地方公共団体における情報システム セキュリティ要求仕様モデルプラン （Web アプリケーション）解説書

第 1.0 版

財団法人地方自治情報センター 自治体セキュリティ支援室

目次

はじめに

(1)　本書の位置づけ

「地方公共団体における情報システムセキュリティ要求仕様モデルプラン（Web アプリケーション)」（以下「モデルプラン」という。）は、地方公共団体（以下「団体」という。）における Web アプリケーションソフトウェアの脆弱性をなくし、安全に運用するために必要な最低限のセキュリティ要求仕様事項例をまとめた特記仕様書の雛形と帳票（ツール）の例です。

モデルプランは Web アプリケーションの導入に伴う脆弱性の諸問題を解決するための現実的な仕様例となるように作成してあり、さまざまな用途の Web アプリケーションの中でも共通的に利用できるものを中心にまとめています（ただし、一部の項目は選択を必要とします）。

本書はモデルプランを利用するにあたっての解説書です。モデルプランの利用方法、各要求仕様項目の解説及び注意事項等をまとめてあります。

(2)　背景

情報システムは、住民向けサービスの基盤として欠かせない存在ですが、情報システムを安全に利用する上で避けては通れない問題があります。それが「脆弱性」に関する問題です。

脆弱性とは、情報セキュリティ上の弱点のことであり、脆弱性の問題を放置すると、情報の流出や、ホームページ等コンテンツの改ざん、サービスの停止などの問題を引き起こす可能性があります。一見すると安定して動作しているように見えていても、脆弱性が内在することもあり、情報システムの調達・構築・運用にあたってこの対処をあらかじめ決めておくことは、安定的な運用に欠かせないことです。特に、近年では、Web アプリケーションの脆弱性を狙ったサイバー攻撃の発生が顕著であり、一般のニュースで取り上げられることも珍しくなくなりました。そして、残念ながら一部の団体でも、Web アプリケーションの脆弱性によって、Web サイトを改ざんされるなどの被害が発生しているところです。

地方自治情報センター（以下「当センター」という。）では、各種情報セキュリティ対策支援事業により団体における脆弱性問題の解消を支援して参りました。しかしながら、特に「Web アプリケーションの脆弱性」については、解決を図ることが難しい課題であるのが現状です。

独立行政法人情報処理推進機構（IPA）からも、「地方公共団体のための脆弱性対応ガイド [1]」がリリースされており、その問題の根の深さが指摘されているところです。

同ガイドでは団体の脆弱性対策に関する実態調査を目的としたアンケートが行われており、そのアンケート結果から次の課題を挙げています。

1　『「地方公共団体のための脆弱性対応ガイド」などを公開』
http://www.ipa.go.jp/security/fy23/reports/vuln_handling/index.html

脆弱性対策に関する課題（IPAプレスリリース[2]より抜粋）

ア 突然発覚する脆弱性への対応について、幹部の理解が得られない。
イ 情報システムを管理する担当部門が脆弱性対策の必要性を理解していない。
ウ 人事異動により、経験を積んだ IT 担当部門の職員の知見が活かせなくなる。
エ 脆弱性が見つかった場合の対策の実施や公表に係る方針が定まらない。

ここで IPA のまとめた課題について、1 つ 1 つ見ていきます。

ア に挙げた「幹部の理解」に関して、IPA の行ったアンケート調査によると、市（特別区含む）では約 4 割が、脆弱性への対応について幹部の理解を得るのが難しいことを「重要な課題」、「課題の 1 つである」と回答しています。「情報システムは時とともに安全性が低下する」ということと「脆弱性は突発的に第三者によって見つけられることがある」ことが組織幹部になかなか理解されないという現状が浮き彫りになっています。

イ に挙げた「脆弱性対策の必要性への理解」については、脆弱性によって引き起こされる様々な問題について、自身が対処を担わなければならないことに関する認識の不足や、開発・運用事業者に対するあいまいな取り決めや不十分な合意形成のみで開発・運用をすることに対して警鐘が鳴らされています。

ウ に挙げた「人事異動の問題」は、どの団体でも共通の悩みとなっています。IT・システムへの理解のある人材育成には時間がかかり、一部の団体では業務手順のマニュアル化などによる経験の引き継ぎが行われているものの、人事異動により担当者の経験蓄積がなかなか進まないという団体も多いようです。また、オープンシステムへの移行により従来情報政策部門が担っていた情報システムの予算化・企画・開発・運用を業務部門が担うよう役割が変化したことが IT・システムに関する知識の風化に拍車をかけ、先に挙げた、開発・運用事業者にすべて依存せざるを得ない状況を生じさせています。

そして、エ でも挙がっているような状態は、脆弱性が見つかった際の備えがないと、より深刻な事態を招く可能性を想起させられます。システムをすぐに停止したほうがよいほどに危険な脆弱性が発見されても、その危険性を正しく判断できなかったり（技術知識不足・人材不足）、改修するための予算の都合が付けられていなかったり（幹部の理解不足）、委託先との調整が進まなかったり（開発・運用事業者に対するあいまいな取り決めや不十分な合意形成）といった、先に挙げた問題が顕在化します。

問題点を整理すると、次のようになります。

問題点の整理

「対策を実施するための予算化・役割分担が困難」（**幹部理解、予算、人事制度上の問題**）
「対策を実施する担当者の技術知識が追いつかない」（**人的リソースの問題**）
「対策を実施するための方法がわからない」（**手順の問題**）

[2] http://www.ipa.go.jp/security/fy23/reports/vuln_handling/index.html

しかしながら、予算、制度上の問題や、人的リソースの問題は短期的に解決できる問題ではありません。一方で、脆弱性の問題は現在進行形で進んでいる問題であり、それによって引き起こされる情報漏えい等は住民生活を脅かす危険性をはらんでいます。

そこで、当センターでは脆弱性対策がなかなか進まない原因の中の 1 つ「必要な対策を実施するための方法がわからない（手順の問題）」について、解決に向かわせる方法の 1 つとして、モデルプランを示すこととし、現在特に問題になることが多い、「Web アプリケーションの脆弱性問題」にテーマをフォーカスし、これを作成することとしました。また、モデルプラン及び本書を通じて「対策を実施する担当者の技術知識が足りない（人的リソースの問題）」に関しても間接的に貢献することができるものと考えています。

(3)　セキュリティ要求事項検討時の課題とモデルプラン作成のねらい

さて、脆弱性対策を進めるために、手順の問題を解決することまでを整理しましたが、システム発注におけるセキュリティに関する要件、特に本書が示す「Web アプリケーションセキュリティに係る要件」は、Web アプリケーションに達成してもらう必要がある業務内容を記述する「機能要件」とは異なり、仕様書に具体的かつ体系的に記述することが難しい要件の 1 つです。こうした機能要件以外の要件のことを一般に「非機能要件」と呼びます。非機能要件にはさまざまな要件があり、例を挙げると次のようなものがあります。

・可用性に関する要求　　（例：MTBF、MTTR[3]など）
・性能に関する要求　　（例：利用者の操作におけるレスポンススピードなど）
・拡張性に関する要求　　（例：パフォーマンスを 2 倍にしたいときにその手段があることなど）

そして、モデルプランのテーマでもある「セキュリティに関する要求」があります。

本来、非機能要件を提示することは、到達すべきレベルを提案者に示し、あるべき姿を“見える化”することで、発注者・応札者（受注者）双方で認識を一にすることが目的です。

しかしながら、このセキュリティに関する要求事項を見える化することは、例えば次のような理由により困難とされています。

セキュリティ要求事項検討時の課題

課題.ア「セキュリティ要件として何を要求すべきかわからない。何があるのかわからない。」
⇒技術知識の問題
課題.イ「要求事項に漏れがあるかもしれない。どうやればいいかわからない。」
⇒網羅的な記述の困難さ、ノウハウ不足の問題
課題.ウ「団体としてどの程度のレベルを要求すべきかがわからない。」
⇒統一的な基準の不存在の問題、曖昧ゆえの事業者選定上の不公平問題（後述）

このような問題から従来の場合、セキュリティ要件が「セキュリティに関して万全を期すこと」といった趣旨の漠然とした要求仕様となり「実現すべき具体的事項」、「達成すべき基準」、「各項目間に

[3] MTBF：Mean Time Between Failure(平均故障間隔)、MTTR：Mean Time To Repair (平均修復時間)

おける優先順位」といった内容を曖昧にしたまま記述せざるを得ない団体が多かったのが現状です。

そして、そのような要求仕様に対して提案事業者(受注事業者)は各社独自の基準に基づくバランス感覚で“万全を期す”の意味するところを解釈し、各社各様で“万全を期す”に応えてきました。

モデルプランは先に挙げた、課題.ウの解決のための一里塚を築き、脆弱性対策に関する統一的な基準を提供します。

また、漠然とした“万全を期す”仕様を調達に使うことによる副作用として、セキュリティ意識が高く、技術レベルの優れた事業者は万全を目指してその要求仕様に応じているのに対し、価格競争を優先する事業者では一般的な基準から比べると全くセキュリティに対する考慮に欠けた対応を取っているケースも見受けられるところです。（これは、当センターが平成 20 年度～平成 22 年度に実施した「ウェブ健康診断事業」の事業内で一部の事業者に感じられたところです。）これはセキュリティに関する提案時の明確な要求内容と評価軸がないことが原因であり、事業者によるレベルの異なるまちまちの対応は、元を正せば、発注者のあいまいな要求が原因という構図が見て取れます。

モデルプランは先に挙げた課題.イの解決のための地図やコンパスとなる、熟慮した仕様例を提供します。

本書の作成にあたっては、団体側にある問題を克服しつつ、先に挙げたような提案事業者間の意識格差を是正するとともに、Web アプリケーションの脆弱性に関する問題について発注者側・受注者側双方を解決に向けて一歩を踏み出していただくことを目的として、作成しました。

そして、今まで自主的に行われてきた各団体の脆弱性対策のほか、当センターの既存の脆弱性診断事業に加え、モデルプラン及び本書を提供することにより、脆弱性の問題のすべてを解決できないまでも、1 つ 1 つ積み上げるように改善が図られることを当センターは期待し、また願っています。

モデルプランにより克服できる課題・残る課題のまとめ

ア モデルプランと本書を通じて、各団体職員において必要な技術知識獲得のきっかけとしていただく。

⇒技術知識不足問題は、徐々に技術知識を蓄積していただくことを期待。残課題。

イ 脆弱性対策に関する網羅的なノウハウの提供

⇒手順の問題が解決。

ウ 統一的な基準で、応札事業者におけるセキュリティ技術の詳細を提案に含め、一律の評価軸のもとで評価が可能となる。

⇒統一基準不在問題、曖昧ゆえの事業者選定上の不公平問題が解決し、セキュリティ対策がしっかりとした事業者が受注する。

(4)　モデルプランの効用

モデルプランは、団体で Web アプリケーションの発注時に作成される、業務要求仕様などをまとめた「仕様書（機能要件等が記載された仕様書)」を親文書とし、本書を参考にモデルプランの内容を一部加除の上で「Web アプリケーションセキュリティに関する特記仕様書」を各団体で作成して仕様書に添付いただくことにより、納品されるソフトウェアに SQL インジェクション、クロスサイト・スクリプティングといった「Web アプリケーションの脆弱性」の混入の予防効果が期待できます。この際、団体に対して各脆弱性に対する対策方法の知識等、深い技術知識を要求はしていません（あれば尚良いことは言うまでもありません)。

そして、先に挙げたような事業者間の意識の違い、セキュアコーディング技術の違い等をシステム選定時における評価軸の 1 つとして据えて事業者を選定し、納品後（運用時）に万一新たに脆弱性が発見された場合でも改修を計画的に進めることを事業者に約束してもらうことでセキュアな Web アプリケーションを導入・維持できるようになります。

またモデルプランは、団体における Web アプリケーションの開発、運用保守の調達段階で事業者に実現すべき具体的事項と、達成すべき基準を示してあるので、その後の運用フェーズにおいて受注者・発注者で認識を一つにしてスムースな対応を図っていただくことを目指すことができます。

モデルプランの効用（まとめ)

ア 特記仕様書を添付するだけで一定のセキュリティレベルを要求することができる（あまり詳しい技術知識を持っていなくても利用可能)。
イ 脆弱性を作り込まないことを、提案事業者（受注事業者）に約束させることができる。
ウ 発注時（事前）に対策を考えることで、納品後の運用（事後）に備えることができる

(5)　モデルプラン利用上の注意点

団体がモデルプランを利用するにあたってはメリット・デメリットがあることを十分考慮する必要があります。注意すべき点については『第 3 章　調達プロセス中における帳票(ツール)の利用場面及び効果並びに注意事項』にて、調達プロセスの順を追って説明します。

なお、本書はあくまでも「Web アプリケーションの脆弱性の混入を防ぐ」ことに特化した特記仕様書の解説書であり、他の、本書が網羅していないセキュリティ要件については、契約書や本書の親となる業務要求仕様など他の仕様書をまとめた「仕様書」に記載されていることを前提としています。本件については、『第 2 章 モデルプランの対象範囲、利用想定 (4)』も参照してください。

(6)　本書の構成

本書の構成は次のとおりです。

第 1 章　提供ツールについて

モデルプランで提供する各種ツール（帳票）の概要、ねらいを解説します。

第 2 章　モデルプランの対象範囲、利用想定

モデルプランをご利用いただくに当たっての前提を解説します。

第 3 章　調達プロセスにおける帳票（ツール）の利用場面及び効果並びに注意事項

調達プロセスごとにセキュリティを検討するにあたってのセキュリティに関する一般的注意事項の概要とともに、各ツールの利用法の想定などを解説します。

第 4 章　逐条解説

モデルプランの特記仕様書（雛型）に記載の要求仕様について、それぞれを解説します。

第 5 章　他の要件の参考情報、参考文献

モデルプランで網羅していないセキュリティ要求等の検討に役立つ URL や、モデルプラン作成に当たって参考とした情報を紹介しています。

第1章 提供ツールについて

(1) 提供する帳票（ツール）の作成経緯概要

当センターでは、平成 20 年から平成 23 年まで、「ウェブ健康診断」事業を通じて、団体の Web アプリケーションの脆弱性を改修するための支援を続けてきました。そして、同事業における各団体の Web アプリケーション脆弱性検出傾向や、再診断（脆弱性を検出した団体を対象に、当該脆弱性改修後に実施する脆弱性の再検査）の結果、診断結果の年度推移、先述の IPA のアンケート調査結果等をかんがみ、Web アプリケーションの構築前の段階でできること、すなわち、「脆弱性を作りこまないための要求仕様の作成と展開」が団体のセキュリティレベル向上に必要であるという結論に至ったところです。

そこで、当センターでは実効性のある Web アプリケーションセキュリティの要求仕様とは、どうあるべきか、また、各団体でその要求仕様を採用する際に発生する問題等について検討するため、有識者及び一部団体職員による Web アプリケーションセキュリティ要求仕様等検討委員会（『モデルプラン別紙 8』参照）を設立し、「特記仕様書（雛型）」及び「特記仕様書（雛形）を利用するための各種帳票（ツール）」から成るモデルプランを作成しました。

(2) 提供する帳票（ツール）の目的

まず、Web アプリケーションの脆弱性の課題を解決するために、調達プロセス、委託業務の履行プロセスにおいて以下のような枠組みを考えました（次ページ図 1 参照）。

① 脆弱性対応のために**何を実施するか**を発注者が明確にする
団体から事業者に「対処する対象の脆弱性」、「実装するセキュリティ機能」を示す。

② 提案者に**実施の約束**をしてもらう
提案者から団体に特記仕様書に対する「遵守状況一覧」と「セキュリティ実装方針」を示してもらう。また、約束できない部分がある場合は提案の時点で「重要事項説明書」の提出をもって説明してもらう。事業者決定後は提案時の方針どおりに実装してもらう。

③ **約束の履行状況を報告**してもらう
受注者から団体に②で約束した内容の履行状況を「実装状況報告書」の提出をもって説明してもらう。また、脆弱性検査の結果も報告してもらう。

④ もし対処の**約束に不備**があった場合、それを**修補**してもらう。
対処する対象の脆弱性に関して全て対処済みであることは納品時に「実装状況報告書」で示してもらうが、何らかの理由[4]で運用中に脆弱性が発見されることがある。その際はこれを受注者に追加費用なしで修補してもらう。

[4] 脆弱性対応ができていないことが後日判明することがあり、次のような場合が考えられる。
・脆弱性を発見した第三者の通報により判明する場合
・LASDEC の脆弱性診断等の実施により判明する場合
・実際にセキュリティインシデント（サイバー攻撃被害の顕在化）が発生し、判明する場合

