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

2006－ＤＤ－56（５）
2006／7／28

ビジネス・プロセスベースの強制アクセス制御による権限管理方法とその実装方法
－SELinux の偽装機能を用いて実装可能－

岡上 智也† 小野 諭‡

† 工学院大学大学院 工学研究科情報学専攻 〒163-8677 東京都新宿区西新宿 1-24-2
‡ 工学院大学 情報学部 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: † em06003@ns.kogakuin.ac.jp, ‡ s-ono@cc.kogakuin.ac.jp

あらまし 企業情報システムは効率化や柔軟性などを目指し，Web Service を活用する方向に向かっている．そこでは会計規定へのコンプライアンスなどにより，業務遂行上必須な最小特権の設定とその強制が必要となる．本論文ではビジネス・プロセスを考慮して，リクエスト毎に最小特権を付与する新たな方式を提案し，SELinux の権限管理機能を用いた実装方法について述べる．本方式を用いると，従来よりも細かい粒度で，かつ業務に即した権限管理が可能となる．

キーワード ウェブサービス，最小特権，ビジネス・プロセス，偽装，強制アクセス制御，SELinux

The MAC-based Authorization Management Method Utilizing Business Process Contexts
－Implementable Using SELinux Impersonation Feature－

Tomoya Okaue† Satoshi Ono‡

† Informatics, Kogakuin University Graduate School 1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo, 163-8677 Japan
‡ Faculty of Informatics Kogakuin University 1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo, 163-8677 Japan

E-mail: † em06003@ns.kogakuin.ac.jp, ‡ s-ono@cc.kogakuin.ac.jp

Abstract The business information system is going into the direction which utilizes Web Service aiming at increase in efficiency, flexibility, etc. In order to perform a task with the compliance to accountability, etc. A setup and its compulsion of the indispensable least privilege are needed on etc. there. In this paper, in consideration of a business process, the new system which gives the least privilege for every request in proposed, and the Implementation method using the security function of SELinux is described. If this system is used, the authorization management which is authorization with finer granularities with better business compliance will be attained.

Keyword Web Service, Least Privilege, Business Process, Impersonation, MAC, SELinux

1. はじめに

企業情報システムは業務の効率化や柔軟性，拡張性を求めて，Web Service（WS）[1]を使ったものへと急速に移行している．そこでは電子情報の信頼性を確保することが問題となり，コンプライアンスにより電子情報信頼性の確保が義務になろうとしている．

この WS を使った企業情報システムにおいて，最小特権[2]で処理を行うことにより一貫した方法で電子情報を生成したことを主張し，電子情報の信頼性を確保する方法がある．これは処理を行う主体に対して必要最小限の権限のみを付与することと，最小特権で処理を行っていたことを事後的に第三者が検証可能なことにより実現する．現在の WS では，最小特権を実現するために認証されたプリンシパル[3]が持つロールを用いて，各 Web Application（WA）でアクセス制御を行う．この方式には，権限下方硬直性の問題とアクセス制御記述の分散化の問題があった．

本論文ではビジネス・プロセス（BP）[4]を考慮した，業務に即した効率のよいロールの決定方法を提案する．この方式では，プリンシパルのロールと BP 上のコンテキストの２つから BP を考慮したロールを新たに生成し，各 WA はそのロールを用いて動作する．これにより，下方硬直性の緩和とアクセス制御記述の局所化を行う．また，強制アクセス制御（MAC）[5]である SELinux[6]を実装環境として用いることにより，事後的に第三者が検証可能となる最小特権を実現する．

章２では WS を使った企業情報システムのアーキテクチャについて述べ，そこで最小特権を実現するための要件について述べる．章３では WA での偽装による動作権限主体を決定するロールについて述べ，提案方式について述べる．章４では最小特権を実現するプラットフォームとして SELinux に注目し，SELinux について評価する．章５では章３の提案方式を，SELinux を用いて実装する方法について述べ，実際に自走した

機能について述べる．章6では全体についてのまとめと，今後の課題について述べる．

2. WSを使った企業情報システムのアーキテクチャと最小特権

現在の企業情報システムは，WSを使って既存のサービスを組合せ，柔軟で拡張性のある業務活動に即したシステムを構築している．このWSを使った企業情報システムとして，次のようなアーキテクチャがある（図1）．

