安全なシステム開発における協調型セキュア構築プロセスの提案

綿口 吉郎† 大久保 隆夫† 海野 雪絵† 金谷 延幸†

† 株式会社 富士通研究所
211-8588 神奈川県川崎市中原区上小田中 4-1-1
{wataguchi, okubo, unno.yukie, kanaya.nobuyuki }@jp.fujitsu.com

あらまし 近年，企業サイトを対象とした攻撃の増加に伴い，利用者の個人情報が危険に曝されており，システム構築会社では安全なシステム構築のしくみづくりを強化している．一方で，従来から開発現場では多忙であり，時間やリソース不足を理由に新しいセキュリティ活動がなかなか浸透しない．本稿では，安全なシステム構築のために当社が独自に開発した協調型セキュア構築プロセスおよび基盤システムへの適用について述べる．

Cooperative Secure Integration Process for Secure System Development

Yoshiro Wataguchi† Takao Okubo† Yukie Unno† Nobuyuki Kanaya†

† FUJITSU LABORATORIES LTD.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki city, Kanagawa 211-8588 JAPAN

Abstract Recently, attacks on corporate web sites are increasing, and personal information on sites faces growing risks. Most developers are making an effort to establish secure system development process. However, they have trouble to practice, because they are busy and short of time or human resources. In this paper, we describe Cooperative Secure Integration Process that we originally developed for secure system development, and introduce the adapted management system.

1. はじめに

近年，企業サイトを対象としたインターネット上の攻撃が増加しており，大手企業や政府のWebアプリケーションは2分に1度の頻度で攻撃を受けている[1][2]．また，利用者の個人情報が漏洩する事件も多発しており，従来に増してWebサイトでの厳重な情報管理を求める声が高まっている．一方，Webアプリケーションのシステム構築会社では従来から安全なコンピュータシステム構築のための体制づくりが進めているが，近年の攻撃増加傾向から，より一層安全な構築技術の確立が不可欠である．

本稿では，システム構築会社での安全なシステム構築の実践と，実施のための開発技術について述べる．2章では従来のセキュアソフト開発手法について述べ，3章では富士通で開発した

手法を述べる．4章および5章では手法の課題と解消のための開発技術について述べ，6章では技術適用の効果について述べる．

2. 安全なシステム構築

2.1. 既存のセキュアソフト開発手法

従来から安全なソフトウェア開発の対策方法や手法について提案されている．例えば，IPAの「安全なウェブサイトの作り方」[3]では安全な実装や攻撃への対策方式が提案されている．

また，開発プロセスについては Microsoft が Security Development Lifecycle[4] (SDL)を提案している．要求段階でセキュリティ要件定義を行い，設計段階で脅威モデリングを行い，実装段階で静的コード解析を行い，検証段階でペネトレーションテストを行うことが示されている．

2.2. 従来手法の課題

従来の多くの手法はシステムインテグレーションを主対象とはしておらず，事業特性が適合しないため適用が困難である．

例えば，ソフト製品はライフサイクルが長いため，セキュリティ投資を長期的に回収できる．一方，顧客から受託してソフトを製造するシステムインテグレーションでは，一般に潤沢なセキュリティ予算が与えられない．セキュリティ専門家を大量に投入する SDL は実施困難である．

また，ソフト製品は自社の判断でセキュリティ仕様を決定できるが，システムインテグレーションでは独断では決定できず，顧客へ提案し，承諾を得る必要がある．顧客調整能力とセキュリティ能力を同時に高いレベルで要求することは現実的ではない．

3. 協調型セキュア構築プロセス

事業特性から富士通では独自の協調型セキュア構築プロセス(CSIP: Cooperative Secure Integration Process)による安全なシステム開発手法を開発した．これは図 1のように多数の