図 1 提供帳票（ツール）の目的

(3)　モデルプランのポイント

モデルプランの目指すものは、安全性を実現するためのあるべき姿を発注者が仕様として要求し、出荷検査及び検収（受入れ検査）で、その仕様（あるべき姿）が実現されていることを確認することです。

しかしながら、脆弱性の完全な検査は容易ではなく、脆弱性診断の専門事業者がこれを実施しても漏れることもあり、検査結果の完全性を保証しないことが多いのが現状です。

この現状を考慮して、モデルプランでは「セキュリティ保証期間」という期間を要求仕様中に示し、同保証期間中に何らかの理由で発見した脆弱性については追加の費用なしに修補することを求めています。

また、Web アプリケーションシステムのセキュリティを保つ上では、OS やミドルウェアの脆弱性対策も必須であり、そのためには、これら OS やミドルウェア等第三者ソフトウェアの脆弱性修正パッチ（以下「パッチ」という。）がセキュリティ保証期間中、提供される必要があります。このため、サイト構築のために（受注者にとっての）第三者から調達するソフトウェアについては、セキュリティ保証期間中のパッチ提供が予定されている製品か否かを確認して選定することも求めています。

なお、何らかの理由で提案者（受注者）が要求を実現できない場合、提案者は「重要事項説明書」をあらかじめ提示することで、発注者が「想定されるリスク」や「発生する追加費用」等を事前に把握できるようにしています。

各帳票の概要は、次ページ図 2 を参照してください。

資料名	説明
特記仕様書（雛形）	「Webアプリケーションセキュリティ特記仕様書」の雛型
別紙１ 脆弱性リスト	対応すべきWebアプリケーションの脆弱性一覧。 特記仕様書（雛型）本文中で参照される。
別紙２ （参考）要求仕様チェックシート	特記仕様書（雛型）で一部の「選択の余地がある項目」等に関して意思決定をする際の検討補助ツール。
別紙３ 遵守状況一覧	・特記仕様書を利用して発注をする際の、特記仕様書記載項目の遵守状況を事業者が記載するもの。 ・事業者が提案書とともに提出する。
別紙４－１ 重要事項説明書	・特記仕様書に記載の項目で、提案者が遵守できないものがある場合に記載する帳票。 ・提案者が提案書とともに提出する（モデルプランでは別紙4－2にてサンプルを示す）。
別紙５ セキュリティ実装方針（サンプル）	・特記仕様書で示す「脆弱性リスト（別紙１）」に対応をするための実装方針を記載したもの。 ・事業者が提案書とともに任意書式で提出する（モデルプランではサンプルのみ示す）。
別紙６ 実装状況報告書（サンプル）	・提案時に提出した「セキュリティ実装方針（別紙５）」の実装状況を報告するための帳票。 ・事業者が最終納品物の１つとして任意書式で提出する（モデルプランではサンプルのみ示す）。
別紙７ 契約書への追加条文例	「特記仕様書（雛型）」を利用して調達を行う際、契約書に追加する条文の例　※利用は任意

図 2 モデルプランが提供する帳票（ツール）一覧

第2章 モデルプランの対象範囲、利用想定

本章では、モデルプランを利用する対象範囲及び利用想定等を記載します。

(1) 対象者

モデルプランの対象団体、対象読者想定は次のとおりです。

ア 対象団体想定

モデルプランを利用する団体像について、団体の人口規模等、特に定めて想定していません。どのような規模の団体でもWebアプリケーションの達するべきセキュリティレベルを目指しています。

イ 対象読者想定

- Webアプリケーション及びプラットフォームの調達を担当する職員
- 情報セキュリティ担当課職員
- 情報政策担当課職員

(2) 対象サービス

具体的なWebアプリケーション例、設置形態は次のとおりです。

ア Webアプリケーション例

- CMS（Content Management System）
- 電子申請
- 電子調達（入札）
- 図書館蔵書検索、(貸出)予約システム
- 施設予約システム
- 地域SNS、掲示板
- 粗大ごみ収集等の申込み関係
- GIS（Geographic Information System）
- 各種内部(庁内)業務系情報システム　等

イ Webアプリケーションの設置形態

- インターネットへの公開サービス
- 内部（庁内LAN）設置システム

  モデルプラン及び本書は、特に断りのない限りインターネットへの公開サービスを想定した記載をしていますが、モデルプランにある要求仕様の内容自体は外部に公開しているWebアプリケーションに限定していません。内部LANへ不正侵入されてしまった際の対応、内部に不正アクセスする者が存在した場合のほか、マルウェア感染による内部システムへの攻撃等も脅威として認識し、内部LANにおけるWebアプリケーションもモデルプランに記載されている程度のセキュリティ要件を満たすことが妥当と考えます。

(3)　発注・導入形態想定

モデルプランは次のような場合にご利用いただくことを想定しています。

ア　発注形態

次のいずれかの発注形態にご利用いただくことを想定しています。

- 構築及び運用保守契約を同一企業に発注する場合（分割発注しない）
- 構築のみ発注する場合（運用保守は別発注、分割契約）

なお、分割発注の場合は、モデルプラン利用時、特記仕様書（雛型）の一部を削除する必要があります。詳細は『第 4 章 逐条解説』を参照してください。

イ　導入形態

次のいずれかの導入形態にご利用いただくことを想定しています。

- パッケージ Web アプリケーション及びそのプラットフォームの導入
- 一部をカスタマイズ開発した Web アプリケーション及びそのプラットフォームの導入
- 委託事業者のスクラッチ開発による Web アプリケーション及びそのプラットフォームの導入
- 統合型パッケージの利用契約で使用される Web アプリケーション及びそのプラットフォームの導入

なお、次については本書そのままの内容で利用できる形態になっていませんが、『3.Web アプリケーション脆弱性対応』、『4.セキュリティ機能』の記載を、各サービス採用検討時の参考情報として活用することも可能と考えています。

- ASP サービス
- クラウドアプリケーション（SaaS サービス）　等

(4)　モデルプランに含まれていない内容

例えば次の事項についてはモデルプランでは言及していません。下記のようなモデルプランに記載のない、システムに求めるべきセキュリティ要件については『第 5 章 他の要件の参考情報、参考文献』に記載の URL 等を参考に、各団体で必要に応じてご検討ください。

- 開発環境に対するセキュリティ要求
- システムを設置する環境の物理的セキュリティ要件
- ネットワーク構成要件
- 可用性に対する要求（許容する停止時間の規定、システム・ネットワークの冗長化等）
- 完全性に対する要求（バックアップ等）
- 保護すべき情報資産に係る考え方
- 外部委託に伴う個人情報の取扱要件

(5)　（参考）モデルプランと他資料を比較したときの位置づけ

情報システムの取引・契約においては、経済産業省「情報システム・モデル取引・契約書[5]（以下「経

[5] http://www.meti.go.jp/policy/it_policy/keiyaku/

産省モデル取引・契約書」という。)」が優れたモデルを提示しています。この経産省モデル取引・契約書と比較した場合にどのような違いがあるかを把握することは、読者にモデルプランの効用及び目的理解の助けになると考えます。経産省モデル取引・契約書と、モデルプランを対比すると、次の点が異なっています（表 1 参照)。

表 1 経産省モデル取引・契約書 との主な違い

	経産省モデル取引・契約書	モデルプラン
モデル対象	情報システム全般	Web アプリケーション及びプラットフォームのみ
セキュリティ要求仕様項目	指定・例示なし （検討が必要として読者に委ねている）	例示有 （最終判断は読者に委ねる点は左と同じ）
瑕疵担保期間	例示なし （〇カ月と記載）	瑕疵担保期間以外に「セキュリティ保証期間(※)」を提示。 （また、それを 5 年と例示）

※「セキュリティ保証期間」の詳細は、別途『第 4 章 1.2 本システムのセキュリティ保証期間について』『第 4 章 8.保守要件』の解説を参照。

(6)　（参考）モデルプランに記載された具体的セキュリティ要求仕様の選定基準

残念ながら、真の意味で「100%安全な（脆弱性のない）Web アプリケーション及びプラットフォーム」の要件を定義する方法は知られていません。そのため、モデルプランを利用しても 100%脆弱性のない Web アプリケーション及びプラットフォームの導入はできません。

モデルプランには、例えば次のようなものへの対策は含まれていません。

- 未知の脆弱性
- 新しく発見された脆弱性でまだ対処方法が確立していない脆弱性
- “脆弱性である”と定義できるかどうかわからないグレーゾーンの脆弱性

これらの脅威は将来顕在化する可能性はありますが、モデルプランでは現時点で発注者が現実的に求めることができる要求仕様の要件を例示しています。

モデルプランにおける各セキュリティ要求仕様項目（及び内容）は、次のように選定しました。

現時点で攻撃に悪用可能で、既に攻撃に使われている手法（図 3（A））
　又は
現時点で攻撃に悪用可能で、まだ使われてないが将来攻撃に悪用される可能性のある手法（図 3（B））
　に対して、
対策の実現が可能 かつ 大幅コスト増にならない現実線の対策のうち、委員会で対応を要すると判断した項目（図 3（C））。

この基準によりモデルプランで選定した項目のイメージは下図 3 の（C）のとおりです。また、モデルプランにおける技術的なセキュリティ要件としては、「脆弱性対応に関する要件」と、「セキュリティ機能に関する要件」の 2 種類があります。

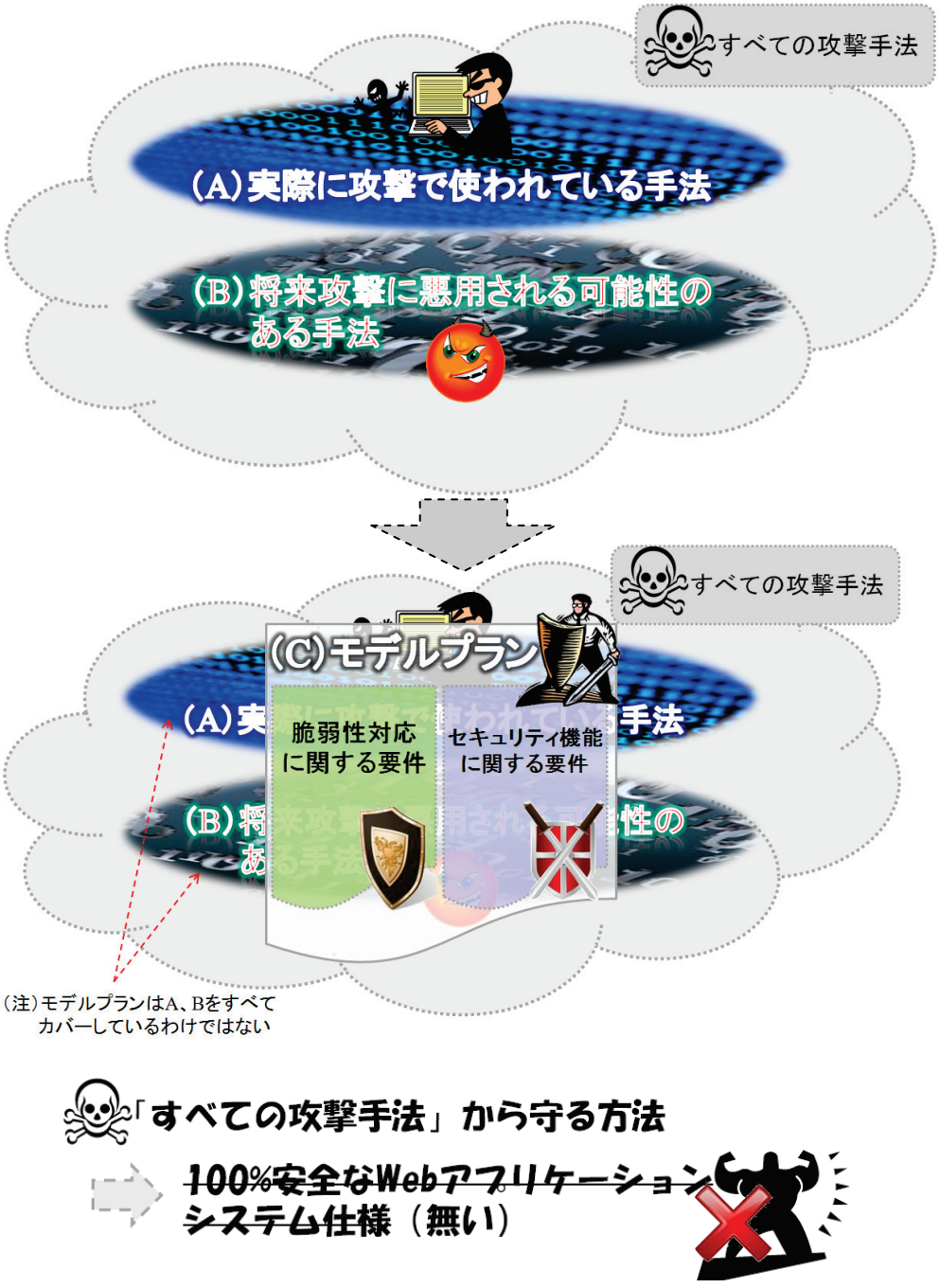

図 3 モデルプラン各項目選定範囲イメージ

また、モデルプラン（特記仕様書（雛型））の一部セキュリティ機能に関する要件には、実施するとなお良いが、実施しなくとも最低限の対策とするには支障がない項目で、各団体にて選択の余地がある項目（オプション）があります。これは対象となる Web アプリケーションが取扱う情報の重要性に合わせて選択することとした項目です。オプションとした項目はモデルプランの脚注に（オプション）と表記し、本書解説にもその旨を記載しています。

第3章 調達プロセスにおける帳票（ツール）の利用場面及び効果並びに注意事項

Webアプリケーションの導入時に、セキュリティを検討する際に留意すべきことは、Webアプリケーションの脆弱性に起因する事故の被害者は、職員ではなく住民であるということです。たとえ受注者に被害発生時の賠償責任を負わせたとしても、住民にとっては取り返しのつかない事故となる可能性もあります。単に発注者としてだけではなく、住民目線から仕様書を定めるという視点も必要なことは言うまでもありません。

「100%安全なセキュリティは存在しない」ことを前提にすると、稼働させるWebアプリケーションについてリスクコミュニケーションの観点からの検討も不可欠です。「守るべき情報（秘密情報）」の概要を調査し、万が一のときの被害の大きさを想定しながら、各団体のセキュリティポリシー等のセキュリティ規程類に基づいて受容できない脅威を見定め、「受容する脅威」を決定するという検討を辿ります。

そして、なぜその脅威を受容したのか、誰が受容すると決定したのかを記録することも肝要です。特に個人情報保護条例における収集禁止情報（例「思想、信教及び信条に関する個人情報並びに社会的差別の原因となる個人情報については、収集してはならない。」）を間接的に取得する可能性がある業務のWebアプリケーション（例：図書館の蔵書検索・貸出予約システム等）については、より慎重に受容する脅威を検討する必要があります。

この受容する脅威がすなわちセキュリティレベルを表し、住民向けにもわかりやすく説明することからリスクコミュニケーションが始まります。セキュリティレベルは、万が一のときに被害を受けることになる関係者（住民）の意向も踏まえるべきです。

そして、到達したいレベルが技術的に不可能であったり、想定外に高額であったりすることで、目標とするセキュリティレベルに達成する見通しがないときは、あえて情報システムの導入を一時見送ると決定することも、大きな選択肢の1つとして視野に入れるべきでしょう。

(1) Webアプリケーション構築の流れ

先述のとおり、モデルプラン及び本書は、Webアプリケーションの調達にあたり発注者が提示する仕様書の一部を構成する特記仕様書を各団体で作成することを支援する目的で作成したドキュメントです。

本来、セキュリティ仕様は調達事務の各段階を経ることで具体化されてゆくものです。モデルプランを利用する際も、その基本は変わることはありません。即ち、モデルプランをそのまま利用する際も、その意図するところを理解し、適正に活用されることを期待します。

次ページの図は、通常調達において使われる主な帳票と、モデルプランで提供する帳票の利用場面イメージです。以降のページではそれぞれの段階の各種パラメータの設定にかかる判断事項とセキュリティ上の留意点の概要を示します。