このアーキテクチャにおける基本的な処理手順について述べる．

①ユーザはポータルよりWSにログインする．
②ユーザはSSO[7]システムにより認証を受ける．
③ユーザは実行エンジンに処理を依頼し，実行エンジンはユーザの処理を開始する．
④実行エンジンは，SSOシステムからユーザの認証情報を得る．ここでの技術としてSAML[8]があり，実行エンジンはユーザの認証情報であるセキュリティコンテキスト（SC）[9]を得る．
⑤実行エンジンはポリシーディレクトリを参照し，ユーザの要求に対してどのサービスに依頼するかを決定する．このフロー決定のポリシーとしてBPがある．BPは「どのように業務を行うかについて決めたルール」であり，これに沿って業務を進めていく．WSにおいて，BPのフローを記述する言語としてWS-BPEL[10]がある．
⑥実行エンジンはポリシーに従ってWAに処理のリクエストを送り，結果を得る．この時実行エンジンは，WAへのリクエストにSCを貼り付け，WAではこのSCを用いてアクセス制御を行う．
⑦各WAは処理を行うために，様々なDBに対して参照・更新を行う．
⑧実行エンジンはユーザの処理に必要となる全てのWAへの処理を依頼し，最終的な結果を受け取るとポータルに結果を返す．
⑨ユーザに結果が返される．

これらのプラットフォームとして.NET[11]やJ2EE[12]がある．

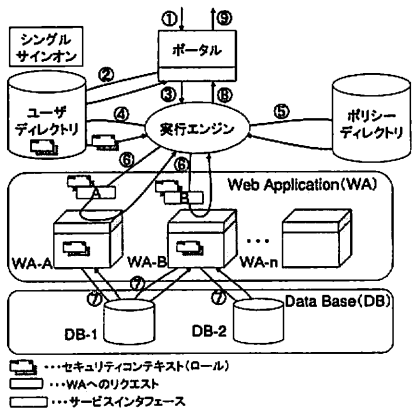

図1　WSを使った企業情報システムのアーキテクチャ

このアーキテクチャにおいて電子情報の信頼性を確保するための方法として，最小特権があることについては述べた．WSを使った企業情報システムにおいて最小特権を実現するためには，次の要件がある（表1）．

表1　WSを使った企業情報システムにおいて最小特権を実現するための要件

要件ⅰ.	処理を行う主体に対して，必要最小限の権限のみを付与すること	
	要件ⅰ-1.	全権アカウント権限からサービスアカウント権限への権限の制限機能
	要件ⅰ-2.	信頼コードを経由した保護データへのアクセス機能
	要件ⅰ-3.	サービスアカウント権限からリクエスト権限への偽装機能
要件ⅱ.	最小特権で業務を行っていたことを，事後的に第三者が検証可能なこと	
	要件ⅱ-1.	アクセス制御記述を強制し，アクセス制御記述の一貫性を保障すること
	要件ⅱ-2.	詳細な記録(ログ)を取得し，処理環境における証拠を確保すること
	要件ⅱ-3.	アクセス制御記述を宣言的に行うこと

ⅰ-1～ⅰ-3の技術として，Capability[13]，Code Access Security[14]，Impersonation[15]がある．

また，要件ⅰ-3の機能である偽装機能を使う場合，リクエストに貼り付いているSCを用いる．偽装したサービス処理の動作を決定するSCが次の要件を満たすことにより，偽装機能を使って必要最小限の権限で動作することができる．

要件ⅰ-3(a)．必要最小限の権限にするために，SCの権限を増やしたり減らしたりすることが容易であること
要件ⅰ-3(b)．SCの権限に対するアクセス制御記述を局所的に行う
要件ⅰ-3(c)．ユーザ環境のワークフローを考慮した権限を付与する
要件ⅰ-3(d)．SCが安定している

要件ⅰ-3(a)～i-3(d)は，要件ⅰ-3の偽装したサービス処理の動作権限主体を決定する処理依頼者のSCの設計の要件として対応付けができる．また，要件ⅰとⅱおよびⅰ-1～ⅱ-3は実際に要件ⅰ-3(a)～ⅰ-3(d)を実装する環境に求められる要件として対応付けができる．

3. 偽装による動作権限主体の決定方法

3.1. RBACによるプリンシパルが持つロール

最小特権を実現する機能にリクエスト権限への偽装機能（要件ⅰ-3）がある．偽装によるサービスの動作権限主体を決定するユーザのSCは，章2の要件①～④を満たすことにより必要最小限の権限を付与することができることについては述べた．ここでは，このユーザのSCの設計について述べる．