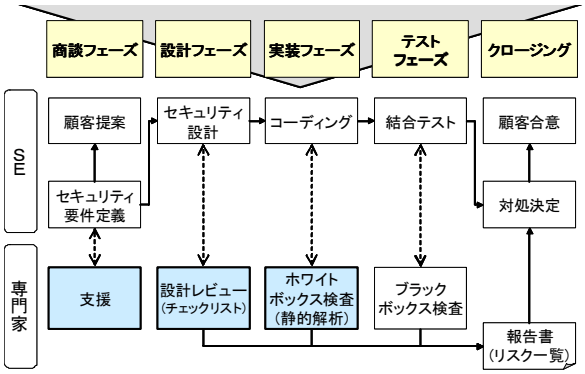

図 1 協調型セキュア構築プロセス(CSIP)概要

システムエンジニアを，少数のセキュリティ専門家がサポートする構築プロセスである．なお，主要なターゲットは Web アプリケーションシステムであるため，これに適合した検査ツールや手法を採用している．

CSIP で中心的な役割を果すのはシステムエンジニア(SE)である．例えば，システム性能や利便性，費用など様々な制約問題の解決を担う．セキュリティはそれらの一要素に過ぎない．一般に SE が網羅的なセキュリティ知識を完備していることは期待できない．

一方で，セキュリティ面で中心的な役割を果すのはセキュリティ専門家である．専門家はセキュリティ知識に精通しているが，各システム固有の知識は備えていない．

専門家はセキュリティ観点から検討領域を切り出して適切な質問を投げかけ， SE はシステム固有のセキュリティの過不足を検討する．また，専門家はセキュリティ問題の発見や指摘するだけに留まらず，実行可能な対策や実装方式を共に考えて SE をサポートする．このように，CSIP では両者が協調しながらシステムの安全を確保するのが特徴である．

以降では，CSIP の各フェーズで実施する検査作業について述べる．

3.1. 商談フェーズ：セキュリティ要件定義

商談フェーズは顧客にシステムを提案して受注するまでの段階である．一般にシステム提案で開発費用やアーキテクチャ，機器構成の大枠が決定される．一方，事前に顧客からセキュリテ

ィ要件が明示されないことも多く，受注後に開発機能が増えるなどプロジェクト採算に関わる問題が生じやすい．CSIP ではセキュリティに関連する開発項目が欠落しないように，専門家がセキュリティ要求パターン[5]などの手法を用いて要件分析を支援する．

3.2. 設計フェーズ：設計レビュー

設計フェーズはシステムの様々な仕様や機能設計，プログラム設計を行う段階である．この段階の設計レビューでは，適切なセキュリティ方式や脆弱性対策が盛り込まれているか確認する．確認には2つの手法を併用する．

一つは，最低限の実施不可欠な遵守事項や対策を列挙したチェックリスト方式である．SE が回答したチェックリストを専門家が確認することで，短時間で一定のセキュリティ水準が確保されていることを確認できる．

もう一つは，専門家がSEに直接対面してインタビューする方式である．システムの前提条件や環境を確認し，これに基づいた脅威分析や対策を精査することで，システム固有のセキュリティ問題点の見落としを防ぐ．同時にSEがセキュリティ問題を適切に理解しているか確認する．これは後工程で不適切な対策を避けるために重要である．時間や労力などの負担が大きいが，システム開発の安全確保の根幹になるため，CSIP ではインタビュー方式を特に重視している．

3.3. 実装フェーズ：ホワイトボックス検査

実装フェーズはプログラミングと単体テストを行う段階である．特にセキュリティ要求が高いプロジェクトでは，この段階でホワイトボックス検査を行う．静的コード解析技術によって SQL インジェクションなどの脆弱性を確認する．

次のブラックボックス検査は幅広い検査が可能だが，システム構築の終盤まで実施できない．ホワイトボックス検査はプログラム段階で実施でき，手戻りを軽減できる利点がある[6]．

3.4. テストフェーズ：ブラックボックス検査

テストフェーズはシステムを実際に動作させて行う総合テストの段階である．この段階のブラックボックス検査では検査ツール[7]を用いて脆弱性が無いか確認する．

この他，ネットワークや OS などを対象としたインフラ検査[8]，利用者マニュアルなどのセキュリティ説明の不備を確認する表記チェックを行う．

3.5. セキュリティ問題の一元管理

我々は，CSIP を支援する基盤ツール SIAT (Security Inspection Assistance Tool)を開発した．CSIP ではシステム納入までの検査計画や検査結果を SIAT 上で一元管理することで，セキュリティ活動を円滑化している．主な機能を次に列挙する．