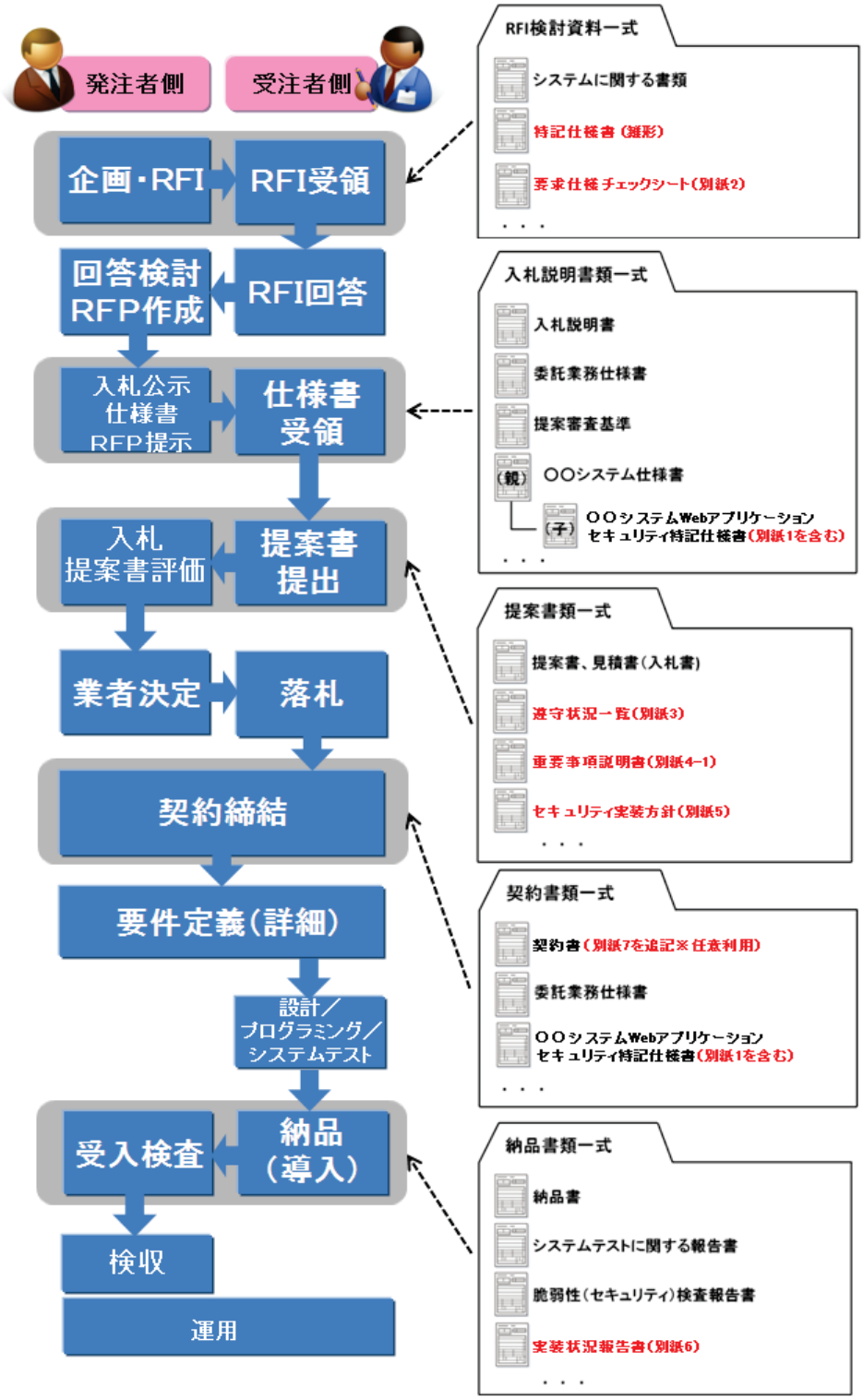

図 4 Web アプリケーション構築の流れと、利用する帳票（ツール）の関係
（赤で示したものが、モデルプランに含まれる帳票です。）

(2) Web アプリケーションの企画

ここでは、Web アプリケーションの企画段階における留意事項について記載します。

ア 企画立案段階におけるセキュリティの取扱い

この段階で参照するのは、個人情報保護条例とセキュリティポリシー（基本方針及び対策基準）に規定された当該システムが取り扱う情報の遵守すべき事項又は当該システムが取り扱う情報と同等の情報を取り扱う類似のシステムに適用されているセキュリティの規定です。

当該システムで取り扱うことになることが想定される情報から、その類型の情報のセキュリティポリシーにおける規定を調べます。また類似システムに適用されているセキュリティレベルを調べ、当該システムに適用するセキュリティレベルを検討・決定します。

この結果を『別紙 2 要求仕様チェックシート』の秘密情報の欄に記入しておきます。なお、モデルプランでは「秘密情報」の定義は親文書である仕様書に書いてあることを前提としています。

イ 情報提供依頼（RFI）

企画が終了すれば、外部事業者に対して情報提供を依頼することになります。通常は外部事業者に対して機能要件を示し、実現可能性、実現方法、概算費用の見積もりを求めますが、あわせて非機能要件についても情報提供を依頼します。特に脆弱性対応については、発注者として求める水準を示して依頼する必要があり、この際、特記仕様書（雛形）を利用することでこれを求めます。

RFI を通じて特記仕様書(雛形)をアレンジし、企画するシステムの「Web アプリケーションセキュリティ特記仕様書」を仕上げていきます。

ウ 費用の見積もり

企画立案段階でのもう 1 つの作業は、概算費用の見積もりですが、ここでも要求するセキュリティ対策の概要について示すことが不可欠です。本書記載のセキュリティ要件を採用した際は、脆弱性検査費用、テスト費用が追加で必要となることも想定されます。特記仕様書（雛型）等を提示して概算費用の見積もりを依頼します。なおこの際、必要に応じて保守契約費用についても依頼しておきます。

当該システムが稼働している期間は、当初に意図した安全性が確保されなければならないところです。このためには全稼働期間にわたる適正な保守管理が不可避であり、この保守管理にかかる費用も見積もることが肝要です。稼働期間を明確に定められない場合でも、少なくとも導入後数年間の見通しは明らかにしておくべきです。

(3)　入札

ここでは、RFI の回答を参考としながら、発注仕様（RFP）を整理し、事業者の選定を行う際の留意事項について記載します。

ア　仕様の重要性

例えば、十分な能力を持ち、セキュリティ技術に明るく、信頼できる事業者を選定できる方法があれば、暗黙のうちに十分なセキュリティ対策を講じてもらうことが期待できます。しかしながらセキュリティに関して信頼できる業者を選定することがいかに困難かは言うまでもありません。特に団体では事業者の絞り込みには競争参加資格を定める以外に方法がないため困難を伴います。

また、よくある選定条件として導入実績を問うことがあります。それ自体は必要ですが、その実績が失敗実績である可能性も頭に置いておく必要があります。真に信頼できる事業者か否かを事例紹介等から各事例団体に評価を聞く方法もありますが、それもまた様々な配慮から正当な評価を聞けるとは限りません。

したがって、団体が発注者として入札過程でセキュリティに配慮したシステムを導入するにあたってできることは、発注者が求めるセキュリティ要件を実現する十分配慮された仕様を定めるほかないと考えます。モデルプランでは十分検討された具体的な仕様が定められています。モデルプランの内容詳細については『第 4 章 逐条解説』を参照してください。

なお、提案型の入札の場合は機能面の評価の高低と、セキュリティ面での評価の高低が異なることがあります。機能要件、非機能要件いずれも必須項目と提案項目（任意項目）を明確に分離して明記し、重視する項目を示すことが不可欠です。必須項目を満足していなければ、たとえ 1 項目であったとしても失格とし、採点しないのが原則です。そのため、仕様の決定においてはどの項目を必須とするかは慎重に検討する必要があります。本書を通じて各要件を吟味していただき、RFI 等を通じて当該事項を必須とするか、任意とするか検討してください。任意項目に対する代替案の提出については『別紙 4-1 重要事項説明書』を利用することを想定しています。

ただし、仕様書の内容と代替案が同等か否かの判断は時に困難を伴います。

イ　サービス品質保証制度（Service Level Agreement：SLA）

SLAには保証型と努力型が存在します。セキュリティについては基本的に保証型である必要があります。

「情報システムに係る政府調達へのSLA導入のガイドライン[6]（平成16年3月）」には、セキュリティ項目として、ウイルス情報の把握、パターンファイルの更新、重大障害の件数、障害復旧時間、障害復旧時間遵守率が例示されています（表2参照）。

表 2　「情報システムに係る政府調達への SLA 導入ガイドライン － サービスレベル評価項目の設定例」から抜粋

サービス分類	サービスレベル評価項目（例）	サービスレベル要求水準（例）
セキュリティ	ウイルス情報の把握	ベンダー検知後1時間以内
	パターンファイルの更新	ベンダーリリースから6時間以内

6 http://www.meti.go.jp/policy/it_policy/tyoutatu/

	重大障害の件数	0回／年
	障害復旧時間	6時間以内
	障害復旧時間遵守率	95%以上(4時間以内)

モデルプランでは、ここにないソフトウェア（OS、ミドルウェア、アプリケーション、Web アプリケーション等）のパッチの適用に関する要件を追加しました。

パッチが供給されれば即刻適用されることを期待したいところですが、現実にはパッチ適用にあたり受注者側で動作確認テストを必要とするため、SLA では相当の余裕をもっている例が多くあります。しかし、この放置されている期間こそ最も危険な期間でもあります。

モデルプランでは次のような内容を入れています。

- 万一『別紙 1 脆弱性リスト』に記載の脆弱性がセキュリティ保証期間中に発見された場合、受注者が追加費用なしに修補対応すること
- 第三者提供のソフトウェアの脆弱性についても同ソフトウェアを採用した受注者にその保全責任を求める。
- パッチが供給されたことや、パッチの修正内容、その影響に関する連絡は 1 週間以内とする。
- パッチの適用は 2 週間以内とする。これが不可能な場合、回避策・代替案を提案、実施する。

なお、モデルプランで示す SLA の要求項目に相当する内容は、パッチにかかわる部分のみで、SLA で定めるべき項目の一部分です。また、SLA で定める値は関係するシステムで同じ水準であるべきと考えます。そのため、他の同等システムなどのポリシーを見ながら対応を検討してください。

ウ　セキュリティ実装方針

モデルプランでは、脆弱性への対応策として提案者に『3. 1 Web アプリケーション脆弱性対応』に係る対処方針を示した「セキュリティ実装方針」の提出を求めることとしています。

モデルプランではその例として、開発規約の抜粋を提出することを例示しています。

エ　重要事項説明書

これは特記仕様書に定める要求内容を履行することができない可能性がある場合、又は提案事業者が独自に高度な回避策等の手法がある場合も想定し、「仕様書にある対策と同程度に脅威を削減できる代替案」であればその提出も可としています。すべて個別に代替策の妥当性を検討することになりますので、検討が可能なように詳細な提案を求める必要があります。

なお、モデルプランでは重要事項説明書による代替案の提示を認める項目は「団体が任意項目と明示した項目」と、『2. 選定ソフトウェアに関する保守性・運用性要件』『8. 保守要件』に限定しています。最終的にどの項目を任意項目と選定するかについては RFI などを通じて検討ください。

オ　納品物の指示

モデルプランでは、『5.テスト(検査)要件』に規定した検査を実施し、その結果である各種セキュ

リティ報告書類を納品する責務を課しています。

これは、自社スクラッチ開発の Web アプリケーションの場合、自社製品の検査となるため品質管理上の責務として、また、SIer がパッケージ Web アプリケーションを利用する場合は、パッケージメーカーによる脆弱性検査結果を当該パッケージの選定責任として提出することを期待したものです。いずれの場合も検査が不十分で脆弱性を内包して納品され、何らかの理由により運用時にこの存在が顕在化した際は、追加費用なしに修補することを求めています。

カ　調達に要する期間の設定

事業者側の検討期間は、通常、「物品に係る政府調達手続について（運用指針）[7]」により、競争入札における調達においては意見招請の期間を少なくとも 20 日間確保することとされており、入札公告の期間は少なくとも 40 日間確保することとされています。これを踏まえ、意見招請、入札公告の期間は、それぞれ 20 日、40 日に設定されることが多いのですが、セキュリティ上の要請、第三者ソフトウェアの取扱い検討にかかる期間の追加による影響を、システム規模等を勘案し、期間的に十分確保することを検討する必要があります。

(4)　業者決定

ここでは、業者決定の際の留意事項について記載します。

ア　仕様書の必須項目について

提案が仕様書の必須項目を満足していなければ、たとえ 1 項目であったとしても「失格」とし、採点しないのが原則です。また、代替案の提案があったとしても、発注者が不適当と認めるときは提案がなかったこととして失格となります。

イ　代替案の取り扱いについて

モデルプランで用意している『別紙 4-1 重要事項説明書』では、『仕様書にある対策と同程度に脅威を削減できる代替案　説明書』の提出を求める方式としています。

団体においては、提案事業者から代替案が提案された場合には、その妥当性を検討することになります。仕様に記載のものを超える優秀な代替案が提案された場合は別として、一般に、代替案が提案されたときは低く評価されます。この際、その内容が不適当だった場合は、「代替案の提案はなかった（要求仕様を充足しない）」とみなさなければならないことに注意が必要です。特記仕様書で示した内容を代替しないことが明らかな提案を受け入れてしまうと運用時、結果論的に何らかの事故が発生した場合に当該受注者との間でトラブルの元となります。

代替案を受け入れる場合は、どのタイミングで評価結果を伝えるのかも検討する必要があります（例：一般競争参加資格確認の通知に併せて結果を通知）。また、受け入れた代替案は契約に盛り込むことを忘れずに実施することが必要です。

逆にもし、代替案を受け入れず、代替案はなかったこととして不採用となった提案事業者には可能であれば透明性の観点からその理由を示すのが望ましいといえます。

---

7 http://www.kantei.go.jp/jp/kanbou/13tyoutatu/huzokusiryou/h1-3.html

なお、代替案の妥当性判断が困難な場合、有識者に相談したり、当センターのITアドバイザー制度を活用してください（当センターでは、当センターの正会員団体からの照会であれば提案内容について妥当か否か、アドバイスする制度[8]を有しています。ただし、代替案については各団体のセキュリティ委員会等、セキュリティに係る権限を有する者が最終意思決定することに変わりないことには留意する必要があります)。

また、重要事項説明書を提出する提案事業者においては、「重要事項説明書で提示する代替案の効用を自ら保証する義務」が発生してしまうこと、つまり、代替案の内容を団体が受け入れ、運用時に結果論的に何らかの事故が発生し、事後検証の結果として代替案が当初要求仕様に書かれていた内容に比して同程度以下のものであったことが判明した場合は、代替案の書類不備について責を負うことになることに留意いただく必要があります。

(5)　契約締結

ここでは、契約締結の際の留意事項について記載します。

業者選定を終えれば、自治法第 234 条第 5 項により契約書に記名押印し契約を締結します。契約書は標準約款だけでなく、仕様書、特記仕様書、質疑、提案書、合意内容を文書化したもの等、各文書類が契約書の一部を構成するという位置づけが明確となるように整理し、契約を締結することが肝要です。この文書類に含まれない、いわゆる紳士協定は契約ではないため注意が必要です。

特記仕様書の内容を受注者との間で有効にするためには例えば次のような契約方法が考えられます。

ア 特記仕様書の内容を覚書にし、取り交わす。

イ 特記仕様書を契約書の一部として含め、契約書には条文を追記する。

モデルプランでは、上記 イの方式をとった場合の追加条文例を『別紙７　契約書への追加条文例』を示しています。

また、特記仕様書以外の既存の他の特記仕様書（例：個人情報の取り扱いに関する特記仕様書）や、現在利用している標準契約書等における記載と、モデルプランから作成した特記仕様書における記載で矛盾が生じてしまう場合は、矛盾が生じないよう確認、調整する必要があります。例えば、重複する内容が書かれている場合、どの書類に記載された内容が優先事項となるか等を明記することが挙げられます。

なお、モデルプランの特記仕様書（雛型）では「本書が優先する」と記載しています。必要に応じて適宜修正してください。

(6)　開発開始・プロジェクト管理

ここでは、開発の開始からプロジェクト管理に係る留意事項を示します。

ア　プロジェクト管理

契約後の情報システム開発・導入は、受注者が中心に行うことから、発注者の主要な役割は、発注者としての品質管理と、進捗管理、プロジェクト管理となります。開発途中では定期的に品質を管理していくことにより、導入後の品質に対する信頼性の低下を防止することにもつながります。

[8] LASDEC 相談助言の実施　https://www.lasdec.or.jp/cms/10,1374,54.html

イ　開発時中間検査

モデルプランではオプション扱いとしていますが、実装上のセキュリティ向上策として、開発時中間検査という方法を示しました。これは開発途中における検査を通じて早い段階で簡易検査をサンプリング実施することで、脆弱性の芽を摘むことを目的としたものです。中間検査を実施する場合、発注者は受注者に対して工程表の提示を求め、あらかじめ適切なタイミングで中間検査の時期、検査対象画面と検査スケジュールを調整し、合意を得る必要があります。

