# 上流工程におけるセキュリティ・メソドロジーの実践

大西 克美＊

# A Methodology of Information Security for Early Stages of System Development

Katsumi OHNISHI＊

我々が直面しているセキュリティ・インシデントの原因を考察すると, 上流工程におけるセキュリティ・アーキテクチャに課題がある場合が多い. 本論文では, 従来型のセキュリティ技術の選定方法の問題点を分析し, 堅牢なソリューションを確立するためにセキュリティ・メソドロジーに着眼する. また実際のお客様へ適用した事例を紹介することで, セキュリティ・メソドロジーの有用性を実証する.

As we investigate security incidents which we are facing today, we find that many of them have been caused by the absence or inadequacy of a security architecture built into the application systems in the upper process of their development. This paper analyzes the problems in the conventional way of selecting security techniques and focuses on a security methodology to establish secure solutions. The effectiveness of this approach is demonstrated by the cases where it has been applied to actual customers' application system development.

Key Words & Phrases：セキュリティ, メソドロジー, アーキテクチャ, e-Riskメソドロジー, 上流工程
security, methodology, architecture, e-Risk methodology, upper process

## 1．はじめに

近年IPAやJPCERT/CCに報告されるセキュリティ・インシデントやウィルス被害の届け出が恒常的に増加し, 相次ぐ不正アクセス, 情報漏えいの発生が, ビジネスのインフラ基盤を脅かしている. セキュリティ製品の充実, セキュリティ・コンサルティング・サービスなど, 新しいセキュリティ対策も定着してきたが, 完全な対応策を施していると言える企業は少ない.

一方セキュリティ技術者の不足も深刻である. 上流工程におけるセキュリティ対策の立案や, 下流工程におけるセキュリティ製品の導入, 構築, 運用局面における新種の脅威, 脆弱性に対する対策の施行など負担が重く, 慢性的に人材の不足が発生している. 特に上流工程における不充分なソリューション立案が原因となって引き起こされる脆弱性やインシデントは, 早急に対処すべき重要な課題である.

こうした背景に基づき, 統括的な視点に立ったセキュリティ・アーキテクチャ（セキュリティの基本設計, 体系）が必要である. 当然ながら, 卓越したコンサルタントやアーキテクトが参加することで, 質の高いセキュリティ・アーキテクチャを構築することは可能である［1］. しかし専門家に依存せず, アーキテクチャを創造するための方法論として, 本論文では, セキュリティ・メソドロジー（セキュリティ方法論）の実践に着眼した. 具体的事例に対して, セキュリティ・メソドロジーを適用することで, 筆者が経験した有用性を説明し, またセキュリティ・メソドロジー自身を向上させるための提言を述べる.

今回取り上げたメソドロジーは問題発見型のe-Riskメソドロジーであるが, IBMには他にも優れたメソドロジーがあり, 適用事例も紹介されているので一読することを推奨する［2］.

提出日：2004年6月1日
*ohnk@jp.ibm.com

## 2．セキュリティ対策の現状と課題

### 2.1　上流工程における理想と現実

セキュリティ・ソリューションを策定するときに, 上流工程で実施すべきことは脅威やシステム要件を整

理し, その対応策をアーキテクティング(アーキテクチャを作成する作業)することである. 理想的には日本情報処理開発協会が推進するISMS(情報セキュリティ・マネージメント・システム)を骨子としたセキュリティ・ポリシーを作成する[3]. 次に具体的な対策の立案となるが, 採用すべき製品, システムについては「ITセキュリティ評価及び認証制度」[4]を参考にし, 堅牢性を確保する. この認証制度は国際規格であるISO/IEC15408(通称コモン・クライテリア)に基づいており, 今後のセキュリティ評価の指針である.