1. プロジェクト情報と開発時期の一覧管理
2. 各検査作業の実施予定の調整
3. チェックリストと補足説明の入手
4. セキュリティチェックリストの回答
5. セキュリティ問題点の一元管理
6. 検査報告書の作成
7. セキュリティ問題点の対策状況の管理
8. プロジェクトと検査結果の一覧表示

4. 課題

協調型セキュア構築プロセス(CSIP)による安全なシステム開発を行うには次の課題がある．

4.1. セキュリティ問題の理解力のギャップ

CSIP では専門家が対策を直接決定するのではなく，SE が顧客に問題を説明して最適なセキュリティ対策を実施する．SE がセキュリティ問題を正しく理解できない場合，問題を過小評価したり，対策を軽視したりすることに繋がる．

SE がシステムへ固有の影響や具体的な攻撃を想像できない場合には，リスク心理学の利用可能性ヒューリスティック[9]の効果によって特に顕著になる．問題を具体的にイメージできるほど発生確率は高いと認知され，逆にイメージの具

体性を欠くほど発生確率は低いと認知される傾向である. また, 顧客に説明する際にも同様の問題が生じる.

利用可能性ヒューリスティックの影響によってSE や顧客が問題を過小評価することを避けるために, 専門家がシステムに応じた具体的な攻撃シナリオで解説することが課題になる.

4.2. セキュリティ達成レベルの不確かさ

プロジェクト管理の観点からは, プロジェクトのQCD(品質, 採算, 納期)の管理が重要である. 品質では障害件数や, 要件達成度, システム性能などの指標に加えてセキュリティ達成レベルが含まれる. システム構築事業の経営管理では, 懸念の大きなプロジェクトを把握し, 適切な施策やリソース投入を判断できるようにプロジェクト間で相互に指標を比較することが要求される.

一方で, 設計レビューの結果からセキュリティ達成レベルをプロジェクト間で比較することは難しい. 各プロジェクトで脅威のウェイトが異なるためである. 例えば, オンラインバンキングではサイト偽装の脅威は重大であるため, 設計レビューで対策不足が確認されればセキュリティ達成レベルは低いと言える. 一方, 一般情報を公開する企業サイトではサイト偽装は大きな脅威ではなく, セキュリティ達成レベルが低いとは言えない.

このようにシステム固有の前提条件を無視して単純なチェックリストの達成率だけで評価する方法はプロジェクト間の比較には適切でない. 要求が異なるプロジェクト間で比較可能なセキュリティ達成レベルの指標化が課題になる.

5. 協調性を高める手法の開発

我々は前述の課題を解決する開発技術をSIAT に実装した. 次に SE と専門家の協調性を高める2つの手法について述べる.

5.1. 類似リスクの抽出

セキュリティ問題の理解力のギャップを解消するために, セキュリティ問題や攻撃手法の解説文では専門用語を避け, 誤解を与えない比喩を使うなど, 周到な解説文にする工夫が必要である.

一方, 攻撃の可能性を正しく理解させるには, システムに応じた具体的な攻撃シナリオで解説する必要がある. これにはシステムの固有名詞を含めるなど, 毎回カスタマイズした解説文にする工夫が必要である.

前者はテンプレート化で効率化できるが, 後者は難しく, また毎回攻撃シナリオを創出するのは非効率である. そこで, 類似する過去のセキュリティ問題を抽出し, 解説文を流用することで攻撃シナリオのカスタマイズを容易にしてこの問題を解決する.

SIAT では設計レビューで発見されたセキュリティ問題点と, チェックリスト項目違反の対応関係を入力する(関係ベクトル **A**). 同様に過去の問題点データも関係ベクトル(**B**)を持っている.

$$\mathbf{A}=(q_{11},q_{12},\cdots,q_{1n})$$
$$\mathbf{B}=(q_{21},q_{22},\cdots,q_{2n})$$
$$q_{ij}=\begin{cases}1 & \text{...違反あり}\\ 0 & \text{...違反無し}\end{cases}$$

A と **B** に共通する違反項目数 C_iと, **A** か **B** いずれかの違反項目数 C_uは次式で得られる.