ただし、中間検査については、そのシステムの規模等によっては有効に働かないことがあります。例えば、小さな Web アプリケーションシステムの場合、中間検査が実施できる頃にはほぼ最終検査に近い時もあります。

ウ　契約変更の取り扱い

開発の進展と具体化に伴い、軽微な仕様変更が発生することは通例ですが、その際、会議録等を作成、保存することは肝要です。業者選定の有力な根拠となった項目など、変更してはならない項目の取り扱いには特に留意する必要があります。

また、明らかにセキュリティレベル低下につながる恐れのある変更については、決定権者は最高情報統括責任者又は統括情報セキュリティ責任者の決裁を受けることが肝要です。契約金額の変更を伴わない場合もありますが、軽微な変更ではないことがあることに留意する必要があります。

(7)　納品・受入れ検査

ここでは、納品・受入れ検査の際の留意事項について記載します。

受入れ検査と検収を分けたのは、受入れ検査は各業務部門等の多くの検査員による部分検査が行われ、これらの部分検査がすべて合格したのちに、長の指名する検査員による検収という手続きを想定したためです。また受入れ検査は、手直し、検査が繰り返し行われることから、事務の煩雑さを回避するためでもあります。

ア　受入れ検査の時期

受入れ検査は、検収の前段の確認作業で、手直しと受入れ検査が反復します。受入れ検査に合格して初めて検収の段階に進むことになります。支払い遅延防止法では完成届から 10 日以内に検査を実施することになりますが、例えば、大規模なシステムの場合には 10 日以内に検査することは困難が予想されるほか、修正期間も必要となることから、本番稼動予定日の 2 カ月前を完成予定日として 1 カ月間を検査期間、その後の 1 カ月を修正期間と想定した工程表を作成することもあります。

大規模システムの場合は特に、あらかじめ十分な期間を確保した工程表を作成しておくことが肝要です。

イ　検査項目

受入れ検査は、要求仕様の機能要件と非機能要件の両面から行うこととなります。本書の対象とする Web アプリケーション検査項目でも契約書・仕様書等で定義した事項から検査項目を抽出し、

検査項目を確定します。この際『別紙 3 遵守状況一覧』を活用すると項目の整理に役立ちます。

Web アプリケーション脆弱性対策については、受注者から完成届に添付して提出される『別紙 6 実装状況報告書』と、受注者社内の検査報告書を参考としながら、それぞれの項目の受入れ検査手法を確定します。受注者の社内検査報告書をもって受入れ検査とすることもありますが、受入れ検査員が実操作を行うことも選択肢の 1 つです。どのような検査を行うかは基本的に発注者の自由ですが、例えば第三者による専門的なセキュリティ検査を行う場合はその費用などを発注段階で検討しておく必要があります。

受入れ検査の結果は受入れ検査調書として整理し、必要ならば受注者に手直しを指示します。

ウ　検査手法

一般に、受入れ検査といえば業務部門の職員が担当し、機能面の検査を行いますが、セキュリティ面の検査については知識を有する職員、例えば IT 部局の職員が担当できると理想的です。しかしながら、様々な事情からそのような専門知識を持つ職員が不足していることを加味し、モデルプランでは、Web アプリケーションの脆弱性検査について次の 3 つの手法を例示しています。

（ア） 書類審査

受注者の社内検査報告書により受入れ検査を行う。

検査手法や検査項目等は受注者の自由となりますが、あらかじめ発注者と受注者で協議し、その内容について合意を得ておきます。ただし、受注者からの報告書のみに依拠した検収では不十分との誹りがある可能性を考慮する必要があります。

（イ） 自ら検査する

受入れ検査職員が実操作等により検査を行う。

検査に用いることのできる仕様としては「ウェブ健康診断仕様（平成 22 年度版）[9]」などがあります。なお、当センターではウェブ健康診断仕様に沿った診断の実技を解説する講習会を団体向けに無償で実施していますのでご活用ください（平成 24 年度現在）。なお、（ウ）の場合と程度の差はありますが、自ら検査を実施する場合は受入れ検査員の技術力にその検査品質が左右される面があります。

（ウ） 第三者検査

第三者であるセキュリティ専門家（脆弱性検査の専門事業者）に検査を委託する。

発注者が指定する脆弱性検査の専門事業者が客観的な検査を行うので信頼性の高い検査結果を期待できます。ただし、検査費用を負担する必要があります。情報漏えいや改ざん等の事故を防止する観点では第三者検査が最も望ましいと言えます。

なお、当該検査事業者が要件に記載のない脆弱性を指摘することも考えられますが、それは修補対象にはあたらないため、発注者と受注者で協議の上、対応を決定することになります。この対応についてもあらかじめ協議しておく必要があります。

(8)　検収

受入れ検査調書が作成され、すべの項目が合格となれば検収に進みます。この段階の課題は契約上の

9 ウェブ健康診断仕様(平成 22 年度版)　https://www.lasdec.or.jp/cms/12,1284.html#siyou-h22

整理です。検査項目のすべてが合格であれば、検収合格となり合格証を交付して支払いへ進みますが、一部の項目に不合格が生じていたとき、契約の履行を求めて工期を延長するか等の意思決定を行うのが原則となります。

他の留意事項としては、提供した資料の返却、消去の確認を行うことが挙げられます。また、テストにおいては生データを使わないことが原則ですが、最終テストとして生データを利用したテストを行っていた場合、当該データが受注者に残らないよう、チェックが必要です。例えば、生データの貸し出しと消去を帳票にまとめておき、確実に消去したことを確認して書面に残す等の対応も考えられます。

(9)　運用

Webアプリケーションを運用する上では、Webアプリケーションパッケージソフトウェア、OS、ミドルウェア等における脆弱性修正パッチのリリースが発生したり、当センターの自動診断システムによる脆弱性診断事業[10]による診断や、独立行政法人情報処理推進機構(IPA)やセキュリティ研究者等の第三者によって当該サイトの脆弱性の指摘を受けることがあります。自身のシステムの脆弱性に関する情報収集に努めることは言うまでもありませんが、実際にパッチを適用したり、脆弱性を修補する際は、収集情報からリスクを適切に判断し、受注者に対して対応を指示する必要があります。

(10) 運用評価、見直し

セキュリティの見直しは、設定したセキュリティレベルが妥当であったかも含めて、評価時時点においても妥当であるかを検討することになります。具体的には、「新たな脅威が存在しないか」を見直す必要があります。

そして、見直しによって新たな脅威が見つかった場合は、回避策を含めて対策するか、その脅威を受容するか等を意思決定する必要があります。万一ですが、受容しがたい脅威が存在し、対策も事情により不可能である場合は、システムの稼働停止も視野に検討をするべきです。

また、SLA の遵守状況等運用状況のレビューによって新たな脅威に気づくこともあります。パッチ適用状況の随時の報告のほか、定期的にセキュリティの課題の棚卸をする機会を設け、残課題について課題を解決するか（パッチ適用するか等)、リスクを受容するか又は代替案を採用するかといった意思決定をまとめて確認することは、発注者、運用者（受注者）双方にとってメリットとなります。

10 自動診断システムによる脆弱性診断（セキュリティ健康診断）https://www.lasdec.or.jp/cms/12,22770.html

第4章 逐条解説

本章では、モデルプランにおける特記仕様書（雛型）で示された各要件について、その内容を解説するとともに、当該項目の必要性と当該項目をなくした場合のリスク等について判断材料となる説明を記載します。

なお、以降はモデルプランにおける特記仕様書（雛型）の章構成をそのまま転記しています。

1.　調達に関する基本事項

1.1.　本特記仕様書の目的と運用

本特記仕様書（以下「本書」という。）は、*本市（都道府県・区・町・村）*が導入する　*（システム名）*（以下「本システム」という。）のシステム調達仕様書（以下「仕様書」という。）に加え、本システムに追加で求めるセキュリティ要件、対応指針を記載するものである。受注者は本書に従わなくてはならない。

なお、本書に記載のないセキュリティ要求仕様に関しては仕様書による。契約書及び他の仕様書等の記載が本書と異なる場合は、本書が優先する。

1.2.　本システムのセキュリティ保証期間について

「セキュリティ保証期間」は次のとおりとし、期間中本書で定める対応を求める。

セキュリティ保証期間

〇年〇月〇日　より　〇年〇月〇日

＝解説＝

セキュリティ保証期間とは、同保証期間中に何らかの理由で発見した脆弱性については一部の例外を除き追加の費用なしに修補することを求める期間です。これは原則としてシステムの稼働予定期間と同じに設定します。契約締結後に団体において稼働予定期間を超えてシステムを継続利用することとなった場合は、セキュリティ保証期間の延長可否についてベンダーとの調整が必要です。後述しますが、特記仕様書に書かれた要求事項はこの期間内のみ有効であることを念頭に置く必要があります。特に『2. 選定ソフトウェアに関する保守性・運用性要件』『8. 保守要件』に係る内容についてはこの期間に密接に関わるため、稼働予定期間を超えるシステムの継続利用においてはセキュリティレベルの低下が発生しないよう、注意が必要です。

なお、本書ではこの期間を、5 年程度とすることが妥当と考えています。5 年と例示している理由については「8. 保守要件」の解説にて補足します。

1.3.　提案時の提出物

以下を記載した書類を提出すること。

(1)　遵守状況一覧

本書で定める要求仕様の遵守状況の概要を示したもの。『別紙3　遵守状況一覧』により提出すること。

＝解説＝

「遵守状況一覧」とは、RFPの各項目に対する準拠を一覧表にしたものです。別紙3は特記仕様書（雛型）の内容一覧です。

(2) セキュリティ実装方針

『3.1. Webアプリケーション脆弱性対応』で示した脆弱性がWebアプリケーションに混入しないように構築するための方針を示したもの。『3.2.セキュリティ実装方針の提出』に従い、提出すること。パッケージWebアプリケーション等自社開発でない場合はその開発元のものを提出すること。

＝解説＝

『別紙5 セキュリティ実装方針（サンプル)』で例示しています。セキュリティ実装方針は単に開発規約を提出させるためのものではなく、「記載の内容により、『別紙 1 脆弱性リスト』で示された脆弱性の対応を図る」ことを明記してもらうことが肝要です。開発規約の提示ではない、別の記載方法も考えられます。

(3)　重要事項説明書

本書で定める要求仕様の一部について、その履行が困難でかつ次のア、イにある各条件に合致する場合、代替案の提出を認める。代替案を提出する場合は『別紙4-1 重要事項説明書』を参考として任意の書式にて作成すること。

ア『2. 選定ソフトウェアに関する保守性・運用性要件』『8. 保守要件』に挙げた要件を満たすことが困難なソフトウェアを選定する場合

当該ソフトウェアについて次の項目を記載し、記載された代替案の内容の履行を保証すること。

□当該ソフトウェア名称及びメーカー名

□当該ソフトウェアの使用目的

□要件を満たすことができない項目名及びその理由

□満たすことができない要件の代替案及び代替案を採用する場合の費用見積(費用が必要な場合のみ)

イ 『3. Webアプリケーション脆弱性対応』『4. セキュリティ機能』に挙げた要件を満たすことが困難なソフトウェアを選定する場合

明示的に任意項目である旨の指定がある要件項目については、重要事項説明書による代替案を

提出してよい。ただし、それ以外の項目は全て必須項目であり、代替案の提出を認めない。

＝解説＝

「重要事項説明書」とは、各項目に対する準拠ができない理由、回避策、代替策などを事前に説明した文書です。モデルプランでは『別紙 4-1　重要事項説明書』にてそのフォーマット案を示しています。同案では、『仕様書にある対策と同程度に脅威を削減できる代替案　説明書』の提出を求める方式としています。重要事項説明書の受け入れについては、『第 3 章 調達プロセスにおける帳票（ツール）の利用場面…(4)』も参照してください。

ほか、本件に関連する留意点としては、セキュリティ保証期間内のパッチ提供の保証ができないソフトウェアを何らかの事情により選定せざるを得ない場合が考えられます。例えば次のような状況が想定されます。

- 採用したい（しなければならない）製品のサポートライフサイクルが明確でない、あるいは短い場合
- オープンソースソフトウェアの採用を予定しており、将来のパッチ提供の約束が得られない場合
- Web サイトのセキュリティ保証期間内にソフトウェアのメジャーバージョンアップが予定されており、コンピュータのメモリ増設やリプレースが必要になる場合

万一パッチがメーカーから提供されない場合は、代替ソフトウェアへのリプレースや、言語のメジャーバージョンアップなどによる回避策及び想定される費用の試算を提案者（受注者）に求めています。ハードウェア増強が想定されるケースでは、ハードウェア増強の費用も含めさせる必要があります。導入当初からこのような予算上の制約による脆弱性への対応の遅延が出かねない事情を把握しておくことで、備えることができます。

モデルプランではこのようにソフトウェアの選定に当たっては留意点が多々あることを示し、サポートライフサイクルが明確なソフトウェアを出来るだけ採用することを推奨しています。

また、「重要事項説明書」にて発注者と受注者が合意した内容は、受注者にとっての免責条件となります。重要事項説明書に記載された想定追加費用は、見積もり費用に加算し、想定総額費用にてベンダー選定を行う必要があります。

なお、代替案として認めるか否かの権限は発注者が有しています。代替案の提案を認めない項目についてはあらかじめ明示する必要があります。モデルプランでは特に断りのない限りすべて必須項目である旨をここで記載していますが、適宜ご検討ください。

2．選定ソフトウェアに関する保守性・運用性要件

システムのプラットフォームソフトウェア（OS、ミドルウェア、ソフトウェア部品（ライブラリ等）を指す。）及びWebアプリケーションソフトウェア（パッケージWebアプリケーションを含む。以下「Webアプリケーション」という。）の選定にあたっては次の要件を満たすこと。また、一部要件を満たせない場合の提案方法については要件の記述に従うこと。

(1) 選定するプラットフォームソフトウェア及びWebアプリケーションのメーカーにおけるサポートライフサイクルポリシーにおいて、当該ソフトウェアのメーカーから*本市（都道府県・区・町・村）*に対して、セキュリティ保証期間の全期間中、脆弱性修正パッチ（以下「パッチ」という。）の開発及び提供がされることが確認できること。

(2) 当該ソフトウェアメーカーのサポートライフサイクルポリシーから判断して、セキュリティ保証期間の全期間を満たす形でパッチの開発及び提供がされない可能性があるソフトウェアを提案する場合『1.3．提案時の提出物（3）』に従って資料を作成し、提出すること。

(3) 納品前の適切な時期に、*本市（都道府県・区・町・村）*と別途協議の上、納品時のパッチ適用状態を定める。定めたパッチは全て適用した状態で納品できること。

(4) プラットフォームソフトウェア及びWebアプリケーションは、当該ソフトウェアで新たに発見された脆弱性に関する情報やパッチのリリース情報（以下「パッチ情報」という。）がインターネットに遅滞なく公開されているものを選定すること。

(5) 当該ソフトウェアに係るパッチ情報がインターネットに公開されない場合、パッチ情報を*本市（都道府県・区・町・村）*に提供するための代替手段について『1.3．提案時の提出物（3）』に従って資料を作成し、提出すること。

＝解説＝

セキュリティ保証期間内のセキュリティレベルを保つために、必要なことについて言及しています。端的に言うと次の事項がソフトウェアを選定（開発）する上で必要となります。

- 当該期間中はメーカーからパッチの提供がサポートライフサイクルポリシー等により明示されていること
- パッチ適用が可能なシステム導入状態であること
- パッチを提供する体制があること

なお、「セキュリティ“保証”期間」の「保証」という言葉についてですが、モデルプランでは上記のようにシステムのサポートライフサイクルや提供体制の確認をセキュリティ保証期間中継続することをもって「保証」という表現をしています。

また、Webアプリケーションの場合、パッケージWebアプリケーションをカスタマイズして導入することがある点がOS、ミドルウェア等とは異なります。カスタマイズした箇所の脆弱性に対しては、パッケージWebアプリケーションのパッチとしては提供されません。また、カスタマイズしてあることによりパッチが適用できないこともありますので、カスタマイズをする場合はそのカスタマイズ部分に関する処遇もパッケージWebアプリケーションと同等にしておく必要があります。

3. Webアプリケーション脆弱性対応

本システムにおける Web アプリケーションの脆弱性対応として次の要件を満たすこと。

3.1. Webアプリケーション脆弱性対応

『別紙 1 脆弱性リスト』で示す脆弱性が本システムに混入しないよう Web アプリケーションを構築すること。

＝解説＝

脆弱性の検出手法の代表的なものとして、ソースコードを知らない状況でのブラックボックス検査が挙げられますが、専門家であっても全ての脆弱性を網羅的に発見することは困難です。