ユーザのSCの決定技術としてRBAC[16]がある．RBACではSCとして，ユーザアイデンティティ（以下ユーザID）が持つロールを用いる．ロールとはユーザのWSでの役割のことである．このロールはプリンシパル（ここではユーザアイデンティティとロールのセットをプリンシパルと呼ぶ）が持っており，SSO認証時にユーザIDに付与される．この方式では全ての

WA において，このロールを用いて偽装機能を使うことになる．SC にユーザのロールを用いることによるメリットとして，柔軟なユーザ変化への対応が容易であることが挙げられる．WS を使うユーザが変わっても，ユーザ毎に WA における設定はそのままでいいため，実際に WA の ACL で記述される SC の安定性に優れている．

要件ｉ-3(a)～ｉ-3(d)に対する RBAC の達成度について評価していく．要件ｉ-3(a)は満たしていない．BP の変化により WS の構成が変わり，それに伴って WS におけるプリンシパルが持つロールの権限も変わってくる．例えば，BP 変化に伴って新しい DB へのアクセスが必要になると，新しい DB へのアクセス権をプリンシパルが持つロールに与える．逆に DB へのアクセスが不要になったら，最小特権上その DB へのアクセス権をロールから削除する必要が出てくる．そのため，WS におけるロールの権限は BP の変化によって増えたり減ったりする．しかし，RBAC によるロールでは権限を減らす場合，今まで正常に動作していたビジネスシステムが正常に動作しなくなる可能性がある．WS のように頻繁に変化するものの場合ロールへの権限も複雑となり，一層ビジネスシステムの停止を恐れて権限の削除を行うことができない環境に向かっていく．これでは権限を増やす方向ばかりに行ってしまい，最小特権を阻害してしまう．このように，権限の下方硬直性の問題を抱える．要件ｉ-3(b)は満たしていない．BP の変化により，プリンパルが持つロールの権限を変更する場合，WA の ACL 以外の場所でロールの権限にかかわるアクセス制御記述が行われている．これはプリンシパルロールが BP のフローを考慮していないため，アプリケーションがフローに依存するアクセス制御記述を行っているためである．このアクセス制御記述の分散化は，権限の変更に伴う手間の増大や検証のしづらさを生じさせており，最小特権を阻害する問題としてある．要件ｉ-3(c)は満たしていない．詳細は[17]を参照されたい．要件ｉ-3(d)は満たしている．前述のようにプリンシパルが持つロールはユーザ ID の役割であり，この役割は変化しづらく安定していると言える．このように RBAC には，権限の下方硬直性とアクセス制御記述の分散化といった，最小特権を阻害する問題がある．

3.2. ビジネス・プロセスを考慮した BP ロール

そこで本論文ではビジネス・プロセスを考慮し，業務に即した効率のよいロールの決定方法を提案する．本方式では，WS のアーキテクチャに新しく次の要素を追加する．

■BP コンテキスト

どの BP のどのプロセスで，どのオペレーションを行う処理なのかを知ることができるコンテキスト．

■BP ロール

BP の位置やオペレーションとそれを実行するプリンシパルを考慮したロール．プリンシパルと BP コンテキストから決まる．

■実行エンジンが持つ BP ロールへのロールマッピング機能

BP ロールを生成するため，実行エンジンがプリンシパルロールと BP コンテキストにより，一意に決める BP ロールへマッピングする機能を追加する．

■実行エンジンへの Delegation 権限の付与

通常の実行エンジンは，プリンシパルのロールである SC に何も変更を加えずにリクエストに貼り付けている．今回加えるロールマッピング機能では，実行エンジンがロールに変更を加えてリクエストを送る．そのための権限が Delegation 権限であり，この権限を実行エンジンへ付与する．

それでは本方式を説明する．ユーザが実行エンジンに処理を依頼する．まず実行エンジンはリクエストを出す WA を決定した後に，ポリシーディレクトリに格納されている BP コンテキストを参照して，現在どの BP の，どのプロセスの，どのオペレーションを行うのかを解釈し，その位置の BP コンテキストを受け取る．次に実行エンジンは，プリンシパルが持つロールと BP コンテキストの組合せにより BP ロールを生成し，リクエストに貼り付ける SC をプリンシパルロールから BP ロールにマッピングする．WA では BP ロールに対しての ACL を持ち，BP ロールに許可された権限に偽装されて動作することになる．これらの一連の動作はリクエスト毎に行われる．これによって，リクエスト毎に BP を考慮したアクセス制御記述を行う．