しかし理想的な上流工程でのあり方に対し, 一般的な現実をかんがみると, コンサルティング・サービスを活用して, 積極的にセキュリティ・ポリシーを検討し, アーキテクトと共に最終ソリューションまで策定している事例は非常に少ない. 実質的には, セキュリティ担当者とメーカー技術者の間で施行錯誤の結果, 自家製のソリューションを作成することが多い. またインフラ・チームとアプリケーション・チーム間の意志の疎通や, 特定の脅威に対する見落とし, 過剰投資など, 全体としての整合性が欠如していることも少なくない. これを解決するために, ソリューション策定のための実践手順や標準化の整備が必要である. また自家製ソリューションへの過度な依存は, 結果として脆弱なシステムを引き起こし, 下流工程から後戻りすることが多い. セキュリティに関する規格や標準化が提唱される反面, 現実ではその実践が行えない問題に直面している.

### 2.2 セキュリティ技術の選定方法

上流工程で失敗しないためには, アーキテクティングを実践するときに, 体系化された手法を活用して, セキュリティ技術者がコンポーネントを選定するべきである. 筆者の場合「脅威とセキュリティ技術をマッピング」する方法を採用していた. 各脅威を明示的に技術とマッピングすることで, 漏れが発生することを抑止していた. またセキュリティ技術を「アプリケーション」「ミドルウェア」「ハードウェア」「ネットワーク」の4層に分類, 整理し, インフラ担当者とアプリケーション開発者の間でも, 脅威に対する抜けが発生しないように工夫していた. これに類似した手法が雑誌で紹介されている[5]. この手法を, 本論文では,「旧来型手法」と定義する. この手法には漏えい, 不正侵入といった脅威に対して, 具現化されたセキュリティ・コンポーネントを適用させる方法を採用することで, 実装を容易にする効果がある.

### 2.3 旧来型手法の限界と課題点

この手法が実用的であると考える反面, セキュリティ要件が多様化し, セキュリティ・インシデントが急増してくると, この旧来型手法の限界を感じ, 次の4点が現状の課題であるという結論に至った.

① 結果(既存製品)から要件を誘導している
② 実施者の経験, 技術力への依存度が高い
③ 再利用できるような標準化が定着していない
④ 生成されたデザインに対する満足度が低い

まず問題の根源は業界におけるセキュリティ担当者の絶対数が少ないことであるが, これが結果として①のように, セキュリティ製品の機能から, 対処すべきセキュリティ要件を導くといった「結論ありきの要件定義」を誘導している. つまり時間の効率化を追求するために, 現状利用可能な製品の機能から, 対応できるセキュリティ要件を絞り込む(制限する)手法である. 当然ながらこの手法では,「抜けのないセキュリティ・システム」の確立が困難となる.

また②で指摘しているように, セキュリティ・デザインの品質は, 実施者の経験, 技術力に左右されることが大きい. 結果として,「あの技術者がデザインしたから安心である」といった根拠のない安堵感を生み出している. しかしこの手法に対する具体的な対案がないため, この手法から脱却できないのが実状である.

単に人材不足だけが問題ではなく, 標準化を意識した手法を実践していないことも課題である. 筆者自身の場合もセキュリティ・デザインを実践するときに標準化を意識したことはあまりなく, むしろ今回指摘した①②のケースに属することが多かった. おそらく他のセキュリティ技術者も同様で, 標準化の重要性を認識しながらも, 現実的には実践できていないと思われる. その結果③で指摘した標準化を上流工程で定着させることが困難となっている.

④に関しては, 実際お客様と会話して気付いた点であるが, 上記①②が制約となり, お客様は成果物であるセキュリティ・デザインに対し妥協されることも少なくない. 提案されたセキュリティ・デザインに対する満足感が低く, 投資の効果と優先順位も不明瞭なため, 最終的なソリューションの適正さに対しても疑問を感じていることが判明した.

## 3. セキュリティ・アーキテクチャの方法論

サーバーやネットワークに対する堅牢性, 安定性の追求, セキュリティ要件の多様化に伴い, 旧来型手法による単一的なデザインではなく, 総括的な視点に立ったセキュリティ・アーキテクチャが必要である. それを具現化した一つのモデルに, IBMのセキュリティ・アーキテクチャ[6]がある.

