特集論文

Webアプリケーション脆弱性への取組み

竹林信博*
柴田幸治*
山本隆二*

Approach to Web Application Security

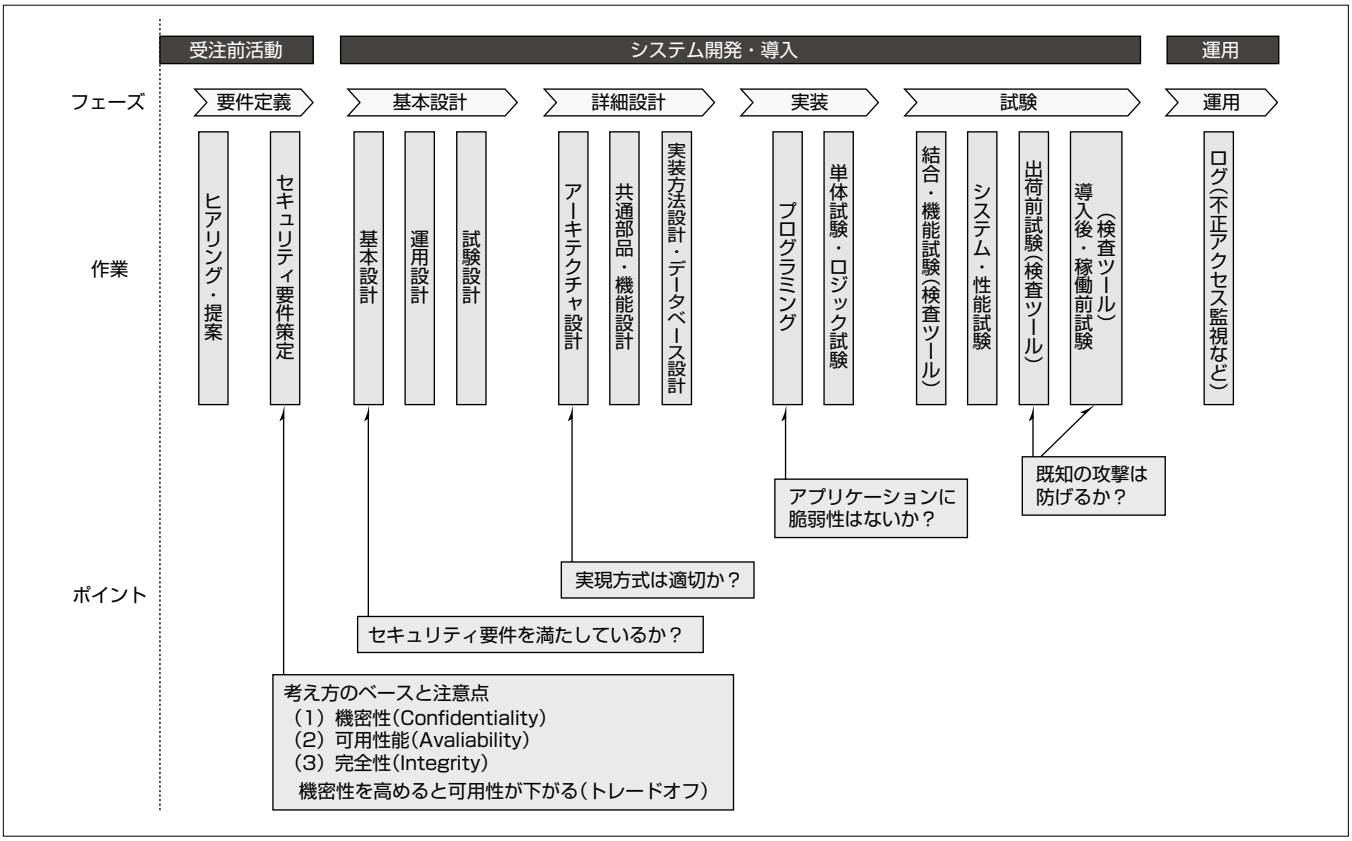
Nobuhiro Takebayashi, Yukiharu Shibata, Ryuji Yamamoto