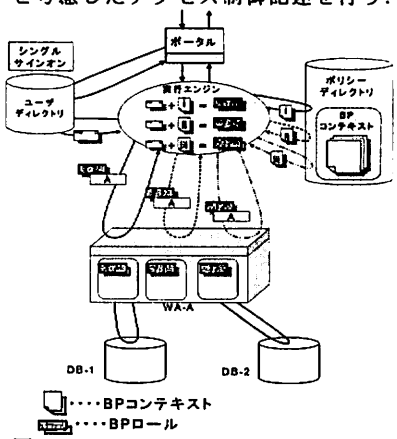

図２　BP ロールへのロールマッピング

この本提案方式では BP ロールに権限を割り当てるため，プリンシパルが持つロールへの権限に変更を加えない．そのため，BP の変化に伴うプリンシパルロールへの権限変更が生じたとき，BP 変更に沿った BP ロールを生成してアクセス制御記述を行うことができる．また，今までプリンシパルロールが必要だった権限がいらなくなっても，対応していた BP ロールへマッピングを行うことができないようにするだけで権限の削除を行うことができる．これにより，プリンシパルロールで行っていた場合の権限下方硬直性の問題を緩和する．また，BP ロールにワークフロー上の制御情報を記述することにより，WA で行い，分散していたアクセス制御記述を BP ロールに局所化する．

要件ｉ-3(a)～ｉ-3(d)に対する本提案方式の達成度について評価していく．要件はｉ-3(a)満たしている．上記効果により権限下方硬直性が緩和でき，権限の増加や減少を，BP ロールを通して行うことができる．要

件ｉ-3(b)は満たしている．上記効果によりアクセス制御記述の局所化が行われる．要件ｉ-3(c)は満たしている．要件ｉ-3(c)に対する評価の詳細は[17]を参照されたい．要件ｉ-3(d)は満たしていない．BP ロールは BP の変化により変化する．これは比較的頻繁に変化するため安定していない．

表２は偽装機能による動作権限を決定する SC を決定する方式として，ユーザ ID ベース，プリンシパルロールベース，BP ロールベースを要件に対して評価したものである．（表２の要件ｉ-3(a)～ｉ-3(d)は章２を参照）

表２　要件に対する各方式の比較

	要件ｉ-3(a)	要件ｉ-3(b)	要件ｉ-3(c)	要件ｉ-3(d)
ユーザID	×	×	×	×
プリンシパルロール	×	×	×	○
BPロール	○	○	○	×

4. 最小特権を実現するプラットフォームの評価

ここでは２章で定義した，章３の方式を実装する環境について述べる．この実装環境は図１の中の WA であり，これが章２で定義した要件ｉとiiを満たすことにより，WS を使った企業情報システムにおいて最小特権を実現する．今回はこの実装環境として SELinux に注目した SELinux の特徴として，まず MAC をサポートしていることがある．MAC を用いることにより，システム全体に一貫したポリシーを強制することができる．また，SELinux は LSM[18]というカーネルレベルでの実装となっており，詳細なログを取得することが可能となっている．他に RBAC や TE[19]といったアクセス制御方式を提供しており，細かい権限の付与とアクセス制御を行うことができる．

章２で定義した要件ｉ-1～ii-3 に対する SELinux の達成度について評価していく．要件ii-1，ii-2 は上記の理由により MAC，LSM によって満たしている．また，要件ii-3 は SELinux 自体が宣言的なポリシー記述を行うことができるため満たしている．

次に SELinux が要件ｉ-1～ｉ-3 の機能を提供しているか評価していく．通常 SELinux は TE により要件ｉ-1 の機能だけを提供し，これによって SELinux 内部の最小特権を実現している．しかし，要件ｉ-2 と要件ｉ-3 の機能を提供しているかは明らかになっていない．そのため，SELinux の MAC 権限管理機能（ドメイン遷移[20]，setexeccon[21]，setcon[22]，setforkcon[23]）を評価対象として，機能ｉ-2 とｉ-3 を提供しているか評価した（表３：要件ｉ-1～ｉ-3 は章２を参照）．評価の詳細は[17]を参照されたい．

