr/sbin/httpd /bin/sh /usr/sbin/sendmail.postfix

4 /etc/group
4 /etc/host.conf
4 /etc/hosts
4 /etc/nsswitch.conf
4 /etc/passwd
4 /etc/postfix/main.cf
4 /etc/resolv.conf
1 /usr/sbin/postdrop
use_profile 3

<kernel> /usr/sbin/httpd /bin/sh /usr/sbin/sendmail.postfix /usr/sbin/postdrop

4 /etc/group
4 /etc/host.conf
4 /etc/hosts
4 /etc/nsswitch.conf
4 /etc/passwd
```

```
4 /etc/postfix/main.cf
4 /etc/resolv.conf
4 /usr/share/zoneinfo/UTC
6 /var/spool/postfix/maildrop/\*
2 /var/spool/postfix/public/pickup
allow_create /var/spool/postfix/maildrop/\*
allow_rename /var/spool/postfix/maildrop/\* /var/spool/postfix/maildrop/\*
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/cgi-bin/namazu.cgi

4 /etc/mtab
4 /usr/local/lib/libnmz.so.7.0.0
4 /var/sample/contents/cgi-bin/.namazurc
2 /var/sample/contents/namazuindex/NMZ.slog
4 @NAMAZU_INDEX_FILES
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/DUMPshow.cgi

4 /dev/urandom
4 /usr/bin/perl
4 /usr/lib/perl5/5.8.5/i386-linux-thread-multi/CORE/libperl.so
1 /usr/local/bin/nkf
4 /var/sample/contents/sslcgi/DUMPshow.cgi
4 /var/sample/contents/sslcgi/jcode.pl
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/DUMPshow.cgi /usr/local/bin/nkf

use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi

1 /bin/date
1 /bin/sh
4 /dev/urandom
4 /usr/bin/perl
4 /usr/lib/perl5/5.8.5/i386-linux-thread-multi/CORE/libperl.so
1 /usr/local/bin/nkf
6 /var/sample/contents/vol1/\*
4 /var/sample/contents/sslcgi/ToMail.cgi
4 /var/sample/contents/sslcgi/ToMail.conf
4 /var/sample/contents/sslcgi/jcode.pl
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /bin/date

use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /bin/sh

1 /bin/date
6 /dev/tty
4 /etc/mtab
4 /etc/nsswitch.conf
4 /etc/passwd
```

```
1 /usr/local/bin/nkf
1 /usr/sbin/sendmail.postfix
2 /var/sample/contents/vol1/\*
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /bin/sh /bin/date

use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /bin/sh /usr/local/bin/nkf

use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /bin/sh
 /usr/sbin/sendmail.postfix

4 /etc/group
4 /etc/host.conf
4 /etc/hosts
4 /etc/nsswitch.conf
4 /etc/passwd
4 /etc/postfix/main.cf
4 /etc/resolv.conf
1 /usr/sbin/postdrop
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /bin/sh
 /usr/sbin/sendmail.postfix /usr/sbin/postdrop

4 /etc/group
4 /etc/host.conf
4 /etc/hosts
4 /etc/nsswitch.conf
4 /etc/passwd
4 /etc/postfix/main.cf
4 /etc/resolv.conf
4 /usr/share/zoneinfo/UTC
6 /var/spool/postfix/maildrop/\*
2 /var/spool/postfix/public/pickup
allow_create /var/spool/postfix/maildrop/\*
allow_rename /var/spool/postfix/maildrop/\* /var/spool/postfix/maildrop/\*
use_profile 3

<kernel> /usr/sbin/httpd /var/sample/contents/sslcgi/ToMail.cgi /usr/local/bin/nkf

4 /var/sample/contents/vol1/\*
use_profile 3
```

```
path_group WEB_CONTENTS /var/sample/web/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*
```

```
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group WEB_CONTENTS /var/sample/web/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*
path_group NAMAZU_INDEX_FILES /var/sample/contents/namazuindex/\*
path_group WEB_CACHE /var/sample/web/vol2/html/cache/\*
```

（2）スクリプト作成例

ログを監視して httpd 関連のエラーがあればメールを送信するスクリプト。
冒頭の"user?@example.com"に電子メールの送付先アドレスを設定した上で実行権限を与え、/etc/cron.hourly ディレクトリに置く。

```
#!/bin/sh

MAILTO="user1@example.com user2@example.com user3@example.com"
SUBJECT="ccs-logwatch"

cd /var/log/tomoyo || exit 1
[ -s reject_log.txt ] || exit 1

mv reject_log.txt reject_log.tmp

grep -A 2 -B 1 '^<kernel> /usr/sbin/httpd' reject_log.tmp > /tmp/reject_log.httpd
[ -s /tmp/reject_log.httpd ] && mail -s ${SUBJECT} ${MAILTO} < /tmp/reject_log.httpd

unlink /tmp/reject_log.httpd

cat reject_log.tmp >> reject_log.archived
unlink reject_log.tmp

exit 0
```

（3）TOMOYO Linux の情報源

- TOMOYO Linux オフィシャルサイト
  http://tomoyo.sourceforge.jp/
- 技術評論社 SoftwareDesign 2007 年 1 月号～12 月号連載 「TOMOYO Linux の世界」
  http://tomoyo.sourceforge.jp/wiki/?WorldOfTomoyoLinux
- 技術評論社 SoftwareDesign 2008 年 4 月号～5 月号連載「TOMOYO Linux の歩き方」
- TOMOYO Linux LiveCD
  http://tomoyo.sourceforge.jp/wiki/?TomoyoLive