要　旨

企業において，EC(Electronic Commerce)，EDI(Electronic Data Interchange)，SNS(Social Networking Service)，情報公開などのインターネットシステムは，SFA(Sales Force Automation)，CRM(Customer Relationship Management)に連携可能であるなどWeb(World Wide Web)ビジネスの一端を担う重要な要素であり，端末にアプリケーションをインストールしなくてもシステムを稼働できるWebアプリケーション導入が多くなってきている。近年はクラウドコンピューティングへの移行を始めている企業も見られ，インターネットシステム活用が更に増えることになろう。

一方，サイバー攻撃は年々高度化・複雑化・悪質化しており，最近では企業内の重要情報の不正な取得を目的として特定の標的に対して行われる標的型攻撃が出現するなど，常にセキュリティの脅威にさらされている。万一セキュリティ事故を起こした場合その影響は計りしれないものであり，失墜した信用や信頼の回復に長い時間と莫大(ばくだい)なコストを費やすことになる。

㈱三菱電機ビジネスシステム(MB)では，Webアプリケーション脆弱(ぜいじゃく)性対策に関して，インターネットシステム構築時の上流フェーズである要件定義や基本設計から，下流フェーズであるシステム試験や出荷前試験にいたる作業工程を見直し，またセキュリティ検査を多重に行うことで，品質向上に取り組んでいる。



Webシステム開発時におけるWebアプリケーション脆弱性への取組み

Webアプリケーションの脆弱性を防ぐためには，上流フェーズから下流フェーズまでの一貫した対策が必要である。要件定義フェーズから基本設計フェーズにかけて実施すべきセキュリティ要件策定を怠ると，実装フェーズや運用開始後に脆弱性混入による手戻りが発生することになる。なお，機密性を確保しようとすると可用性が下がってしまうなど，各フェーズで実施する対策は相互に関連しているため，セキュリティ要件策定ではトレードオフを考慮しなければならない。

*㈱三菱電機ビジネスシステム

1. まえがき

Webシステム上に公開・格納されている情報やデータに対するサイバー攻撃は年々高度化・複雑化・悪質化しており，従来の“愉快犯”的なものから，標的型攻撃に見られる“窃盗犯”的なものまで多様化している。

これらのサイバー攻撃を防御するため，Webアプリケーションの開発に当たっては，①設計フェーズでは脆弱性対策仕様を盛り込み，②実装フェーズでは脆弱性対策実装済みのフレームワークを使用し，③試験フェーズでは複数のセキュリティ試験ツールを利用することによって，脆弱性対策を行ったので，本稿ではその概要について述べる。

2. Webアプリケーションの脆弱性

Webアプリケーションの脅威，脆弱性，リスクとは何かを述べ，脆弱性関連の最新情報について述べる。

2.1 脅威，脆弱性，リスクについて

⑴ 脅威とは，Webアプリケーションに対して害を及ぼす，又は害を及ぼす可能性のある事象を指し，サーバに異常なアクセスを行い稼働停止に追い込むものや，不正侵入して改ざんや機密情報を取得する不正アクセスなどが該当する。

⑵ 脆弱性とは，Webアプリケーションが脅威となる攻撃に対して弱い状態，脅威から守るための対策が不完全な状態を指す。

⑶ リスクとは，Webアプリケーションのセキュリティの脆弱な部分から脅威が侵入し，情報漏洩（ろうえい）などによって損失を被る可能性の度合いのことである。

2.2 脆弱性に関する最新情報

Webアプリケーションの脆弱性に関する最新情報については，（独）情報処理推進機構（IPA）が脆弱性関連情報を公開しており，不正アクセス手法などの脅威に対する脆弱性の実態がまとめられている。また，OWASP（The Open Web Application Security Project）が公開している情報では，セキュリティリスクについて報告されている。

⑴ IPAが公開している最新の脆弱性関連情報によると，2012年1月～3月にIPAに届けられた脆弱性関連情報の届出件数のうち，Webアプリケーションに関するものが216件であり，その中で“クロスサイト・スクリプティング”が最も多く，全体の89%を占めている。