また、仮にソースコードが検査できる状況でのホワイトボックス検査を専門家が実施したとしても、網羅的に内在する全ての脆弱性を検出することは困難です。

そのため、モデルプランでは Web アプリケーションに内在する全ての脆弱性を瑕疵とする過大な責任を受注者に負わせるものではなく、『別紙 1 脆弱性リスト』に挙げた、現在の技術水準で十分事前に解決可能な（解決方法が容易に入手・参照でき、検査方法もある）脆弱性のみ、対応を求める対象として限定しました。

そして、検収・納品後に『別紙 1 脆弱性リスト』に挙げる脆弱性が第三者からの指摘等により発見された場合は、当該脆弱性は仕様を満たしていない「隠れた瑕疵」に相当するものとし、被害を未然に防ぐため、追加費用なしにこの修補を受注者に求めることが適当としています。

このような取り決めをモデルプランで定めた理由は、団体を含め、一般に Web アプリケーションの発注者は、受注者（提案者）に比してセキュリティに関する知見が不足するのは致し方ないためです（情報の非対称性）。

そのため、モデルプランでは受注者の納入する Web アプリケーションが引き起こす被害発生の可能性有無、すなわち、そのリスク評価、リスク判断の一部を受注者に委ねています。具体的には、発注者はセキュリティ実装方針の提案を求め、提案中で提案者に、脆弱性がないように構築することを約束（リスク評価、リスク判断）してもらい、約束どおりに実装したことの報告を受けるという流れを基本としています。

ただし、第三者からの指摘等により発見された脆弱性とされる現象が「隠れた瑕疵（『別紙 1 脆弱性リスト』にて定義された脆弱性がある状態）」なのか、「隠れた瑕疵ではない（『別紙 1 脆弱性リスト』に定義された脆弱性ではない）」かの判断で発注者及び第三者と、受注者の間で意見に相違が生ずる場合が考えられます。その際は、有識者に相談したり、当センターまでご相談ください。例えばそのような状況は、クロスサイト・スクリプティング脆弱性等において発生する可能性があります。

なお、『別紙 1 脆弱性リスト』に挙げていない脆弱性[11]や、発注者が追加提案して対処を約束した脆弱性ではない脆弱性を第三者等が発見した場合についてはこれを瑕疵とせず、受注者・発注者双方協議の上、当該脆弱性を修補するか、リスクを受容するか、代替案を選択するかの最終判断を発注者がするのが妥当と考えます（この場合、保守対応若しくは別途費用により解決します）。

11 現在の技術水準では対策できないものや、Web ブラウザ側の問題、脆弱性か否かの判断がセキュリティ専門家の間でも難しく、当該現象に対して名称も付与されていないものがある。

なお、各脆弱性の定義や、一般的な対策方法の解説については、本書のほか、『別紙 1 脆弱性リスト』にある脆弱性名称の定義に関する参照先等を参照してください。

以下、『別紙 1 脆弱性リスト』で挙げている各脆弱性について概要を解説します。

(1) SQL インジェクション

SQL インジェクションは、SQL の呼び出し方に不備がある場合に発生する脆弱性です。SQL インジェクションがある場合、以下のような影響を受ける可能性があります。

・データベース内のすべての情報が外部から盗まれる
・データベースの内容が書き換えられる
・認証を回避される

対策としては、以下のいずれかを実施します。

・ プレースホルダにより SQL 文を組み立てる
・ アプリケーション側で SQL 文を組み立てる際に、リテラルを正しく構成するなど、SQL 文が変更されないようにする。

(2) OS コマンド・インジェクション

Web アプリケーションの開発では、様々な言語でシェルから OS コマンドを呼び出す機能を提供しています。問題のある実装の場合、開発者の意図に反して、OS コマンドが外部から実行可能な状態に置かれることがあります。これを OS コマンド・インジェクションと呼びます。OS コマンド・インジェクションは OS の提供する様々な機能を実行できるため、攻撃者の思うままに、Web サイトを操ることができてしまうため極めて危険な脆弱性です。

対策としては、一般に OS コマンド呼び出しを使わない実装が最も確実です。どうしても必要な場合は、シェル呼び出し機能のある関数の利用を避けたり、外部から入力された文字列をそのままコマンドラインのパラメータに渡さない等が考えられます。

(3) ディレクトリ・トラバーサル脆弱性

外部からパラメータの形でサーバ上のファイル名を指定できる Web アプリケーションでは、ファイル名に対するチェックが不十分な場合、アプリケーションの意図しないファイルに対して閲覧や改ざん、削除ができる場合があります。この脆弱性をディレクトリ・トラバーサル脆弱性と呼びます。ディレクトリ・トラバーサルによる影響は以下のとおりです。

・Web サーバ内のファイルの閲覧
・Web サーバ内のファイルの改ざん、削除
・ファイルの改ざんが可能な場合、不正なスクリプトの実行

スクリプト言語にはincludeあるいはrequireなどの機能により、外部のソースを動的に取り込めるものがあります。このinclude機能のファイル指定部分にディレクトリ・トラバーサル脆弱性がある場合は、命令を不正に実行するという攻撃が可能となる場合があります（ファイルインクルード攻撃、CWE-98[12]）。

また、PHP は設定によってはファイル名の代わりに URL を指定して外部からファイルを取り込むことが

12 http://cwe.mitre.org/data/definitions/98.html

でき、その場合、ファイルとして取り込む内容を自由に指定できてしまいます。これをファイルインクルード攻撃に組み合わせた場合、任意の命令を外部から実行することができてしまいます（リモートファイルインクルード攻撃(RFI)）。

対策は以下のいずれかを実施します。

・外部からファイル名を指定できる仕様を避ける
・ファイル名にディレクトリ名が含まれないようにする
・ファイル名を英数字に限定する

(4) ログイン機能の不備

①推測可能なセッション ID

Web アプリケーションに用いられるセッション ID の生成規則に問題があると、利用者のセッション ID が推測され、セッションハイジャックに悪用される可能性があります。

一般に、現実的で効果的な対策としては、Web アプリケーション開発ツールが備えるセッション管理機構を利用することが挙げられます。

② URL 埋め込みのセッション ID の外部への漏えい

セッション ID を URL に埋め込んでいると、Referer ヘッダを経由して、セッション ID が外部に漏えいし、なりすましの原因になる場合があります。対策としてはクッキーにセッション ID を保存することが一般的です。

ただし、携帯電話向けの Web サイト（i モード、EZweb、Yahoo!ケータイ等、従来型携帯電話向けサイト)の場合は一部機種がクッキーに対応していないため､セッション ID を URL に埋め込むほかなく、この場合については差し支えありません。ただし、外部ドメインのサイトに遷移する際に、リンク先に Referer としてセッション ID が漏えいするため、外部サイトに直接リンクすることは避け、リンク先との間にセッション ID の付与されないページ（クッションページ等と呼ぶ）を挟むことで Referer 経由によるセッション ID 漏えい問題が解決できます。

③クッキーのセキュア属性不備

クッキーにはセッション ID などセキュリティ上重要な情報が格納されていることが多いため、クッキーが盗聴されるとなりすましの被害に直結しやすいと言えます。クッキーにセキュア属性が付与されていない場合、平文通信(HTTP)でリクエストが送信された場合にクッキーも共に送信されてしまうので、当該クッキーが盗聴される危険性が生じます。

対策としては以下のいずれかを実施します。

・クッキーにセキュア属性をつける。
・セッション ID とは別にセキュア属性つきのクッキーとしてトークンを発行し、ページごとにトークンを確認する。

④ セッション ID の固定化

セッション ID の固定化とは、ログイン後にセッション ID が変化しない脆弱性のことです。利用者が第三者（攻撃者）からセッション ID を強制された後に利用者がログイン操作を行うと、ログイン状態のセッション ID を第三者が知り得ることになり、なりすましされる危険性が生じます。

対策としては、以下のいずれかを実施します。

・ログイン後にセッションを開始する

・ログイン前にセッションを開始している場合、ログイン後にセッション ID を変更する

・ログイン成功後にトークン（秘密情報）を発行し、ページ毎にトークンの値を確認する

(5) クロスサイト・スクリプティング(XSS)

クロスサイト・スクリプティング（XSS）は、Web アプリケーションにスクリプトを埋め込むことが可能で、利用者のブラウザ上で不正なスクリプトが実行されてしまう脆弱性です。

一般に対策としては以下を実施します。

・動的に表示する項目のエスケープ処理

・HTML タグの属性値をダブルクォートで囲む

・HTTP レスポンスヘッダに文字エンコーディングを指定する

XSS は、一般に HTML や JavaScript ファイルの表示時に脆弱性が発現しますが、画像や PDF などのファイルをダウンロードする際にも、XSS が問題になる場合があります。悪意の利用者がアップロードした画像や PDF などをダウンロードする際、当該データに HTML タグや JavaScript を含めておくと、ブラウザが HTML ファイルと誤認し、JavaScript を実行してしまうことがあります。

対策としては、以下が有効ですが、ブラウザに依存する部分もあるため利用を想定するブラウザで有効な対策を施す必要があります。

・ファイルの Contents-Type を正しく設定する

・HTTP レスポンスヘッダに、X-Content-Type-Options: nosniff を指定する

・ダウンロードを想定したファイルにはレスポンスヘッダとして「Content-Disposition:attachment」を指定する

(6) 利用者の意図に反した実行の防止機能の不備

① クロスサイト・リクエスト・フォージェリ (CSRF)

CSRF は、悪意のある者により、利用者が予期しない処理を実行させられてしまう脆弱性です。一般に CSRF 脆弱性がアプリケーションに存在する場合、以下のような影響を受ける可能性があります。

・利用者のアカウントによる物品の購入

・利用者の退会処理

・利用者のアカウントによる掲示板への書き込み

一般に対策としては以下を実施します。

・CSRF 対策の必要なページを区別する

・正規利用者の意図したリクエストを区別できるように呼び出し側で秘密情報（トークン）を埋め込み、処理のページでチェックする

② クリックジャッキング

クリックジャッキングとは、CSS により iframe を透明化することで利用者を視覚的にだまし、意図しない動作に誘導する手法のことです。この手法を攻撃者が悪用することで、悪意のあるサイトに誘導された利用者が不正操作を引き起こすクリックをさせられる可能性があります。

対策としては、以下の HTTP レスポンスヘッダを送信します。

X-FRAME-OPTIONS: SAMEORIGIN　あるいは　X-FRAME-OPTIONS: DENY

(7) メールヘッダ・インジェクション脆弱性

メールヘッダ・インジェクションは、宛先（To）や件名（Subject）などのメールヘッダを外部から指定する際に、改行文字を使ってメールヘッダや本文を追加・変更する手法です。一般にメールヘッダ・インジェクション脆弱性による影響は次のようなものがあります。

・件名や送信元、本文を改変される
・迷惑メールの送信に悪用される
・ウイルスメールの送信に悪用される

一般に対策としては以下のいずれかを実施します。

・外部からのパラメータをメールヘッダに含ませないようにする
・外部からのパラメータには改行を含まないようにチェックする

(8) 「アクセス制御」と「認可処理」の不備

①アクセス制御の不備

『4.1.2. アクセス制御機能』にてアクセス制御機能を要求しています。同項の実装に不備や欠落があり、ログインしていない利用者が、ログインの必要な画面の閲覧ができる場合をアクセス制御の不備と言います。

②認可処理の不備

『4.2. 認可処理』にて認可処理を要求しています。同項の実装に不備や欠落があり、権限のない利用者が画面を表示できたり、権限のない機能を実行できる状態を認可処理の不備と言います。

(9) HTTP ヘッダ・インジェクション

HTTP ヘッダ・インジェクション脆弱性は、リダイレクトやクッキー発行など、外部からのパラメータを元に HTTP レスポンスヘッダを出力する際に発生する脆弱性です。

Web アプリケーションに HTTP ヘッダ・インジェクション脆弱性があると、一般に以下の影響があります。

・任意のクッキーの生成
・任意の URL へのリダイレクト
・表示内容の改変
・任意の JavaScript 実行による XSS と同様の被害

一般に対策としては以下を実施します。

・HTTP ヘッダ・インジェクション対策のされたライブラリにより HTTP レスポンスヘッダを出力する
・HTTP レスポンスヘッダとして出力するパラメータから改行文字を削除する

(10) eval インジェクション

多くのスクリプト言語は与えた文字列をスクリプトのソースとして解釈して実行する機能（eval 関数）があります。同機能の実装に問題がある場合、eval インジェクション攻撃ができる場合があります。なお、同攻撃の影響は、OS コマンド・インジェクションと同様です。

一般に対策としては以下を実施します。

・eval 関数を使わない
・引数に外部からのパラメータを含めない（含める場合は英数字に限定する）

(11) 競合状態の脆弱性

複数の要求を同時に受け付けることが多い Web アプリケーションでは共有資源の取扱に関する問題が発生することがあります。競合状態の脆弱性は Web サーバのコンピュータ資源において複数のプロセス、スレッドから同時に利用している変数、共有メモリ、ファイル、データベース等の共有資源に対する排他制御が不十分な場合に発生します。特に有名な現象としては、別人問題が挙げられます。別人問題は、競合状態において、異なる利用者の情報が画面に表示されてしまうという問題です。

また、他にもデータベースの不整合、ファイル内容の破損等の影響が出ることがあります。

一般に対策としては以下を実施します。

・共有資源に対する適切な排他制御
・可能であれば、共有資源の利用をできるだけ避ける（脅威の回避）

(12) 意図しないファイル公開

意図しないファイル公開とは、本来は公開するのが適当ではない、個人情報等秘密情報を記載したファイルが Web サーバの公開ディレクトリに配置されてしまっている場合が例として挙げられます。URL がわかればインターネット経由で当該ファイルを閲覧が可能となるものです。

主に Web サーバの設定における、ディレクトリ・リスティング機能が有効になっているところから発覚することが多く、2004 年以前に発生している情報漏えい事件・事故の多数がこのパターンによるものでした。

意図しないファイル公開は公開ディレクトリにファイルが置かれ、アクセス制限がなく、URL が知り得る状態の場合に発生する脆弱性です。なお、URL を知り得る状況は以下のようなパターンが考えられます。

・ファイル名が類推可能なもの（日付や氏名、連番、ありがちな名前（user.txt 等）
・エラーメッセージ、他の脆弱性によりファイル名がわかる
・外部サイトからリンクされ、検索エンジンに登録される

一般に対策としては、非公開情報を公開ディレクトリに置かないことが挙げられます。

(13) アップロードファイルによるサーバ側スクリプト実行

Web アプリケーションには、Web サイトを経由して様々なファイルをアップロードする機能を提供する

ものがあります。

団体の場合、「地方公共団体における情報セキュリティポリシーに関するガイドライン」において、Webで利用できるフリーメール、ネットワークストレージサービス等の利用は禁止とされており、各団体が事業者と大容量のデータをやり取りする際等に、自前のファイルアップロードサイトを構築するケースがあります。

一般のファイルアップローダの中には、アップロードしたファイルを例えば庁内 Web サイトの公開ディレクトリに保存するものがあり、ファイル名の拡張子が、「.php」「.asp」「.aspx」「.jsp」等の場合、当該ファイルをスクリプトとして実行することができてしまいます。そのため、これが悪意あるスクリプトだった場合、OS コマンド・インジェクションと同様の影響をもたらすことがあります。

一般に対策としては以下が考えられます。

・アップロードされたファイルは公開サイトではなく、スクリプト経由で閲覧させる

・アップロードできるファイル拡張子を制限する

(14) 秘密情報表示時のキャッシュ不停止

Web ブラウザには、表示速度を上げるためにキャッシュ機能が実装されているほか、表示速度を上げる等のためにネットワーク上にもプロキシサーバによってコンテンツのキャッシュがなされます。

Web サイト側で秘密情報を表示の際は、キャッシュ機能を停止する命令を HTTP レスポンスヘッダとして出力する必要があります。これをしていない場合、表示された秘密情報がプロキシサーバにキャッシュされ、同じプロキシサーバを使用する別人が同じ画面を閲覧した際に、キャッシュされた情報を閲覧する結果となることがあります。

(15) オープンリダイレクタ脆弱性（意図しないリダイレクト）

Web アプリケーションの中には、パラメータにより指定した URL にリダイレクトできる機能を備えるものがあります。このリダイレクト機能のことをリダイレクタと呼び、リダイレクタの中で任意のドメインにリダイレクトできるものをオープンリダイレクタと呼びます。オープンリダイレクタが可能な場合、以下のような影響が出る可能性があります。