$$C_i=\mathbf{A}\cdot\mathbf{B}=\sum_{j=1}^{n}q_{1j}q_{2j}$$
$$C_u=\left(\sum_{i=1,2}\sum_{j=1}^{n}q_{ij}\right)-C_i$$

また類似度 Sは次式で得られる.

$$r=C_i/C_u\ [0\le r\le 1]\quad (\text{ただし}C_u=0\ \text{ならば}r=0)$$
$$S=C_i\cdot r=C_i^{\,2}/C_u$$

なお, 類似度 S は共通数 C_iだけでなく合致比率rを考慮している. 例えば, **A**={1,1,1,0}と **B**={0,1,1,1}の評価と, **A'**={0,1,1,0}と **B'**={0,1,1,0}の評価では C_iはともに 2 で等しいが, 完全一致している後者の類似度はより高く評価される.

この類似度の評価式を用いて, チェックリスト違反項目の入力から適切な過去の類似するセ

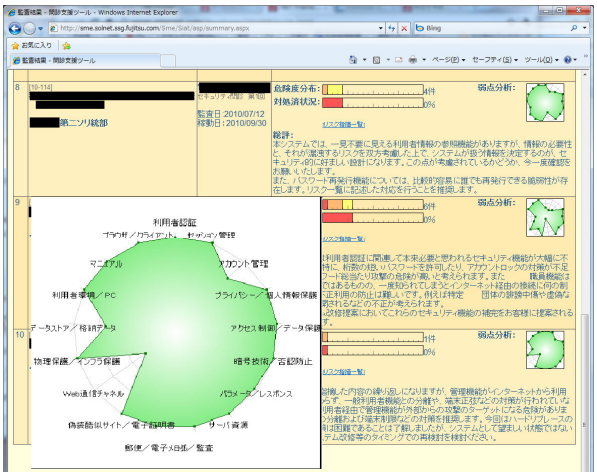

図 2 セキュリティ達成レベルの視覚化

キュリティ問題を抽出でき，これを流用して攻撃シナリオをカスタマイズした解説が容易になる．

5.2. セキュリティ達成レベルの多軸指標

セキュリティ達成レベルの不確かさを解消するために，SIAT では各々のシステムの達成レベルを多軸の脅威観点ごとに相対評価する方式にした．例えば，脅威観点には個人情報，利用者認証，通信路，データストア，サーバ資源，偽装酷似サイト，利用者環境などに対する脅威が含まれる．

また，相対評価の結果はベクトルとして扱い，単一の指標に集約しない．ただし，このままでは SE や経営管理者にとって理解が困難であるため，図 2のようにプロジェクト一覧とともにセキュリティ達成レベルをレーダチャート方式で表示することで直感的に比較可能にしている．

6. 適用結果

6.1. 理解力ギャップの解消と対策の向上

類似リスクの抽出によって，SE に適切にセキュリティ問題を理解させることが可能になった．SIAT では検査結果に対する SE の対策案や，顧客合意の状況を入力している．SIAT への技術適用後は，SE の不適切な対策案の回答が減少し，同時に顧客が対策案に合意することが増加した．

年度	実施件数
2007	23
2008	34
2009	34
2010	35

図 3 年間実施件数の趨勢

さらに，専門家によるセキュリティ報告書の作成時間は従来と比較して半減しており，専門家の稼働率が向上した．少数の専門家が多数の SE をサポートする CSIP では，専門家不足が全体の実施件数のボトルネックとなっていたが，SIAT によりボトルネックが解消され，年間実施件数が拡大した(図 3)．

6.2. 達成レベルの相互比較と経営管理

多軸の指標化によってセキュリティ達成レベルをプロジェクト間で比較可能になった．また，多軸の評価指標は，システム構築事業の経営管理者や SE が事業リスクを把握できる効果が得られた．全体に低スコアだが均整のとれたチャートと，脅威観点ごとに極度に偏ったチャートを比較した場合，後者はセキュリティを十分に考慮していないことが推察できる．このように相互比較できることは，経営管理の上で特に有益である．

例えば，SE のセキュリティ意識が乏しくセキュリティの懸念が大きな開発プロジェクトがあったが，バランスの悪い歯抜けのレーダチャートになっており，多くの脅威観点でセキュリティ達成レベルが不十分なことが明白になった．従来，経営管理者がセキュリティの事業リスクを把握することが遅れ，プロジェクト採算が悪化することが多かった．多軸の評価指標とレーダチャート方式によって設計フェーズまでに採算性にかかわる事業リスクを把握できる効果が得られた．