⑵ OWASPが公開している2010年度版のOWASP Top 10-2010 Japanese PDF[1]で報告されているWebアプリケーションセキュリティの10大リスクと，その他の考慮すべきセキュリティリスクを**表1**，及び**表2**に示す。OWASP Top 10-2010年度版では“リスク”が高く，重大な影響を及ぼすものに焦点を当てたものになっている。

3. Webアプリケーションに対する脆弱性対策

3.1 脆弱性対策に対する基本的な考え方

情報セキュリティの基本理念は，情報の機密性，完全性，可用性を維持することであるが，脆弱性対策を検討する場合機密性と可用性がトレードオフの関係になりやすい。例えば，全ての情報の持ち出しを禁止にすれば機密性は高まるが，顧客，取引先や協力会社と情報交換をするための持ち出しも不可能となり可用性は下がる。このような場合，情報漏洩事故が発生した時の損失などのリスクと業務上の有益性を比べて，機密性・可用性のどちらを優先するか決定する。

3.2 脆弱性対策のプロセス

MBでは，Webアプリケーション開発に当たって，**2章**で述べた脅威，脆弱性，リスクに対応するため，上流工程から下流工程まで一貫した脆弱性対策の設計，実装，試験を行っている。次に，各フェーズでの実施内容を述べる。

3.2.1 要件定義フェーズ

開発するWebアプリケーションシステムの規模，機密情報取扱いの有無，情報が漏洩した場合のリスクなどを評価し，機密性確保優先か，可用性確保優先かを顧客と協議の上決定し，セキュリティ要件を定義する。

表1．Webアプリケーションに対する10大リスク（2010年度版）

No	内容
1	インジェクション攻撃（SQL，OS，LDAP等）
2	クロスサイトスクリプティング（XSS）
3	不完全な認証とセッション管理
4	安全でないオブジェクトの直接参照
5	クロスサイトリクエストフォージェリ（CSRF）
6	セキュリティの不適切な設定
7	安全でない暗号化によるデータ保存
8	URLアクセス制御の不備
9	不十分なトランスポート層の保護
10	未検証のリダイレクトとフォーワード

SQL ：Structured Query Language
OS ：Operating System
LDAP：Lightweight Directory Access Protocol
URL ：Uniform Resource Locator
出典：OWASP Top 10-2010 Japanese PDF

表2．その他の考慮すべきセキュリティリスク

内容
クリックジャッキング
悪意あるファイルの実行
情報漏洩と不適切なエラー処理
不十分なログ取得とアカウンタビリティ
DoS（Denial of Service，サービス不能）攻撃
同時処理に関する欠陥
自動攻撃に対する不十分な対抗措置
不正侵入の検知と対応に関する不足

出典：OWASP Top 10-2010 Japanese PDF

3.2.2 設計フェーズ

(1) 基本設計フェーズでは，脆弱性対策を盛り込んだシステム全体の外部設計，試験設計，本稼働後の運用設計を行う。主な内容を**表3**に示す。

(2) 詳細設計フェーズでは，具体的な画面の入出力インタフェース，HTTP(Hypertext Transfer Protocol)メソッドの選定，入出力の特性に応じたデータ項目のデータベース設計，セキュリティ対策を考慮した共通部品化設計などを行う。内容を**表4**に示す。

なお，共通部品化設計の具体例として，クロスサイトスクリプティング，SQLインジェクションの脆弱性に対する対策の一つであるエスケープ処理について**表5**に示す。

3.2.3 実装フェーズ

実装フェーズでは，設計フェーズで作成された仕様にしたがい，実装を行う。開発言語がJava(注1)の場合，脆弱性に対する対策があらかじめ実装されている，MB独自のJava製Webアプリケーションフレームワーク(**表6**，**表7**)を基盤としてプログラムの実装を行う。このフレームワークを利用することで，プログラマは特にWebアプリケーションの脆弱性を意識することなく均一なセキュリティ品質を作り込むことができる。