・フィッシングサイトに誘導され、重要情報を入力させられる

・デバイスドライバやパッチと称してマルウェアを配布される

一般に対策としては以下のいずれかを実施することが考えられます。

・リダイレクト先の URL を固定にする

・リダイレクト先の URL を直接指定せず番号指定にする

・リダイレクト先のドメイン名をチェックする

(16) クローラへの耐性

クローラとは、インターネットの検索エンジンのインデックス作成のために各 Web サイトを巡回する自動プログラムです。Web アプリケーションによっては、クローラからのアクセスのような低負荷にすら耐えられない欠陥を持つものが発見されており、状況によっては Web サーバが停止する不具合が生じます。

本件の詳細及び検査方法については、LASDEC ウェブ健康診断仕様平成 22 年度版の『2.5 (M)クローラへの耐性について』を参照してください。

3.2.　セキュリティ実装方針の提出

『3.1. Web アプリケーション脆弱性対応』で示した脆弱性が Web アプリケーションに混入しないように構築するための方針を『別紙 5 セキュリティ実装方針(サンプル)』を参考として任意の書式で作成し、提出すること。なお、『別紙 1 脆弱性リスト』に記載されている項目に関して漏らさず記載すること。

＝解説＝

『3.1. Web アプリケーション脆弱性対応』に要求する脆弱性対応を提案者が実現できる根拠として、「セキュリティ実装方針」の提出を求めています。これは提案のために新たに作成することを求めているものではなく、応募者が元々使用している「セキュア開発ガイドライン」や「開発標準」のセキュリティに関する章を抜き出して提出することを想定しています。ただし、これらが『3.1. Web アプリケーション脆弱性対応』に対して対応漏れの項目がある場合、提案者は対応可否を検討の上、必要に応じて内容を追記することを期待しています。ただし、ここではガイドラインや開発標準等によらない提示方法を否定していません（他の方法で示せるのであれば他の方法でも構わない)。

「セキュリティ実装方針」を提出する狙いは次の（1)、(2）のとおりです。

(1)　提案者（受注者）が『3.1. Web アプリケーション脆弱性対応』の要件を実現する能力を有していることを確認する

(2)　「セキュリティ実装方針」を仕様書の一部とすることで、セキュリティ仕様の明確化を図る

納品された Web アプリケーションがセキュリティ実装方針に従っていない場合は、受注者が約束した仕様と相違することを理由に追加費用なしで修補を求めます。

一方、セキュリティ実装方針に不備があるために、セキュリティ実装方針には従っているが脆弱性が混入してしまった場合は、「『3.1 Web アプリケーション脆弱性対応』を満たしていない」という理由で瑕疵となる可能性があります（遅滞なく修補する場合は除く)。

4. セキュリティ機能

本システムにおけるセキュリティ機能は、次の仕様要件を満たすこと。

＝解説＝

この章で定義するセキュリティ機能は、プログラムのミス（バグ）による脆弱性とは異なり、セキュリティ上の機能を満たすために必要な実装を示しています（実装していないことがそのまま脆弱性となる)。なお、この項はあくまで例示であり、ここに記載の内容で十分という訳ではありません。各システムの特性に応じて必要なセキュリティ機能の追加提案を求めてください。

また、一部の機能要件については、実施するとなお良いが、実施しない場合でも最低限の対策とするには支障がない項目があります。それらについては解説の冒頭に、オプション提案である旨の記載をしています。オプション提案を採用すると、費用がかかったり、運用上の負担がかかることがあります（セキュリティレベル向上とのトレードオフ)。

また、Webアプリケーションの特性によっては、一部の機能要件で不要なものがあります。例えば、ID、パスワードによる利用者のログイン処理を必要としないWebサイトの検討をする場合は、次節の『4.1 ログイン処理』は全て不要になります。そのような場合、不要な機能の節を削除してください。

4.1.　ログイン処理

＝解説＝

この節はログイン処理についての要求です。ログイン処理内容として、利用者認証とアクセス制御を含みます。なお、ログイン処理を必要としない Web アプリケーションの場合は本節を消去していただくことを想定しています。他の各セキュリティ機能についても同様のため、このような記述は以降割愛します。

4.1.1. 利用者認証方式

利用者の認証方式はパスワード認証とする。

＝解説＝

認証を利用する場合、認証方式にはいくつかの方法があります。Webアプリケーションの特性に応じて、認証方式を選択してください。ここでは典型的なパスワード認証を例示しています。

パスワード認証は利用者 ID など利用者が個別に持つ ID と、利用者のみが知っているパスワードを照合することにより本人を識別する方法です。

4.1.2. アクセス制御機能

(1) 利用者の認証を行い、認証した利用者のみが本システムの「利用者認証を要する機能(画面)」を利用できるようにすること。
(2) 利用者認証を経ていない者は本システムの「利用者認証を要する機能(画面)」を利用できないようにすること。
(3) 「利用者認証を要する機能(画面)」は、セッションが終了した後は利用できないこと。
(4) 「利用者認証を要する機能(画面)」について、『5.3. 最終提出物 (1)』で示す画面遷移図に識別マーク等を使って示し、その通りに(1)、(2)の機能を実装すること。

＝解説＝

ログイン処理として利用者認証を求める目的は、認証できた利用者のみに特定画面を閲覧させる又は特定機能を利用させることです。本項はどの画面について認証を要求するかを画面遷移図上で識別できるようにするとともに、その通りに実装することを求めています。

4.1.3. パスワードに利用できる文字

パスワードに利用する文字は以下を遵守すること。ただし、二要素認証の第 2 要素（ワンタイムパスワードトークンの生成するパスワードなど）はこの限りでない。

(1) パスワードに利用できる文字種は、英字（大文字、小文字を区別）、数字、記号の 3 種とし、それぞれ自由に利用できること。

(2) パスワードに利用する文字数は 8 文字未満を受け付けないようにすること。また、少なくとも 64 文字のパスワードは受け入れられること。

＝解説＝

認証方式としてパスワードを選択した場合、第三者が利用者になりすますことができないよう、予測困難なパスワードが設定できる必要があります。安全なパスワードを決める責任は利用者にありますが、Web アプリケーション側として利用者が安全なパスワードを付けることを邪魔しないために満たすべき、2 つの要件があります。

まず、文字種ですが、半角の英字（大文字、小文字を区別）、数字、記号が望ましく、最低限英字と数字の両方が使用できることが望ましい対応です。

次に、パスワードの長さは 8 文字以上を必須とし、文字数に上限を設ける場合 64 文字以上確保することが望ましいといえます。

例えば「数字のみ（10 種）、4 ケタのパスワード」を設定可能な場合、パスワードの組み合わせ総数は 1 万個です。これを「英数記号（94 種）、8 ケタのパスワード」にすると、約 6,096 兆個となります。つまり文字種と文字数を増やせば、パスワードの組み合わせの総数は増加し、利用者が予測困難なパスワードを決めやすくなります。

4.1.4. ログインフォームの実装方法

パスワードの入力欄は入力した文字を伏字にする（input 要素において type 属性の値に password を指定（type=”password”）する）こと。又は、伏字にする・しないを選択できる機能を持つこと。

＝解説＝

ログインフォームに対する要件はパスワード入力欄をマスク表示することです。

パスワードのマスク表示とは、例えば、画面上で「*******」と表示されるようにし、入力したパスワードが画面上に表示されないようにすることです。パスワードをマスク表示させることで、ショルダーハックなどによってパスワードを盗み見されるリスクが低減します。

また、長いパスワード文字の入力をしやすくするため、利用者側に伏字機能をオン・オフ選択させるという実装もあります。言うまでもなく、オフにする場合は周囲に人がいないことを確認してパスワード入力する必要があります。

4.1.5. ログイン失敗時のメッセージ出力

パスワード認証に失敗した際に、利用者 ID の間違いか、パスワードの間違いかが区別できるメッセージを表示しないこと。

＝解説＝

ログイン機能が表示するエラーメッセージでは、パスワード認証に失敗した場合、利用者 ID（ログイン ID）の間違いかパスワードの間違いかを区別できるメッセージを表示しないことが重要です。例えば「指定したユーザは存在しません。」「パスワードが間違っています。」といったメッセージを表示するのは不適当です。

なぜなら、利用者 ID とパスワードのどちらに間違いがあるかがわかると、まず利用者 ID について探索をし、存在する利用者 ID があれば、次にパスワードを探索するといった手法によるパスワード総当たり攻撃が行いやすくなるためです。一方、利用者 ID とパスワードのどちらが間違っているかわからない場合は、全ての ID とパスワードを組み合わせる必要があり、これは膨大な計算量となるため、パスワード総当たり攻撃が成功しにくくなります。

4.1.6. アカウントロック機能

パスワードを連続して 10 回間違った場合は、当該アカウントを 30 分間ロックすること。

＝解説＝

アカウントロックとは、特定のアカウントでパスワードを一定回数間違った場合にそのアカウントを無効化することです。

アカウントロックを行わず、何度ログインに失敗しても何ら制限を設けていない場合、総当たり攻撃が行によって、ログインが成功してしまう可能性があります。アカウントロックの身近な例としては、ATM で暗証番号を 3 回間違えた場合にキャッシュカードが使えなくなるというルールが挙げられるとおり、一定回数間違った際にアカウントを無効化することはパスワード総当たり攻撃の有効な対抗策です。

アカウントロックの要件には次の 2 つポイントがあります。

まず、ログイン失敗の回数とアカウントロック時間を利便性が損なわれない値に設定することです。ここでは、パスワードを 10 回間違った場合は、当該アカウントを 30 分間ロックすることとしていますが、あくまでも例示です。ログイン失敗の回数、アカウントロック時間はサービスの特性を考慮して決定してください。ただし、厳しい設定をする際は利用者の利便性低下とトレードオフの関係にあることに注意が必要です。ATM のように 3 回失敗でアカウントロックするなど、失敗の回数を少なくしすぎると、正規利用者がロックされる頻度が高まってしまうことが懸念されます。また、ロックする時間が長い場合、アカウントロック中は正規利用者も利用できないため、締め出される可能性が高くなります。

なお、これは特に要件には入れていませんが、「管理者によりアカウントロックを解除できるようにすること」が追加の要件として考えられます。管理者がアカウントロックを解除できるようにしておくことで、利用者からの問い合わせに基づき、何らかの方法による適切な本人確認ができた際に再有効化できるようにすれば利便性向上が期待できます。ロック時間を長くしたい場合はこのような機能も取り入れることを検討してください。

4.1.7. オフライン攻撃からのパスワード保護

(1) パスワードは平文で保存せず、ソルトつきハッシュの形で保存すること。

(2) ソルトは利用者毎に別々に設定すること。

(3) ソルトは最低 5 文字以上とること。

＝解説＝

利用者のパスワードが外部に漏えいした場合、第三者が利用者になりすますことにより、本来利用者のみが閲覧可能な情報が閲覧されたり、機能が実行される等により、利用者が被害を受けることになります。そのため、万一サーバからパスワードが漏えいしても悪用されないように、パスワードを安全に保存しておく必要があります。

パスワードを安全に保存するには、平文で保存せず、ソルトつきハッシュで保存する方法が知られています。平文とは、暗号化されていないデータのことです。

ソルトとはパスワードに追加する文字列のことです。見かけ上のパスワードが長くなる上、利用者毎に異なるソルトを設定することで、パスワードが同じでも異なるハッシュ値が生成されます。平文保存していたパスワードが漏えいすると、漏えいしたパスワードがそのまま利用できてしまうため、すぐに被害につながる可能性があります。一方、ソルトを利用せずハッシュのみの場合は、レインボーテーブルと呼ばれるテクニックによってハッシュ値を解析し、平文を得ることができてしまうことが知られています。

このことから、パスワードはソルトつきハッシュで保存するとよいとされます。パスワード 8 文字以上とソルト 5 文字以上を合わせて 13 文字以上のパスワード文字列にすることで、現時点でのレインボーテーブル対策には有効です。なお、ソルトの文字列は 5 文字にこだわる必要はなく、長くしても結構です（より強固です)。

またソルトつきハッシュに加え､｢ストレッチング｣を施すことも､総当たり攻撃への対策になります。ストレッチングはハッシュ計算を繰り返し行い、計算時間を延ばすことで総当たり攻撃を受けた際のハッシュ計算の速度を遅くすることができるものです。なお、ストレッチングもオフライン攻撃に対するパスワード保護として有効ですが、ここでは、アプリケーションが対処すべき必要最低限の脅威を「レインボーテーブルによるハッシュ値解析」と位置づけており、レインボーテーブル対策にはソルト付きハッシュの採用で十分有効であることから、ストレッチングについては紹介に留めます。

4.1.8. セッション管理機能

(1) 利用者のセッション管理にはプログラミング言語や Web アプリケーション実行環境の備えるセッション管理機構を用いること。

(2) ログイン状態にある利用者のセッション識別のための情報（セッション ID）は、クッキーを用いて保持すること。

＝解説＝

ログイン情報は、利用者も含め、外部から閲覧・書き換えできないことを求めています。その標準的

な手法としてセッション管理機能を用いることができます。セッション管理機能とは、利用者ごとにセッション識別のための符号（セッション ID）を発行して、そのセッション ID をクッキーに保持することです。クッキー以外に保持する場所として URL 埋め込みも用いられていますが、URL 埋め込みのセッション ID は Referer 情報から外部サイトに漏えいする可能性があるため、クッキーにセッション ID を保持するよう要求しています。

4.1.9. セッションの開始

セッションはログイン処理成功後に開始すること。

＝解説＝

原則としてセッションはログイン成功後に開始することを求めています。ログイン成功前にセッションが開始されていると、セッション ID の固定化攻撃（Session Fixation Attack）が成立しやすくなるからです。

4.1.10. セッションの有効期間

セッションの有効期間は 30 分とすること。

＝解説＝

利用者が操作をしない状態で一定時間経過すると、セッションを破棄することを求めています。セッションが長時間有効な場合、XSS 攻撃や CSRF 攻撃等の影響を受けやすくなるためです。

なお、30 分としているのは例示のため、サイトの性質に応じて決定してください。

4.1.11. セッションの終了

次の場合はセッションを終了し、セッション情報を破棄すること。

(1) 利用者がログアウト機能を呼び出した場合（ログアウトボタンを押す等）

(2) 最後にページが表示された時刻を起点としてセッションの有効期間を超えた（セッションタイムアウト）場合

＝解説＝

ログアウト機能とは、利用者の指示によりセッションを破棄する機能のことです。認証を要する各ページにログアウトボタン等を設け、ログアウトができるようにすること、ログアウト処理でセッションを破棄することが必要です。

4.2. 認可処理

＝解説＝

認可とは認証された利用者に対して権限を与えることです。次のような認可対象の例が挙げられます。

・認証済みの利用者のみに許可された機能実行

予約の完了、新規ユーザの作成（管理者権限）、退会処理など

・認証済みの利用者のみに許可された情報の閲覧

非公開の個人情報の閲覧（利用者として、管理者として）、個人宛てのメッセージの閲覧など

・認証済みの利用者のみに許可された編集操作

利用者による設定変更（パスワード変更、登録個人情報の変更、画面表示設定など）など

4.2.1. 認可処理の要件定義と文書化

認可処理は次のとおり文書化し、権限毎の役割をロールとして作成すること。

(1) 認可処理の必要な機能、情報を識別して、認可処理の必要な画面には、『5.3. 最終提出物（1)』で示す画面遷移図上に識別マーク等をつけること。

(2) 各ロールと権限を一覧表（権限マトリックス）に整理すること。

＝解説＝

認可処理を正しく実装するためには、次のように認可処理の要件をまとめ、文書化することが望ましいといえます。

まず、認可処理の必要な機能、情報を識別して、認証処理の必要な画面に、画面遷移図上でマークを付けます。これにより開発やテストを正確に行うことができます。

次に、各ロールと権限の一覧表を文書化することも肝要です。「ロール」とは、権限を組み合わせて役割を示す名前を付けたものです。ロールを利用者とは独立して存在させ、利用者ごとにロールを割り当てる使い方をします。これにより 1 人 1ID をもつこととなり、担当者の追跡を容易にし、パスワードの共有による事故を防ぐことができます。一覧表を文書化しておくことで、開発やテストを正確に行うことができ、第三者による認可処理の不備有無を確認する診断においても、仕様の妥当性を確認しやすくなります。