セキュリティ・アーキテクチャを作成する方法論として, IBMには, 複数のセキュリティ・メソドロジーがあるが, 構成や実践手順などの考え方に共通点が多

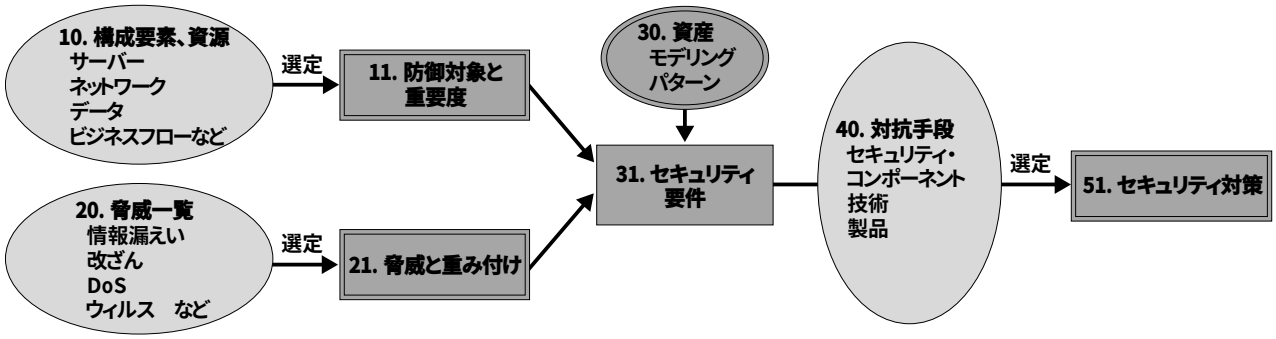


**図1. e-Riskメソドロジーの実践手順**

い. 以下に, 筆者が参考にしたe-Riskメソドロジーの構成, 実践手順を紹介する. セキュリティ・アーキテクチャを作成するための方法論として, 2.2で紹介した旧来型手法とはかなり違うことが分かる.

### 3.1 e-Risk メソドロジーの構成

e-Riskメソドロジーは次のような特徴を有する.

①既存システムのヘルス・チェックや簡易的なセキュリティ技術の選定に適する

②多くのコンポーネントがテンプレートの形で提供されるため, 初心者でも実践が容易である

またe-Riskは次の五つの機能から構成される.

(1) ビジネスのパターン化

e-Riskメソドロジーではシステムの形態を七つのパターンに分類する. パターンは構成されるサーバー, ネットワーク, アプリケーションの特性で差別化され, 対象となるシステム形態がどのパターンに類似しているかを選定し, そのパターンを構成するリソースの確認を行う.

(2) 脅威の選定

代表的な脅威がテンプレートで提供されており, セッションを通じて対象となる脅威を絞り込む. 脅威を選択する方式を採用することで, 脅威に対する漏れをなくすことを実現する.

(3) セキュリティ要件の選定

対処すべきセキュリティ要件を六つに分類し, その要件を実現するための技術, 対抗手段をテンプレートから選定する. e-Riskメソドロジーで提供される六つのセキュリティ要件とは「機密性」「完全性」「可用性」「否認不能性」「識別＆認証」「アクセス制御」であり, ISO7498-2の要件とも相通ずる. 一般的な手法では防護すべき資源や脅威の選定から, セキュリティ要件を導き出すことが多いが, e-Riskメソドロジーでは, あらかじめ想定できる要件を事前に定義し, 選択する形式を採用している. 時間的な制約があるケースや, 技術不足でセキュリティ要件をまとめきれないケースでは, 有用な方法である.

(4) セキュリティ技術の選定

6つのセキュリティ要件を実現するために採用すべき技術について, 効果の点から重み付けを実施し, 重要度を決定する. 各セキュリティ要件に対するセキュリティ技術の効果を整理し, 最終的なソリューション選定への草案をまとめる. ただしテンプレートとして提供されている技術, 対抗手段を理解し, 効果的な選定を行うには, 実施者にセキュリティのスキル, 経験が必要とされる課題点も併存している.