表３　SELinux の MAC 権限管理機能の評価

	要件ｉ-1	要件ｉ-2	要件ｉ-3
ドメイン遷移	○	×	○
setexeccon	○	×	×
setcon	○	△	×
setforkcon	○	△	×

評価の結果，要件ｉ-1 の機能は TE，ｉ-2 の機能はドメイン遷移，ｉ-3 は setcon/setforkcon が満たしている（偽装機能はプロセスレベルのみ）ことがわかった．このように，SELinux では要件ｉとiiの各要素を満たしており，章３の提案方式を実装する環境の要件を満たしていることがわかった．

5. SELinux による提案方式の実装

5.1. SELinux による提案方式の実装の手順

章３で提案したビジネス・プロセスを考慮し，業務に即した効率のよいアクセス制御記述の方法を，章４で評価した SELinux を用いて WS を使った企業情報システムにおいて最小特権を実現する方法について述べる．実装の仕方として次の手順（図３）の沿って進めていく．各項目は図１の各要素の設定である．

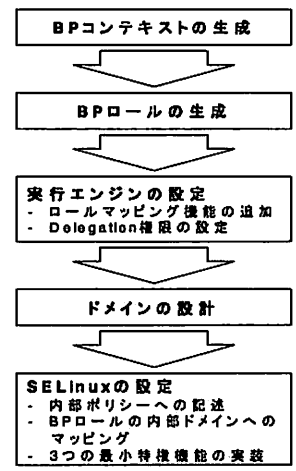

図３　SELinux による提案方式の実装手順

■BP コンテキストの生成

BP ロールへのロールマッピング機能のために BP コンテキストを作る．BP コンテキストは図１のポリシーディレクトリに入っており，BP の位置やオペレーションが書かれている．これにより，企業の BP に沿った BP コンテキストを作る．

■BP ロールの生成

偽装機能を使うときに用いる BP ロールを作る．これは図１の実行エンジン内部やポリシーディレクトリに置かれる．BP に考慮して BP ロールの配置を考える．BP の位置やオペレーションで必要な分だけの BP ロールを決めておく．

■実行エンジンの設定

現在の実行エンジンではこの提案方式を使うことはできない．そのために，２つの要素を加える．

- ロールマッピング機能

実行エンジンはプリンシパルロールと BP コンテキストを用いて，BP ロールへのマッピングを行う．そのため，どのプリンシパルロールとどの BP コンテキストの組合せがどの BP ロールへマッピングされるかをあらかじめ決めておき，それを実行エンジンが解釈できるように変更を加える．

- **Delegation 権限**
  通常の実行エンジンは SC への変更を行うことができないが，ロールマッピング機能では変更を行う．そのために，実行エンジンの設定により SC に変更を加えることができるような Delegation 権限を付与する．

■ドメインの設計

ドメインの設計では，WA 内部の権限の設計を行う．サーバの権限や保護データなどへのアクセスの許可などを決め，最小特権になるように権限を決める．ここで重要なのが，BP ロールに対しての権限の設計である．企業で定められたポリシーと BP とを考慮しながら，偽装機能で使う BP ロールの権限を必要最小限にするように決める．

■SELinux の設定

実装環境として使用する SELinux の設定を行う．提案方式の実装のため，次の 3 つの設定を行う．

- **内部ポリシーへの記述**
  『ドメインの設計』において設計した権限の構成や，あらかじめ決めておいた SELinux における動作などを，SELinux で実際に記述する内部ポリシーへ反映する．
- **BP ロールの内部ドメインへのマッピング**
  SELinux で偽装機能を使う場合，リクエストに貼り付けられた SC を内部ドメインにマッピングし，この内部ドメインで実際に MAC を使ったアクセス制御を行う．そのために，BP ロールに対応した内部ドメインを決めておく．
- **3 つの最小特権機能の実装**
  SELinux が要件 i-1～i-3 の機能を提供していることは述べた．実際に SELinux において，この 3 つの最小特権の機能を実装するために，SELinux の設定ファイルなどを設定する．

5.2. SELinux における3つの最小特権の実装

今回は図 3 のフローチャートで，『SELinux の設定』の中の 3 つの最小特権機能の実装を実際に行った．これは WA の設定であり，章 4 で評価した結果，SELinux において提供できることがわかっている．ここでは，評価結果のように実際に SELinux において要件 i-2 と i-3 の機能を提供できるのかについて，実装し確認する．また，今回 WA である SELinux では新しいロールである BP ロールを扱う．これに伴って BP ロールを扱うことができるように，SELinux の各権限管理機能を拡張する必要があるのかを確認する．実装環境は FedoraCore4[24]，カーネルは 2.6.14 である．