6.3. さらなる効果

SIAT ではプロジェクトの開発予定や設計レビューの実施時期を従来よりも早期に把握できるようにした．また，専門家の稼働状況と対比することで設計レビューなどの実施時期を調整できるようになった．

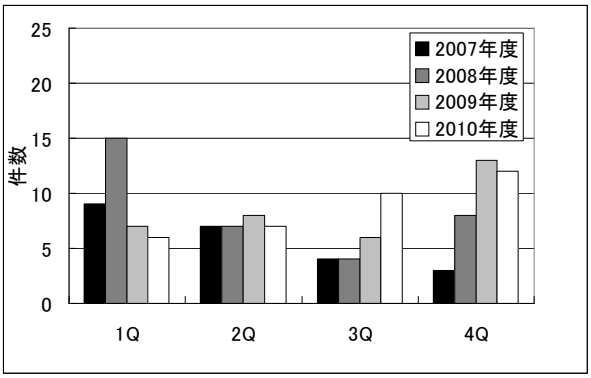

図 4 設計レビュー実施時期の分布

例えば，2008 年度は第1四半期(1Q)には年間実施数の 44%が集中したため専門家の人的リソースが逼迫される状況にあった．一方 2010 年度にはピーク集中が 32%にまで改善した(図 4)．専門家不足のボトルネックが解消し，玉突き的な検査作業の実施遅延が解消した．

7. まとめ

本稿では，富士通で独自に開発した協調型セキュア構築プロセス(CSIP)の概要を述べた．システムインテグレーション事業では顧客調整など重要な役割を果たす SE とセキュリティ専門家が協調し，システム構築の各段階でセキュリティを確保するのが特徴であった．

また，SE と専門家がインタビュー方式でシステム固有のセキュリティ問題の見落としを防ぐ設計レビューが特に重要であるが，セキュリティ問題の理解力のギャップと，セキュリティ達成レベルが不確かなことの 2 つが課題であった．

これら課題を解消するため，類似リスク抽出と，相互比較可能な多軸の評価指標の手法を SIAT に適用した．これらによって，SE の適切なセキュリティ問題の理解が可能になり，またセキュリティ問題を抱える開発プロジェクトの把握が可能になることを示した．

今後の課題には，脆弱性の影響度を反映したセキュリティ達成レベルの改良が考えられる．今回提案した指標化ではチェックリストやセキュリティ対策のカバレッジを評価することに留まっているが，影響度を反映することでより適確な経営管理が可能になると考えられる．

参考文献

[1] JPCERT/CC, "インシデント報告対応四半期レポート" (2011 年 4 月 1 日～2011 年 6 月 30), 2011. http://www.jpcert.or.jp/ir/report.html

[2] Imperva, "Imperva's Web Application Attack Report" Edition #1, July 2011. http://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed1.pdf

[3] 独立行政法人 情報推進機構(IPA), "安全なウェブサイトの作り方 第5版", 2011. http://www.ipa.go.jp/security/vuln/websecurity.html

[4] Michael Howard and Steve Lipner, "The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software", Microsoft Press, ISBN: 0735622140, June 2006.

[5] 大久保, "セキュアなアプリケーション開発のための要求・デザインパターンの提案", コンピュータセキュリティ研究会(CSEC) 第 44 回, 2009.

[6] 大久保, 中山, 綿口, 田中, "セキュリティ要件を充足するソフトウェア開発方式の提案", コンピュータセキュリティシンポジウム(CSS)2006, 2006.

[7] 山岡, 森川, 兒島, 中山, "ページ再現性を考慮した Web アプリケーションブラックボックステスト手法の提案", コンピュータセキュリティシンポジウム(CSS)2004, 2004.

[8] 富士通, "富士通グループ情報セキュリティ報告書 2011", 2011. http://jp.fujitsu.com/about/csr/management/security/reports/

[9] Tversky, A. and Kahneman, D., "Judgment under Uncertainty: Heuristics and Biases", Science Vol18,pp.1124-1131, 1974.