(5) ビジネス・パターンへの組み込み

(1)で選定したビジネス・パターンに対し, (4)で選定したセキュリティ技術を組み込む. またこの段階でお客様の投資や意向に基づき, 個々のセキュリティ技術に対して, 具体的な製品の選定を行い, 適応すべきリソースに組み込み, ソリューションを完成させる.

### 3.2 e-Riskメソドロジーの実践手順

3.1でe-Riskの構成を紹介したが, その各機能を実践する手順を**図1**に示す. 特に筆者が創意工夫し, 新規に追加した手順を二重枠線で記述する.

3.1(1)のパターンに属する「10. 構成要素, 資源」はテンプレートで提供されるが,「11. 防御対象と重要度」の手順を組み込むことで, パターンを構成するリソースに対して, どこを重点的に守るべきかを整理することができる.

3.1(2)の機能は「20.脅威一覧」で実践される. 続いて「21. 脅威と重み付け」で重み付けを行い, 防御対象や脅威に重要度を設定する.

「10. 構成要素, 資源」と「20. 脅威一覧」で同様の手順を取ることで, 実施者が容易に実践できるだけでなく,「11. 防御対象と重要度」,「21. 脅威と重み付け」で重み付けを変えることで, 案件ごとに差別化が行え, 結果として異なったソリューションに発展させることが可能となる効果がある.

3.1(3)で定義された6つのセキュリティ要件に対し,「11. 防御対象と重要度」と「21. 脅威と重み付け」の結果を関連付けることで, ストーリー性を考慮した. また「30. 資産」で提供されるモデリングやパ

ターン事例を同時に参照することで, より洗練された要件定義が可能となる. この手順が「31. セキュリティ要件」である.

3.1(4)は「40. 対抗手段」で実践される.「31. セキュリティ要件」から最終成果物である「51. セキュリティ対策」に展開する上で, このフィルタリングを通す. ここでもセキュリティ対策を実現するためのコンポーネント, 技術に関するテンプレート, 評価が用意されており, 最終選定の補助を担っている.

最終的な「51. セキュリティ対策」は3.1(5)でパターンに組み込んで整理される. この段階で, 最終的な製品名との対応付けを行う手順を新たに補完することで, より現実的なソリューションの策定が可能となる.

## 4. e-Riskメソドロジーの適用事例

この章では筆者がe-Riskメソドロジーを活用し, セキュリティ・アーキテクチャを行った事例を用いて, セキュリティ・メソドロジーの有用性を考察する.

### 4.1 対象事例

今回の事例は, 2.3で指摘した課題に類似している. 最終の製品が既に想定されており, 脅威の評価, セキュリティ要件の確定, セキュリティ技術の選定に関しても, 特段の手法は必要と考えられなかった. 具体的な製品の機能から, 脅威, セキュリティ要件を導き出した「結論ありきの要件定義」に近い事例である.

今回対象となるシステムの形態はメール・サーバー, Webサーバーによる情報交換, 情報発信システムである. そしてセキュリティ要件として, 次の2点のみが挙げられていた.

① 添付ファイルに対するウィルス対策が講じられること
② メール送信による情報漏えいを抑止すること

この要件に対処すべきセキュリティ・ソリューションとして, お客様の判断は実績のあるソフトウェアの採用, つまり1 中継メール・サーバーへのウィルス駆除ソフトウェアの導入, 2 コンテンツ・フィルタリング・ソフトウェアの機能による漏えい対策の実現であった.

### 4.2 e-Riskメソドロジーの実践

実際のセッションにおいて, 3.1で記述したe-Riskメソドロジーの五つの構成に対し, 筆者自身が追加した機能と, 3.2で説明した実践手順に準じて説明する.