(注1) Javaは，Oracle Corp. の登録商標である。

3.2.4 試験フェーズ

試験フェーズでは，**表8**のツールを利用して試験を行う。

(1) 結合試験・機能試験では，機能単位，入力フィールド単位で確実に目視確認しながら試験を行うため，Odysseus(注2)やParos(注3)を活用している。試験実施者は，試験設計書にしたがってWebブラウザからWebアプリ

表3．基本設計フェーズで行う脆弱性対策

フェーズ	項目	設計内容の例
外部設計	ログインの方式	セッション管理 Secure Cookie システム認証 ベーシック認証
	画面設計	入出力に関わる全般的な制約事項
試験設計	機能試験	データ改ざん試験
	セキュリティ検査	検査シナリオ作成
運用設計	ログ取得	操作ログ取得
	不正アクセス監視	アクセスログ監視
	バックアップ設計	事故後のデータ復旧

表4．詳細設計フェーズで行う脆弱性対策

フェーズ	項目	設計内容の例
詳細設計	画面の入出力インタフェース	最大入力値，最大入力長
		入力禁止文字種
		サニタイジング方式，規則
		画面遷移時のデータの引渡し・データの戻し方法
	HTTPメソッド	POST／GETのうち，極力POSTを使用

表5．エスケープ処理での共通部品化

シーン	置換前	置換後
全ての表示における HTMLエンコーディング	"	&quot;
	&	&amp;
	<	&lt;
SQL文の全挿入文字	;	¥;
	%	¥%
	_	¥_

HTML：HyperText Markup Language

表6．Webアプリケーションフレームワークの機能

機能	内容
基盤エンジン	メール送受信，CSV・Excel(注4)出力，帳票，数式演算，メッセージ送信など
データベースエンジン	簡易問合せ，更新，削除であればSQLの記述は不要，SQLが必要な場合は外部ファイルに記述が可能
画面制御エンジン	Webアプリケーションを開発するために必要な機能を集めたビューコンポーネント

CSV：Comma Separated Values

表7．フレームワークで実装されている脆弱性対策

脆弱性，リスク	対策
SQLインジェクション	バインド変数を使用してパラメータを渡す。
クロスサイトスクリプティング	画面制御フレームワークでサニタイジングする。
不完全な認証とセッション管理	AOP(アスペクト指向プログラミング)を利用して，全てのページを監視し，ログイン状態を確認する。
クロスサイトリクエストフォージェリ	サニタイジングやセッションIDの変更などを行うユーティリティを提供する。
安全でない暗号化によるデータ保存	あらゆるデータを暗号化。方式はAES(Advanced Encryption Standard)である。
URLアクセス制御の不備	URLパラメータのAES暗号化と検証を実施する。URLごとにAOPによる認証チェックを行う。
未検証のリダイレクトとフォーワード	やむを得ず飛び先をリクエストに含む場合は，AES暗号化する。

表8．試験フェーズで利用するツール

ツール	特徴／機能
Odysseus (フリー)	【位置付け】脆弱性検査補助ツール ・プロキシとして動作 ・HTTP／HTTPSの通信を傍受 ・HTTPリクエスト／レスポンスの書換えが容易 ・操作が直感的で分かりやすい
Paros (フリー)	【位置付け】セキュリティ評価ツール ・プロキシとして動作 ・HTTP／HTTPSの通信を傍受 ・脆弱性スキャナ・フィルタ機能 ・簡易レポート機能 ・HTTPリクエスト／レスポンスの書換えインタフェースが分かりにくい
AppScan (商用)	【位置付け】脆弱性検査ツール ・不正なHTTPリクエストを送信し擬似攻撃 ・自動巡回機能 ・自動テスト機能 ・詳細レポート機能 ・セキュリティルール(攻撃パターンに相当)のアップデート ・難易度が高く，講習会受講などが必要