■サービスアカウント権限への制限

この機能は既に SELinux で提供されている TE を使うことにより，実装することができる．TE によるドメインの制限により，サーバの動作権限を必要最小限の権限に制限することができる．

■信頼コードを経由した保護データへのアクセス

今回この機能は次のように実装した．保護データへアクセスする権限の無いユーザが保護データへアクセスする場合，信頼コードを実行することにより保護データにアクセスすることができる特権に移る．これにより，信頼コードを経由した保護データへのアクセスを行う．

信頼コードとして『bpsh』というシェルを作成し，bpsh は保護データを実行できるように作成した．権限の無いユーザが bpsh を実行したときのみ特権を取得できるようにするため，ドメイン遷移によるエントリ・ポイントを使った特権への遷移を行う．これにより，ユーザは bpsh を実行したときにのみ特権ドメインに移り，保護データにアクセスできるようになる．ここで注意しなければならないことがある．SELinux のドメイン遷移の設定では，元ドメインから先ドメインへの遷移許可の記述と bpsh を実行したときに遷移を行うということを別々に宣言するため，前者の遷移の許可があれば setexeccon により bpsh の経由なしで遷移できてしまう．これを防ぐために，ユーザが持つドメインから setexeccon 権限を除去し，ドメイン遷移以外では特権ドメインに遷移できないようにする必要がある．

これらのドメイン遷移とエントリ・ポイントの指定，ユーザドメインからの setexeccon 権限の除去により信頼コードを経由した保護データへのアクセス機能を実装することができた．

■リクエストアカウント権限への偽装

今回はこの機能の実装を次のようにした．外部から BP ロールを SELinux の内部 SC にマッピングする信頼コードを作成し（今回は bpsh とした），これに特権を与える．これはこの信頼コードでのみ，内部 SC への偽装を行えるようにするためである．

この機能は既に SELinux で提供されている setcon 機能を使うことにより実装することができる．しかし，いくつかの設定をする必要がある．1 つ目はドメインへの dyntransition 権限の付与である．setcon でドメインの遷移を行うとき，この dyntransition 権限がないと遷移することができない．実装コードの例は次のとおりである．bpsh が持つ bpsh_t に権限を付与している．

```
－/etc/selinux/strict/src/policy/domains/user.te
allow bpsh_t bpsh_sale_t:process dyntransition;
```

2 つ目は constrains の緩和である．通常この constrains の制約によって，setcon を使ってドメイン遷移を行うことができるのは，同じユーザ ID で同じロールへの遷移のみであった．それを今回は，マッピングするための特権を持つドメインからならどんなドメインへも遷移できるように制約を緩和した．この特権を持つドメインは，BP ロールを SELinux の内部 SC にマッピングする機能を持つ信頼コードにのみ与える．

特権の定義と constrains の緩和の実装コードは次の通りである．

```
－/etc/selinux/strict/src/attrib.te
attribute privssetcon;

－/etc/selinux/strict/src/constrains
constrain process dyntransition
        (u1 == u2 and r1 == r2 ) or
        ( t1 == privsetcon and
          ( ( u1 == u2 or t1 == privuser ) or
            ( r1 == r2 or t1 == privrole ) or
            ( t1 == privuser and t1 == privrole ) ) );
```

これらの作業を行うことにより，3 つの最小特権の機能を実装することができた．実装結果については[17]を参照されたい．

次にこれらの実装により実証できたことについて述べる．この実装によって，章４で評価した通りに要件ｉ-2とｉ-3の機能がSELinuxにおいて提供できることを確認することができた．ただ，要件ｉ-3であるsetconによる偽装機能では，プロセスレベルの偽装までしか提供していない．そのため，リクエスト処理で必要となるスレッドレベルの偽装は提供できていないことがわかった．　また setcon の偽装機能により，新しいロールであるBPロールをSELinux内部のSCにマッピングするため，BPロールを扱うために各権限管理機能への機能拡張をする必要がないことを確認することができた．

6. まとめ

既存システムの再利用や効率化などから，企業情報システムのWS化はさらに進んでいくと考えられる．そこには日本版 SOX 法などのコンプライアンスにより最小特権が必要となる．