#### ステップ1:脅威の選定

3.1(2)で説明した脅威を選定する機能に対し, 3.2の「21.脅威と重み付け」の手順を追加した. その効果として, ステップ2で実施したセキュリティ要件の重要度の整理に役立った. 今回の実施結果を**表1**に記す.

表1. 脅威の選定と重要度

| 定義する脅威 | 重要度 |
|---|---|
| 情報の盗難・漏えい | ◎ |
| 悪意を持ったコード | |
| DoS(サービス妨害) | ○ |
| 否認 | ○ |
| アクセス違反 | ○ |
| プログラム・エラー | |
| ソーシャル・エンジニアリング | ○ |
| ウィルス | ◎ |

表2. セキュリティ要件と脅威の相関関係

| セキュリティ要件 | 重要度 | 脅威との対応 |
|---|---|---|
| 機密性 | ◎ | 情報の盗難・漏えい, ソーシャル・エンジニアリング |
| 完全性 | | |
| 可用性 | ◎ | DoS, ウィルス |
| 否認不能性 | ○ | 否認, ソーシャル・エンジニアリング |
| 識別&認証 | ○ | 情報の盗難・漏えい, アクセス違反 |
| アクセス制御 | | |

まず脅威の説明を事前に行い, セッション参加者と共に, その重要度を考察していった. 既にこの時点で, 当初想定していた脅威である「情報の漏えい」,「ウィルス」以外に見落としていた脅威が内在していたことが判明した. 重要度は低いがシステムの安定稼動, 問題発生時の対応の効率化を考慮すると, 検討に値する脅威であると考え, 追加定義した. 今回の事例では脅威一覧が, e-Riskメソドロジーのテンプレートで提供されていたことが奏功した結果となった.

#### ステップ2:セキュリティ要件と脅威の対応

このステップは3.1で紹介したe-Riskメソドロジーの標準的な機能としては提供されない.

手順としては, 3.2の「31.セキュリティ要件」で実践されるが, 脅威とセキュリティ要件の関連性が希薄であると判断し, 対応付けを行うステップを意図的に設けることでストーリー性を補足したステップとして改良した. 今回の実施結果を**表2**に記す.

まずe-Riskメソドロジーで定義されている六つのセキュリティ要件に対し, 3.2の「11. 防御対象と重要度」,「21. 脅威と重み付け」の結果を考慮し, 脅威とセキュリティ要件との関連, セキュリティ要件の重要度を整理した. 例えば「機密性」の重要度を考えるときに,「セキュリティ要件を誘導した脅威の種類」,「システムに影響を及ぼす脅威の重要度」を考慮することで, その重要度と相関関係をまとめる. このステップの効果は, 感覚的に理解していたセキュリティ要件と直面する脅威の相関関係が明確になり, 要件を正しく理解, 整理することが可能となる.

このステップの結果として, 当初から「情報の盗聴・漏えい」と「ウィルス」を最大の脅威と考えていたので, セッションの結果としても,「機密性」「可用性」の重要度が高くなった.

### ステップ3：セキュリティ技術の選定

3.1（4）で説明したセキュリティ技術の選定において, 最新のセキュリティ技術を標準のテンプレートに追加したことにより, 最終成果に先進的なソリューション, 製品を包含することが可能となった.

実際のセッションでは, ステップ2で選定した四つのセキュリティ要件について全て討議を行ったが, 本論文では実施結果に基づき,「可用性」に対する結果のみを**表3**として記す.

効果に関しては,「非常に効果的：++」,「有効：+」,「最低限必要：x」の三段階で評価した.