4.2.2. 認可処理の実装

(1) 各利用者の権限確認には、セッション変数に保存された利用者識別情報（利用者 ID 等）を基準とすること。

(2) 認可を要する情報表示や機能実行をする前に、実行中の利用者が、当該情報の表示や機能を実行するための権限を有していることを画面毎に確認すること。

(3) 認可されなかった場合は、適切なエラー表示をすること。

＝解説＝

認可処理を正しく実装していないと、認可処理の不備により認可されていない者による不正利用ができてしまいます。認可処理の正しい実装方法は次のとおりです。

まず、各利用者の権限の確認には、セッション変数に保存された利用者情報を基にします。よくある認可処理の不備に hidden パラメータやクッキーに権限情報を保持し、それに基づき認可処理をしていることがありますが、hidden パラメータやクッキーを書き換えると権限を不正利用できてしまいます。外部から書き換えのできないセッション変数に保持することが重要です。

次に、情報表示や機能実行の前に、利用者が必要な権限を有することを確認することが挙げられます。万一、情報表示や機能実行ページの URL 等が知られてしまった場合でも、権限を必要とする処理の前にパスワードが正しいかどうかを確認することで、操作の権限がある利用者か否かを確実にチェックすることができます。また、操作権限がない場合には権限を有する人を示さずに「××を表示する権限がありません」等、適切なエラー表示を行います。

4.3. アカウント管理

＝解説＝

団体がインターネットで提供する Web アプリケーションには、不特定多数の住民に当該 Web アプリケーションの利用権を付与することがあります。また、利用者 ID、パスワード、メールアドレスのほか、必要に応じて個人情報を収集することがありますが、特に利用者 ID （ログイン ID） 、パスワード、メールアドレスの管理に不足があるとセキュリティ上の問題に直結しやすくなります。

ここでは、アカウント管理において実装を要する機能について、その必要とする背景や、注意事項を解説します。

4.3.1. 利用者登録（アカウントの作成）時における登録メールアドレスの確認

(1) 利用者登録時にメールアドレスを登録させること。

(2) 利用者によって登録されたメールアドレスに対してメールを送付し、登録メールアドレスが利用者に利用されているアドレスであることを確認する処理を実装すること。

(3) 登録メールアドレスが利用者に利用されているアドレスであると確認できた後に本システムにおける利用者登録を完了（登録の確定）とし、利用者登録の完了を経てからアカウントを作成すること。

(4) 登録されたメールアドレスに対してメールを送付する際に、利用者が登録したパスワードを記載しないこと。

＝解説＝

認証が必要なサイトでメールアドレスは重要な役割があります。パスワードリセット機能で使用するほか、 パスワード変更やアカウントロック発生時の通知などに利用することができます。故に例えば、パスワードリセット機能のあるサイトの場合、その機能を使った通知メールが誤って別人に届いてしまうと、なりすましを引き起こす原因となることがあります。

Web アプリケーションにおいて利用者(利用者 ID)とメールアドレスを紐づけて初期登録や登録変更する際には、登録されたメールアドレスにメールを送信し、当該利用者が受信できること、つまりメールアドレスが真に登録利用者のもので、正しいことを確認する必要があります。

具体的には、例えば次の方法が考えられます。

・利用者が登録したメールアドレスにトークン付き URL を添付したメールを送付し、その URL をクリックしたことでメールアドレスが正しいか否かを確認の上、処理を継続する。

・利用者が登録メールアドレスを入力した後、Web アプリケーションから当該メールアドレスにトークン

（確認番号）を送信し、アドレス登録後に遷移した入力画面において送信したトークンを入力してもらい、当該メールアドレスが正しいか否かを確認する。

4.3.2. 利用者IDの重複防止機能

利用者 ID が重複しないよう、チェック処理を含めること。

＝解説＝

利用者 ID の一意性は当然の要件ですが、まれに利用者 ID の重複を許してしまう実装があります。同一利用者 ID で別のアカウントが作成できると、誤って別の利用者としてログインできてしまう可能性があり、注意が必要です。これを避けるため、データベース上で利用者 ID を保存する列には、データベースの一意制約を付けることが望ましい実装です。データベース上でできない場合は、アプリケーション側で利用者 ID の重複を防ぐ排他制御等を実装する必要があります。

4.3.3. 登録メールアドレス変更機能

(1) 利用者が登録したメールアドレスを変更する機能を実装すること。
(2) メールアドレス変更機能の実行前に、パスワードの入力を利用者に求め、正しいパスワードであることを確認すること。
(3) メールアドレス変更機能の実行後は、利用者登録時と同様の処理を経ること。
(4) 変更前のメールアドレス（旧メールアドレス）にも登録メールアドレスが変更された旨の通知をメール送付すること。

＝解説＝

メールアドレス変更も下記のパスワード変更時と同様の配慮が必要です。

4.3.4. パスワード変更機能

(1) 利用者がパスワードを変更する機能を実装すること。
(2) パスワード変更機能の実行前に、現在のパスワードの入力を利用者に求め、正しいパスワードであることを確認すること。
(3) パスワード変更機能の実行後に、登録されているメールアドレスへ、パスワードが変更された旨の通知をメール送付すること。

＝解説＝

利用者が適宜パスワード変更できる機能を実装しておくことは重要です。例えば、利用者側の何らかの理由によりパスワードが漏えいしてしまった場合や、初期パスワードを発行、配付するタイプの Web アプリケーションにおける対応等が考えられます。

まず、パスワード変更においては当該機能の実行前に、現在のパスワードを入力してもらい、正しいパスワードか否かを確認することが肝要です 。これにより、セッションハイジャックされた状態で第三者がパスワードを変更することを防止できるほか、CSRF 脆弱性対策にもなります。また、パスワード変

更のような重要な処理が実行された場合は、その旨を利用者にメール通知することで、万一第三者が不正にパスワード変更した場合であっても、早期に利用者が異変に気づくことができ、必要な対応を取ることを可能にします。

4.3.5. パスワードリセット機能

利用者がパスワードを失念した場合の対処機能は次の(1)、(2)いずれかの方式とし、(3)または(4)の要件を満たすこと（利用者確認の手段として、予め登録したメールアドレスに宛てたメールが受信できることを用いる）。

(1) パスワードリセット機能を利用するための URL を登録メールアドレスにメール送付する方式

(2) 仮パスワードを発行し、メールで通知する方式（仮パスワードでログインした場合は、パスワード変更機能のみが利用できるものとする）

(3) (1)の機能の実装に際して、第三者がパスワードリセット機能を使えないように、URL には十分長い乱数による秘密情報（以下「トークン」という。）をつけること

(4) (2)の機能に対する総当たり攻撃対策を施すこと。

＝解説＝

本項はオプション提案です。

当該 Web アプリケーションの利用者がパスワードを失念してしまった場合、パスワードリセット機能があることで、パスワードを早期に提供することができ、Web アプリケーションの管理者の手間を軽減し、利用者の利便性を高める効果があります。

パスワードリセットの実装には、上に挙げた(1)、(2)の 2 種類の方式がよく知られています。実装上の要件に対策条件を加えたとおりパスワードリセット機能はなりすまし等の危険性を伴うため、そのリスクを正しく評価し、利便性とセキュリティレベルのバランスを図る必要があります。

なお、トークンに利用する乱数は、暗号論的擬似乱数生成器を使います。

4.3.6. 管理者によるアカウント削除・一時利用停止機能

(1) 管理者による利用者アカウントの削除機能を実装すること。

(2) 管理者による利用者アカウントの一時利用停止機能を実装すること。

＝解説＝

本項はオプション提案です。

本機能は、利用規約違反等善良な利用者ではないアカウントの停止等が考えられます。例えば地域 SNS のようなコミュニティサイト等における実装が想定されます。本機能は『4.3.7 利用者によるアカウント削除機能』と同様、退会処理をしようとしている Web アプリケーション管理者の本人確認のためにパスワード再入力を求めるほか、CSRF 対策（パスワードの再入力にて対策可能）、SQL インジェクション対策が万全か、十分配慮する必要があります。

4.3.7. 利用者によるアカウント削除機能

(1) 利用者による自身のアカウント削除機能を実装すること。

(2) アカウント削除機能の実行前に、パスワードの入力を利用者に求め、正しいパスワードであることを確認すること。

(3) アカウント削除機能の実行後、登録されていたメールアドレスにアカウントが削除された旨の通知をメール送付すること。

＝解説＝

本項はオプション提案です。

利用者による退会処理は、Web アプリケーション管理者の管理負担を軽減し、利用者に利便性を提供するものですが、通常取り消しができない処理となるため、この機能で退会処理をしようとしている利用者の本人確認（本人からの退会請求であるか）のためにパスワード再入力を求めるほか、CSRF 対策（パスワードの再入力にて対策可能）、SQL インジェクション対策が万全か、十分配慮する必要があります。

退会処理がオンライン上で即座に行える必要があるケースは、例えば月額課金を伴うサービスなどが考えられます。それ以外のケースではそのような即座の処理が真に必要か否かについて十分検討すべきです。

4.4. ログイン状態にある利用者の意図に反した機能実行の防止機能

外部リンク等により本システムの画面(機能)に遷移するだけで、本システムの機能がログイン状態にある利用者の意図に反して実行されることを防止すること。なお、ここで言う「ログイン状態にある利用者の意図に反した機能実行の防止」とは、クロスサイト・リクエスト・フォージェリ(以下「CSRF」という。）対策及びクリックジャッキング対策を指す。

＝解説＝

Web サイトには、画面表示以外の機能を持つ場合があります。具体的には、データの投稿、データの更新、パスワードの変更、予約、商品等の購入、送金などです。これら機能をログインしている利用者の意図に反して第三者が実行させる、クロスサイト・リクエスト・フォージェリ（CSRF）とクリックジャッキング攻撃という手法が知られています。ここでは、これら攻撃への対策を求めています。

4.4.1. 該当画面の洗い出し

CSRF 対策及びクリックジャッキング対策を施すべき画面(機能)を洗い出し、『5.3. 最終提出物（1)』で示す画面遷移図上に識別マーク等を付けること。なお、当該機能のページは POST メソッドで呼び出すようにすること。

＝解説＝

すべての機能に対して CSRF 攻撃及びクリックジャッキング攻撃の対策が必要とは限りません。影響が軽微な場合は、これら攻撃を受容する場合があります。そのような処理の例としてログアウト処理があります。このため、CSRF 攻撃及びクリックジャッキング攻撃の対策を施すページを設計時に検討して、

その結果を画面遷移図に識別マークをつけるよう要求しています。

4.4.2. CSRF対策

対策対象の画面(機能)を実行する前のページにてトークンを生成して埋め込み、処理を実行する際は、その値が正しい場合のみ実行すること。

＝解説＝

CSRF 攻撃に対してはトークンを用いた対策を求めます。

(1)　機能を呼び出すページにトークンを埋めこむ

(2)　機能を実行するページではトークンの正当性を確認する

トークンとしては第三者が推測できない文字列を用います。具体的には、暗号学的に安全な擬似乱数生成器により発生させる方法があります。

4.4.3. クリックジャッキング対策

対象画面の 1 つ手前の画面にて、次の(1)、(2)いずれかの HTTP レスポンスヘッダを出力すること。なお、対象画面以外にも出力してよい。

(1) X-FRAME-OPTIONS: DENY

(2) X-FRAME-OPTIONS: SAMEORIGIN

＝解説＝

クリックジャッキング攻撃は、見えない frame や iframe に攻撃対象ページをはめこみ、視覚的な錯誤を使って利用者にボタン等をクリックさせる攻撃です。対策としては、frame や iframe 内に表示することを禁止する HTTP レスポンスヘッダを出力します。

4.5. ログ出力

システム監査、事故調査を目的として次によりログを出力・保管すること。

＝解説＝

発注者のセキュリティポリシーにログの規定がある場合は、その規定に従って検討します。なお、ここで記載している各種ログの項目等は例です。システムの規模や種別によってはログ出力サイズが大きくなってしまい、パフォーマンスに影響を及ぼす可能性がある場合等も想定されますので、必要性とのバランスを加味して決定する必要があります。

4.5.1. 出力するログの種類

次のログを出力すること。

(1) Web サーバのアクセスログ

(2) アプリケーションログ

(3) データベースのアクセスログ

(4) エラーログ

＝解説＝

本項(3)データベースのアクセスログ はオプション提案です。

何をログ出力するかは、Web サイトごとに定義する必要があります。Web サーバのアクセスログとアプリケーションログは必須としています。データベースのアクセスログを取得する場合は、同ログからの情報漏えいに留意する必要があります。

エラーログは、文字通りアプリケーションの様々なエラーを記録するものです。Web アプリケーションでエラーが発生した場合は、画面には「アクセスが集中しているのでしばらく待ってからお試しください」のような利用者向けのメッセージを表示しておき、エラーの詳しい内容や原因はログに出力するようにします。

4.5.2. 出力しないログの種類

次のログ取得については、構築時、動作テスト時には出力してよいが、本番稼働時までに無効にしておくこと。ただし、システム検証やトラブル対応のために、*本市（都道府県・区・町・村）*の管理者が認めた場合は除く。

(1) デバッグログ

＝解説＝

デバックログは通常のログよりも出力データ量が多い上、取得が不必要な個人情報等を含むこともあるため、本番稼動後は停止します。

4.5.3. アプリケーションログで取得するイベント

次のイベントをアプリケーションログにて取得すること。なお、次に記載していない他のイベントも取得してもよい。

(1) ログイン（成功・失敗問わず）
(2) ログアウト
(3) アカウントロック
(4) 利用者登録・登録削除
(5) 利用者の登録内容更新
(6) 利用者のパスワード変更
(7) 秘密情報の参照
(8) その他重要な操作（CSRF 対策の対象となる操作は必須）

＝解説＝

アプリケーションログはエラーのみを取得するのではなく、セキュリティ上重要な処理が行われた際の正常系のログも取る必要があります。上記はログを取得すべき代表的なイベントですが、アプリケーション毎にログの要件として、取得すべきイベントを定義します。

4.5.4. 出力するログの項目

次の情報をログに含めること。なお、これ以外の情報を含めても良い。

(1) アクセス日時（年、月、日、時、分、秒）
(2) アクセス元 IP アドレス（IPv4 又は IPv6）
(3) 利用者 ID
(4) アクセス対象（URL 又はページ番号等）
(5) 操作内容
(6) 操作対象（利用者 ID、文書 ID など）
(7) 実行結果（成功あるいは失敗、処理件数など）

＝解説＝

各ログの出力項目は、4W1H（いつ、誰が、どこで、何を、どのように）に従った項目を取得する必要があります。

4.5.5. 出力しないログの項目

次の情報はログの項目として取得しないこと。

(1) パスワード

＝解説＝

『4.5.6 ログからの情報漏えい・改ざん対策』 にあるように外部・内部問わずログに対する不正アクセスを防止することはもちろんですが、万一ログが第三者に参照されてしまった場合も考慮し、パスワ

ード等の重要情報はアプリケーションログには出力しないようにします。

4.5.6. ログからの情報漏えい・改ざん対策

(1) ログが不正に参照・変更・削除されないよう保護すること。

(2) ログから個人情報等の秘密情報が漏えいすることを防ぐため、ログの目的（監査、事故追跡）を損なわない範囲で秘密情報を含めない処理又は秘密情報の一部のみの出力(マスク処理)をすること。

＝解説＝

ログが改ざん・削除されるとログの目的を達成できないため、ログに対する不正アクセスができないよう保護対策をする必要があります。また、万が一改ざん・削除された場合は通知する仕組みが必要です。ほか、ログファイル自体に個人情報、プライバシー情報など機密情報が含まれることもあるので、権限のある者のみが閲覧できるように制限し、ログ取得の目的に合致しない個人情報等が含まれないようにログ取得をします。可能であればログを保存するサーバは Web サーバや DB サーバとは別に用意し、サイト管理者とは別に、ログ管理者を割り当てることが望ましい対応です。

4.5.7. ログの保管

(1) ログの保管年限は 3 年とする。

(2) ログの安全な保管方法（媒体、保管フォーマット、保管場所等）を定めること。

＝解説＝

Web サイトの特性に合わせてログの保管期限を運用ルールとして定めておく必要があります。セキュリティ上の事件や事故の事後調査という目的を考慮すると、無期限で保存するという考え方もありますが、保存場所の制限等様々な理由により 3 年や 5 年などと保管期限を定義しておく場合があります。

ログの保管方法については、電子媒体へ定期的に記録しておくことで、長期保存が可能となります。また、過去のログデータは常に利用するデータではないため、普段は安全な場所に電子媒体を保管しておくのが望ましい対応です。

4.6. 暗号化

＝解説＝

本項（4.6 項すべて）はオプション提案です。

SSL/TLS（Secure Sockets Layer と、Transport Layer Security の略。以下合わせて「SSL」と略す。）には、通信回線の暗号化機能と、第三者機関（Certification Authority、以下「CA」という。）によるドメイン名の正当性証明機能があります。