本論文では，このWSを用いた企業情報システムにおいて最小特権を実現させるために方法について述べ，その中の機能である偽装機能の権限決定をする主体にビジネス・プロセスを考慮した BP ロールを使う方式を提案した．また，実装環境としてSELinuxを評価し，SELinux を用いた最小特権の実現方法を提案した．それと共に，SELinux において３つの最小特権を実現する機能の実装例を示した．

今後の課題として，.NET や J2EE の Impersonation とのシームレスな連携がある．SELinux の setcon による偽装機能では，リクエスト毎の偽装に必要なスレッドレベルの偽装ができないため.NET や J2EE の Impersonationとの連携をすることを考慮する必要があると考える．また，BPフロー記述言語であるBPELが，BP コンテキストを扱えるように仕様拡張する必要があると考える．それに伴って，拡張した BPEL を扱えるように実行エンジンを実装する必要がある．そのため，実行エンジンの構築には拡張した BPEL を考慮した設計が必要であると考える．

文　献

[1] IT用語辞典 e-Words，“Webサービス”，
http://e-words.jp/w/WebE382B5E383BCE38393E382B9.html

[2] @IT，“セキュアOS「LIDS」入門　第１回”，
http://www.atmarkit.co.jp/fsecurity/rensai/lids01/lids02.html

[3] @IT，“インサイド .NET Framework　第１０回”，
http://www.atmarkit.co.jp/fdotnet/technology/idnfw11_10/idnfw11_10_01.html

[4] 物流 EDI センター，“解説「わかりやすい XML/EDI」　第８回”，
http://www.transport.jp/edi/xml/pdf/xml_04.pdf

[5] IPA，“強制アクセス制御に基づくWebサーバに関する調査・設計に関する調査報告書”，IPA，pp.67-68，IPA，2005.

[6] NSA，“Security-Enhanced Linux”，
http://www.nsa.gov/selinux/

[7] IT用語辞典 e-Words，“SSO”，
http://e-words.jp/w/SSO.html

[8] @IT，“Webサービスのセキュリティ　第４回”，
http://www.atmarkit.co.jp/fsecurity/rensai/webserv04/webserv01.html

[9] 独立行政法人 産業技術総合研究所，“グリッドコンピューティング標準化調査研究 成果報告書”，pp.111-111，財団法人 日本規格協会，2006.

[10] 趙京揚，“WS-BPEL 2.0 概要”，IBM，2005.

[11] Microsoft，“.NET”，
http://www.microsoft.com/net/

[12] Sun Microsystems，“Sun Developer Connection - Java EE & Web サービス”，
http://sdc.sun.co.jp/java/j2ee/index.html

[13] @IT，“セキュアOS「LIDS」入門　第３回”，
http://www.atmarkit.co.jp/fsecurity/rensai/lids03/lids01.html

[14] @IT，“インサイド .NET Framework　第９回”，
http://www.atmarkit.co.jp/fdotnet/technology/framework09/framework09_01.html

[15] Microsoft，“ASP.NET アプリケーションに偽装を実装する方法”，
http://support.microsoft.com/default.aspx?scid=kb;ja;306158

[16] NIST，“Role Based Access Control”，
http://csrc.nist.gov/rbac/

[17] 岡上智也，“SELinuxを用いたビジネス・プロセスベースの強制アクセス制御による権限管理方法”，Linux Conference 抄録集，第４巻，no.CP-07，2006.

[18] 中村雄一他著，“SELinux 徹底ガイド”，pp.18-20，日経 BP 社，2004.

[19] BILL MCCARTY，“SELinux システム管理”，田口裕也他（訳），pp.151-173，O'REILLY.

[20] BILL MCCARTY，“SELinux システム管理”，田口裕也他（訳），pp.31-33，O'REILLY.

[21] die.net，“setexeccon(3) - Linux man page”，
http://www.die.net/doc/linux/man/man3/setexeccon.3.html

[22] die.net，“setcon(3) - Linux man page”，
http://www.die.net/doc/linux/man/man3/setcon.3.html

[23] Fernando Vázquez，保理江高志，原田季栄，“The need for setuid style functionality in SELinux environments”，Linux Conference 抄録集，第２巻，no.CP-07，2004.

[24] The XOOPS Project，“Fedora.jp Project”，
http://fedora.jp/