表3. セキュリティ技術

| 「可用性」を実現する技術 | 効果 |
|---|---|
| スタンバイ構成 | ++ |
| クラスタリング | ++ |
| ハードニング | X |
| DoS対策 | X |
| バックアップ／リカバリー | X |
| キャパシティ管理 | |
| ウィルス対策 | ++ |
| アセット管理 | |

表4. ソリューションの選定

| 効果 | セキュリティ技術 | ソリューション |
|---|---|---|
| 非常に効果的 | コンテンツ・フィルタリング<br>スタンバイ構成<br>クラスタリング<br>ウィルス対策 | コンテンツ・フィルタリング製品<br>サーバー多重化<br>負荷分散装置<br>ウィルス駆除製品 |
| 有効 | ロギング | サーバーでの設定 |
| 最低限必要 | ハードニング<br>DoS対策<br>バックアップ・リカバリー<br>パスワード認証 | OSハードニング<br>OSハードニング<br>バックアップソリューション<br>認証サーバー |

### ステップ4：セキュリティ対策の立案

このステップはe-Riskメソドロジーでは標準的に提供されない. 3.2で説明した実践手順では「40.対抗手段」までが提供されるにとどまる. しかし抽象的なセキュリティ技術を, 具現化された製品に対応付ける手順を追加することにより, ソリューション選定のプロセス, 採用理由が明確となることを実現した.

今回のセッションで検討してきたセキュリティ要件の重要度, セキュリティ技術の効果からソリューションを選定したものが**表4**である.

## 4.3 セキュリティ・メソドロジー活用による評価と効果

今回e-Riskメソドロジーを活用して, セキュリティ・ソリューションの策定を行ったが, お客様はセキュリティ・メソドロジーに先進性と有用性を認め, 次のように評価された.

① 内在していた脅威を網羅できる
② 優先順位, 既存技術を提示されることで, 次回の投資計画が明確になる
③ 成果物への同意が容易に行える
④ 選定のプロセス手法, 成果のまとめ方が参考になる

今回の事例では, 当初お客様から提示されたセキュリティ・ソリューションは, ソフトウェア製品による「ウィルス駆除」と「コンテンツ・フィルタリング」のみであり, 表4と比較すると, 不充分な結論であったことが分かる. セキュリティ・メソドロジーを実践しなかった場合, 内在化した脅威を発掘できず, 抜けのあるソリューションになっていた危険性があったことが分かる.

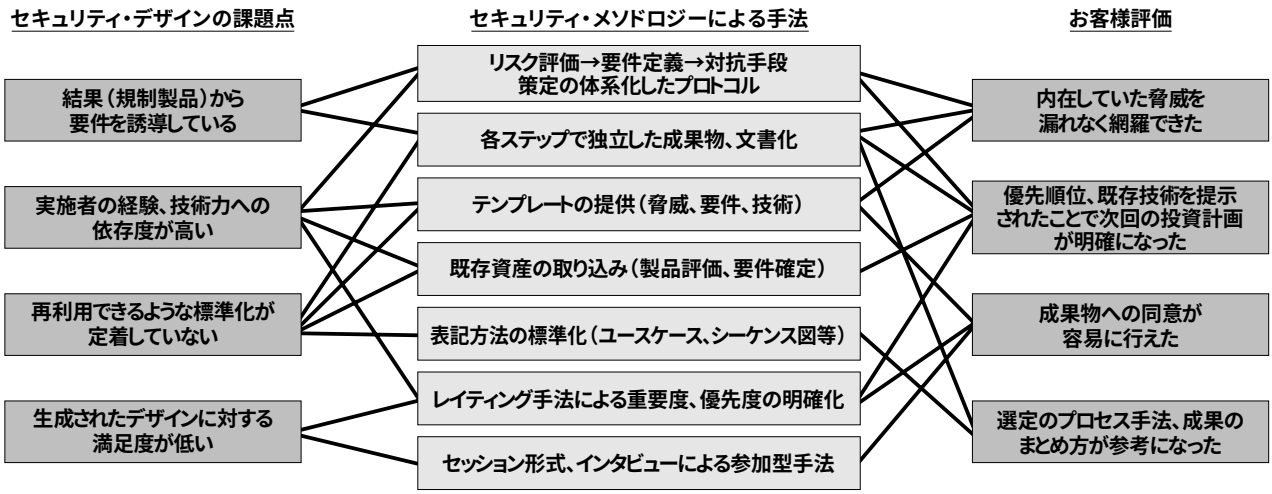


図2. メソドロジーによる課題解決と評価の実現

## 5. 考察：上流工程おけるセキュリティ・メソドロジー実践の価値

この事例におけるお客様の評価は，他のセキュリティ案件においても，共通部分が多いと考える．セキュリティ・メソドロジーに対して，お客様の評価は肯定的であり，我々がその価値を再考するきっかけになると思われる．