HTTPS：HyperText Transfer Protocol over Secure Socket Layer

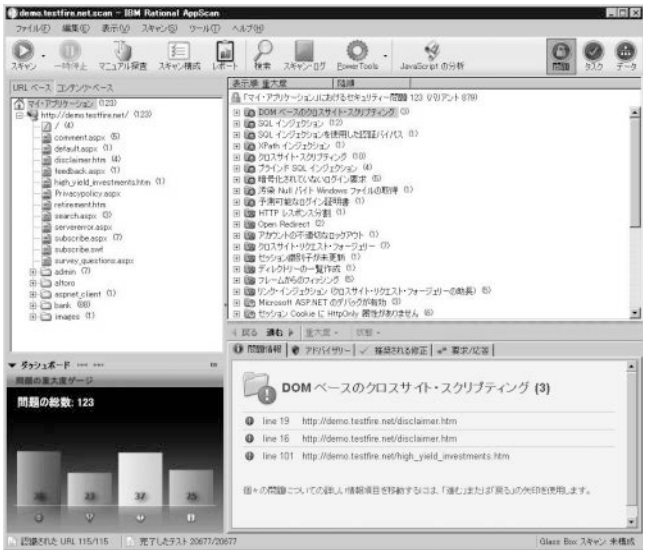


図1．AppScanのスキャン結果例

ケーションに向けてPOSTされたHTTP Headerをインターセプトし，パラメータの値を故意に変更(改ざん)して，バッファオーバーフロー，クロスサイトスクリプティング，SQLエラーインジェクションなどが発生しないかどうか試験する。これによって実装フェーズで脆弱性対策の漏れがないことを確認する。

(2) システム試験や性能試験完了後，試験設計書と実装方式についての事前ヒアリングを行い，検査対象画面の全シナリオを作成し，IBMのRational AppScan(注5)を使用して開発環境で出荷前のセキュリティ検査を行う。

図1にAppScanの画面例を示す。**図1**ではシナリオに従って試験した結果を示し，画面左上は試験対象のURL，画面右上に発見された脆弱性(危険度高の脆弱性はレッド，危険度中の脆弱性はブラウン，危険度低の脆弱性はイエローで表示)，画面右下に発見された脆弱性の詳細，画面左下にサマリーが示されている。原則として検出された高・中・低の全ての脆弱性に対して対策を実施する。なお，当該システム運用上問題とならないような場合，例えば，出力結果のHTMLにコメントが含まれているという脆弱性低の指摘があっても，そのコメントに機密情報が含まれていないような場合には，問題はないため対策を実施しないことがある。

(3) 導入後本稼働前に，MBから検査会社への依頼によって，インターネット上の実環境で本稼働前のセキュリティ検査が行われる。検査環境の相違，使用する検査ツールの相違によって，新たな脅威，脆弱性，リスクが検出された場合は，(2)と同様に対策を行う。

(注2) Odysseusは，bindshell.netから提供されているフリーソフトウェアである。
(注3) Parosは，MileSCAN Technologies Ltd. が開発した無償版ソフトウェアである。
(注4) Excelは，Microsoft Corp. の登録商標である。
(注5) Rational AppScanは，International Business Machines Corp. の登録商標である。

4. むすび

これまで述べてきたプロセスを経てWebアプリケーションの脆弱性が取り除かれるが，次に示す課題も残っており，今後，継続して対策を検討し，より高品質のシステムを提供していく所存である。

(1) 今後，オンプレミスなシステムからクラウドコンピューティングへの移行が進展すると考えられるため，クラウドサービス化するWebアプリケーションのセキュリティ対策(安全性)，性能，可用性について対応方法を検討・実施する。

(2) 最新のサイバー攻撃である標的型攻撃は機密情報の流出につながる可能性があり，対策が急務である。最新動向として，ネットワーク接続ポイントに専用機器を設置し，通過するパケットやメールを監視・分析することで，ウイルス感染や挙動不審な動きを検出するツールが提供されつつあり，そのようなツールを調査し，システム設計時への適用，顧客への提案などを検討する。

(3) 脆弱性対応の手法には，アプリケーション側で対応するのではなくWAF(Web Application Firewall)を利用する方法もある。WAFはソフトウェアとして提供されるものとハードウェアとして提供されるものがあり，今後，調査・検証を進めていく。

参考文献

(1) The Open Web Application Security Project：日本語版OWASP Top 10-2010，Creative Commons(CC) Attribution Share-Alike Free version at http://www.owasp.org（2010）