次にセキュリティ・アーキテクチャを確立するための方法論として，セキュリティ・メソドロジーの有用性を考察する．この考察を実証するために，**図2**における課題点との対比，評価を誘導したメソドロジーの手法との整理が一助になると考える．

アーキテクチャが必要とされる目的の一つに網羅性，完全性がある．プロセスを明確化し，局面化した成果物を設定することで，漏れのないソリューションの確立が可能となる．メソドロジーは体系的な実践手順を採用し，ステップ毎に成果物を独立させることで，これを実現している．

またアーキテクティングを推進するためには，再利用できるような標準化が必要である．メソドロジーは表記法や文書化において標準化を提供している．またテンプレートを併用することで，経験の少ない実施者への援助や，特定技術者からの自立を推進することが可能となる．

他事例における評価結果やパターン化されたモデルを盛り込むためのインターフェースが整備されていることも重要な要件である．参照結果を包含することで，より汎(はん)用的なアーキテクチャの創造が可能となる．

このように上流工程でセキュリティ・メソドロジーを活用することは，現状のセキュリティ・デザインの課題点を解決し，かつ包括的なセキュリティ・アーキテクチャを展開するための強力な礎となる．

## 6. おわりに

現在直面しているセキュリティの課題を解決するために，上流工程におけるセキュリティ・メソドロジーの実践が有用であるという結論に至った．筆者は，従来のセキュリティ・メソドロジーに，次のアイデアを導入し利便性を高め，具体的事例において，その有効性を確認した．

① 各手順間のストーリ性の補完
② 脅威，セキュリティ技術における新規コンポーネントの追加
③ セキュリティ技術とソリューションの対比

またセキュリティ・メソドロジーの更なる整備のために，以下の2点の改善が望まれる．

(1) 再利用を推進するインフラ整備

他事例を参考にする仕組みを整備することが，実施者の負担を軽減し，選択したコンポーネントに対する妥当性を高めることに役立つ．

また実践した結果を，他者に還元，再利用できるインフラ整備も必要である．

(2) 積極的なメソドロジーの実践

実施者の手法，アイデアを付記することで，メソドロジーが一層洗練，成熟されたものとなる．実践を繰り返すことで，より利便性の高いメソドロジーへの改善が期待できる．

参考文献

［1］山崎哲，「ネットワーク社会におけるセキュリティ・アーキテクチャの活用」，PROVISION No.35，2002年10月
［2］渡辺芳明，「セキュリティを確保したシステムを構築するための方法論」，PROVISION No.29，2001年4月
［3］日本情報処理開発協会（JIPDEC），http://www.isms.jipdec.jp/isms/index.html，2004年6月1日
［4］情報処理推進機構（IPA），http://www.ipa.go.jp/security/jisec/，2004年6月1日
［5］古川浩，菅野孝治，「Webシステムのセキュリティと運用管理」，日経インターネット・テクノロジー，2002年8月
［6］J.J. Whitmore，「*A method for designing secure solutions*」，*IBM System Journal* vol.40，2001年11月

**日本アイ・ビー・エム株式会社**
**コンサルティングITアーキテクト**
**大西 克美** Katsumi OHNISHI

**［プロフィール］**
**1986年、日本アイ・ビー・エム入社．公共事業部のシステム・エンジニアとして，お客様に対する提案活動，システム構築や運用支援を担当．UNIXサーバーやインターネット基盤の設計，導入を経て，90年代後半よりセキュリティ案件に参画．2002年よりセキュリティ・アーキテクトとして活動中．**