インターネットの通常の通信では盗聴やサーバのなりすまし攻撃を受ける可能性がありますが、SSL を使うことによって安全性が高まるため、ログイン画面や会員登録画面など機密情報を送受信する Web サイトでは、SSL を用いた運用を行うことを推奨します。

4.6.1. 利用者と本システム間におけるWebアプリケーション通信の暗号化

(1) システムで送受信する情報のうち、秘密情報に該当するものを要件定義時に一覧表にまとめること。

(2) 利用者と本システム間で秘密情報を送受信する際に利用する画面（機能）を SSL/TLS の利用対象とし、『5.3. 最終提出物 (1)』で示す画面遷移図上に識別マーク等を付け、そのとおりに実装すること。

(3) サーバ証明書は利用を想定するすべてのブラウザで警告の出ないものを使用し、証明書の発行先名は、運営者の名称とする。地方公共団体組織認証基盤（LGPKI）を用いる場合は、Firefox を利用想定ブラウザから外すこと。

(4) SSL2.0 は使用しない設定にすること。

＝解説＝

まず、SSL を利用するにあたって守るべき対象を明確にするために、秘密情報を定義して文書化する必要があります。

秘密情報の例：

・住民の個人情報（名前、住所など個人を特定できる情報）

・住民のプライバシー情報（図書貸出し履歴等）

・利用者 ID／パスワード等ログイン情報

次にどこを守るのかですが、Web サイトにおける SSL 使用は、Web サイト全体を SSL で保護する場合と、一部のページを SSL で保護する場合の 2 通りがあります。ここで記載した内容は、一部を保護する場合を主に想定した記載となっています。設計時には SSL で保護する対象画面を定義し、文書化してください。特に、入力画面を SSL の対象とすることによって、当該ドメインのドメイン正当性が保証できます。入力画面を SSL の対象にしないと、見た目を似せたフィッシングサイトなどの偽のページと見分ける手段がありません。

また、ブラウザは、サーバ証明書に問題がある場合に警告を表示する仕様となっています。SSL によるドメイン名の正当性証明機能は、ブラウザで警告が出ないことで実現しています。そのため、利用想定ブラウザすべてで警告の出ないようにする必要があります。なお、LGPKI の証明書については、Firefox のデフォルト設定では警告が出てしまうため、Firefox の利用者には LGPKI の証明書入手と正しいインストール方法及び正しくインストールするための確認内容・方法を伝える必要があります。

参考：https://www.lgpki.jp/CAInfo/install.htm

携帯電話の場合は正規のサーバ証明書でもエラーが出る場合があります。携帯電話の場合は、使用できる証明書が限定されるため利用者環境想定の対象とする携帯電話の事業者に確認し、エラーの出ないサーバ証明書を使用する必要があります。

【参考情報】主要な携帯電話キャリアのサーバ証明書一覧（2012 年 10 月現在の URL であり、変更となる可能性があります。）

NTT docomo　http://www.nttdocomo.co.jp/service/developer/make/content/ssl/spec/index.html

au(KDDI)　http://www.au.kddi.com/ezfactory/web/index.html

Softbank Mobile　http://creation.mb.softbank.jp/mc/tech/tech_web/web_ssl.html

なお、プロトコルの選定における注意点としては、SSL2.0 にはプロトコル上の脆弱性が指摘されているので、サーバの設定により SSL2.0 を使わないようにすることが挙げられます。

4.6.2.　内部の通信に関する補足

インターネットを介さない、内部の秘密通信については暗号化ではない方法による通信の秘密確保も可とする。通信の秘密確保方法について提案書に記載すること。

＝解説＝

暗号化ではない方法の例としては、データセンター内の物理的に保護された LAN 上の通信や、データセンター間を専用線で接続している場合等を想定しています。

4.6.3.　データベースの暗号化

(1) 秘密情報をデータベースに保存する際は暗号化を施すこと。

(2) 暗号化アルゴリズムは電子政府推奨暗号リストに記載されたアルゴリズムを用いること。

(3) 暗号鍵の管理方法を提案書に記載すること。

＝解説＝

インターネット経由の攻撃に対する「出口対策」としてデータベース暗号化が有効となります。

データベース暗号化には 2 種類の方法があります。1 つが、アプリケーションによる暗号化、もう 1 つがデータベースの機能による暗号化(透過的暗号化: Transparent Data Encryption（以下、「TDE」という))です。

TDE はデータベースに対してデータ挿入、検索に応じて自動的に暗号化をするためのもので、取り扱いは簡便ですが SQL インジェクション等の攻撃に対しても復号済みのデータを返すため、インターネット経由の攻撃に対する対策としては適しません。

インターネット経由の攻撃への対策となるデータベース暗号化の手法としては、アプリケーションで暗号化・復号を行い、暗号化されたデータをデータベースに保存する方法が知られており、妥当な方法です。なお、その方法の場合、暗号鍵の保存方法が重要となりますが、その点は提案者（受注者）へ提案を求めることとしています。

4.6.4.　ファイルの暗号化

(1) 秘密情報をファイルに保存する際は暗号化を施すこと。

(2) 暗号化アルゴリズムは電子政府推奨暗号リストに記載されたアルゴリズムを用いること。

(3) 暗号鍵の管理方法を提案書に記載すること。

＝解説＝

ファイルの暗号化は必須ではありませんが、万一ディレクトリ・トラバーサル脆弱性などによりファイルが漏えいした際の保険的な対策として暗号化を要求することも有効です。あるいは、ファイル漏えい対策について提案者（受注者）に提案を求めるのもよい方法と考えます。

なお、暗号化のアルゴリズムは電子政府推奨暗号リストに記載されたアルゴリズムを用いることを要求することが肝要です。ベンダーが独自開発し、世に出ていないアルゴリズムを採用してはいけません。また、前節同様、暗号鍵の管理方法は重要ですが、実装方式に依存するためここでは提案者（受注者）に提案を求める形としています。

5.　テスト（検査）要件

本システムの構築にあたって次のテスト（検査）を行い、報告書を提出すること。

5.1. 開発時中間検査

受注者は、開発途中に*本市（都道府県・区・町・村）*が指定する時期に、以下の中間検査を実施し、報告すること。

仕様　　:『LASDEC ウェブ健康診断仕様について 平成 22 年度版』

検査箇所：受注者、*本市（都道府県・区・町・村）*協議の上、抜取検査が可能な箇所を特定する。

提出物　：セキュリティ検査結果報告書、検査ログ（データ）

＝解説＝

本項はオプション提案です。

開発時中間検査は、Web アプリケーション開発の途中で脆弱性検査を実施することにより、開発者のセキュリティ習熟度や、開発標準の不備を検査することを目的としています。この中間検査の成績があまりに悪い場合は、受注者に対して開発体制の改善を申し入れます。開発者の教育やガイドラインの見直し、レビューの強化などが必要な施策となります。

なお、開発時中間検査は受注者の自己検査を想定しています。検査箇所や時期については、発注者、受注者の協議により決定します。

また、中間検査はある程度開発規模の大きな Web アプリケーションを想定したものです。小さな Web アプリケーションや、パッケージ Web アプリケーションの場合、ほぼ最終検査と同等の時期に中間検査を実施することが想定され、改善効果が期待できないこともあるため、中間検査の実施を予定する場合は RFI 等で効果がありそうか否かを検討しておきます。

5.2. 出荷時検査（最終検査）

(1) 受注者は最終検査として脆弱性の有無を調べるセキュリティ検査を実施し、本書の要件を満たしていることを確認すること。

(2) 検査手法(概要)、検査項目、検査箇所、検査箇所ごとの検査結果を記載したセキュリティ検査報告書を任意の書式で作成し、提出すること。

(3) セキュリティ検査ログのデジタルデータをセキュリティ検査報告書に含めて提出すること。検査ログを提出できない合理的な理由があると認めるときは、DVD-R など変更できない媒体に書き込み、セキュリティ保証期間は受注者が厳重に保管するとともに、データのハッシュ値を提出すること。

(4) 納品時における OS、ミドルウェア等ソフトウェアのバージョン及び最新パッチのリリース状況を確認すること。

＝解説＝

出荷検査は Web アプリケーションが仕様を満足することを確認するために行いますが、従来、セキュリティ要件の検査は曖昧なケースが多く見受けられたところです。このため出荷検査としてセキュリティ要件の検査を明示的に要求しています。発注者は、検査結果の受け入れ時には検査手法(概要)、検査項目、検査箇所、検査箇所ごとの検査結果の網羅性を確認することを想定しています。

なお、要件において検査ログの提出あるいは保管を要求している理由は、Web サイト公開後に脆弱性が発覚したり、指摘された場合に、出荷時検査での状況を事後確認するためです。また、受注者側の（脆弱性検査等の）ノウハウ流出防止等の理由で検査ログを提出できない場合は、受注者が安全にログを保管するとともに、当該ログデータのハッシュ値も成果物として発注者に提出してもらいます。ログデータのハッシュ値は、後日ログを閲覧する際に当該ログが改ざんされていないことを確認するために取得するものです。

また、出荷時におけるバージョンの把握は、以降のパッチ適用判断における基本情報として必要なことは言うまでもありません。タイミングによっては運用開始直前・直後にパッチ公開されることなどもあることから、どこでサービス開始時のバージョン（パッチ適用状態）とするかを決めておきます。

5.3. 最終納品物

本書による納品物は次のとおり。

(1) 画面遷移図

次の各項目の識別マーク等をつけた画面遷移図を提出すること。

ア アクセスするためにログイン処理が必要な画面(機能)

イ 認可処理の必要な画面(機能)

ウ 暗号化通信の適用画面(機能)

エ CSRF 対策の実施画面(機能)

オ パッケージソフトウェアを用いる場合、カスタマイズした画面（機能変更した画面）

カ パッケージソフトウェアを用いる場合、新規追加した画面（機能追加した画面）

＝解説＝

画面遷移図については既に『4. セキュリティ機能』の該当箇所で説明した通りです。Web アプリケーションが実現する複数の画面と、その画面の変化状況を規定する画面遷移図に、各項目のマークを付与してもらうことで、テストにおいて正しく実装されているかどうかを確認することができます。

なお、オ及びカは、パッケージ Web アプリケーションを用いる場合に、カスタマイズを施したページや新規追加したページの識別を求めるものです。これにより、外部からの脆弱性指摘があった場合に、パッケージに元々脆弱性があったものか、新規追加ないしカスタマイズしたことにより脆弱性が混入したものかの区別が可能となります。

(2) 権限マトリックス表

『4.2.1. 認可処理の要件定義と文書化』に従い作成された各ロールと権限を一覧にした表を提出すること。

＝解説＝

これは『4.2.1. 認可処理の要件定義と文書化』で要求しているロールと権限の一覧表（権限マトリックス表）を成果物として要求しています。

(3) セキュリティ検査報告書

次の資料を提出すること。

ア『5.1. 開発時中間検査』に従い作成されたセキュリティ検査報告書[13]

イ『5.2. 出荷時検査（最終検査)』に従い作成されたセキュリティ検査報告書

ウ『5.2. 出荷時検査（最終検査)』にあるセキュリティ検査時の検査ログデジタルデータ。提出時の媒体等は容量を勘案の上、別途指定する。

エ 納品時のソフトウェアの最新パッチに関する情報が掲載されている URL 又は最新パッチに関する情報を入手する手段を記載した報告書。書式は任意とする。

オ 納品時のソフトウェアに納品時点の最新パッチが適用されていない状態で納品する場合は、当該パッチ未適用時にもたらされる脅威に関する説明及びその回避策を記載した報告書。書式は任意とする。

＝解説＝

『5.2. 出荷時検査（最終検査)』で説明した通りです。

アはオプション提案の『5.1. 開発時中間検査』を実施した場合のみ、中間検査終了後に求めます。

[13] （オプション）発注者が中間検査を求めない場合は本項目を削除すること。

6.　検収

本システムの Web アプリケーション脆弱性対応は次のとおり検収する。

6.1.　脆弱性検査結果の確認

(1) 本書で定める要求を満たしていることを受注者の提出した検査報告書をもって検査する。

(2) 本書で定める要求を満たしていることを『LASDEC ウェブ健康診断仕様について 平成 22 年度版』を用いて*本市（都道府県・区・町・村）*が検査する。

(3) 本書で定める要求を満たしていることを*本市（都道府県・区・町・村）*が指定する第三者の実施する脆弱性診断結果をもって検査する。

＝解説＝

発注者は、(1)～(3)のいずれかを選択します（複数選択し、条件としてもよい)。

発注者には検収の責任があり、納品された Web アプリケーション品質の妥当性を検収時に確認する義務があります。これは、セキュリティ要件についても例外ではありません。しかしながら、セキュリティ要件の検査には専門的な技術が要求されるため、発注者自らがセキュリティ検査を実施する例はまだ多くなく、上記（1）によりセキュリティ要件に対する検収としている（あるいは何もしない）案件が多いのが実情です。

売買契約上の原則に従う意味と、安全な Web アプリケーションを公開するという意味から、発注者責任として、発注者がセキュリティ検査を実施することが望ましい対応です。すなわち、(2) 又は (3) が望ましいことになります。

(2) のウェブ健康診断仕様を検収に用いる場合、検査項目がモデルプランにおけるセキュリティ要求仕様の一部しか網羅していないことや、元々抜き取り検査のための仕様であることが課題となります。(2) にする場合は、受注者との間で、検収をウェブ健康診断仕様による抜き取り検査とすることについて合意を得ておく必要があります。

(3) の第三者による診断の場合、診断事業者独自の診断項目だけによらず、受注者のセキュリティ実装方針に沿った診断内容等を踏まえ、発注者が合意した内容で診断するのが望ましいといえます。なお、診断事業者が、セキュリティ要件に記載のない事項について脆弱性ありと指摘した場合、当該指摘内容（脆弱性）は瑕疵にはあたらないため発注者と受注者で協議の上、対応を決定することになります。

6.2.　実装状況報告書の確認

『3.2. セキュリティ実装方針の提出』に示したセキュリティ実装方針に基づき、『別紙 6 実装状況報告書』を検査する。

＝解説＝

実装状況報告書の検査では、発注者はすべての実装方針項目に対応していることを確認します。万一非対応の項目があった場合は、回避策の有無と回避策の具体的内容を確認し、受容するか否か判断します。

7.　セキュリティ保証期間中の脆弱性対応

本システムにおけるセキュリティ保証期間中の脆弱性対応として、次の事項について誠意をもって行うこと。

7.1. セキュリティ保証期間中の脆弱性対応（パッチの開発・提供）

セキュリティ保証期間中に発見された脆弱性への対処について、以下の場合は追加費用なしで修補（パッチの開発・提供）すること。

(1) 『別紙 1 脆弱性リスト』に含まれる脆弱性で、受注者が対処済みである旨をセキュリティ実装方針によって宣言したもの。

(2) 本書によらず、受注者が追加提案として対処を約束した脆弱性。

なお、修補（パッチの開発・提供）以外の代替案によって脆弱性の影響を回避できる場合は、受注者、*本市（都道府県・区・町・村）*双方協議の上、代替案による対処も可とする。

ただし、次の場合は対応内容及び費用負担について、受注者、*本市（都道府県・区・町・村）*双方協議の上、決定する。

ア 『別紙 1 脆弱性リスト』に含まれない脆弱性が発見されたとき。

イ 受注者が追加提案として対処を約束した脆弱性に含まれない脆弱性が発見されたとき。

＝解説＝

『はじめに　(5) (参考)モデルプランに記載された具体的セキュリティ要求仕様の選定基準』で述べたとおり、モデルプランでは、「未知の脆弱性」「新しく発見された脆弱性でまだ対処方法が確立していない脆弱性」「“脆弱性である”と定義できるかどうかわからないグレーゾーンの脆弱性」等は納品時に対処が必要な脆弱性として定義していません。特記仕様書中で対応必須とされていない脆弱性が発見された場合は、受注者の対応範疇外（瑕疵ではない）とし、団体がこの新たな脆弱性を対処したい場合は受注者と協議の上、対応内容及び費用について決定する必要がありますが、(1)及び(2)については検収後に発見された「隠れた瑕疵」として、あらかじめ決めた Web サイトのセキュリティ保証期間内は受注者に追加費用なしで当脆弱性を対処することを求めています。

なお、重要事項説明書に代替案が記載されており、発注者が当該対処に代替案を適用することが妥当であると認めた場合は、代替案による対処も可能としています。一般に、脆弱性の対処の代替案としては、不正侵入防御装置（IPS）や Web アプリケーションファイアウォール（WAF）によるバーチャルパッチ対応等が挙げられます。

8.　保守要件

本システムの保守対応（パッチの適用作業又はバージョンアップ作業）として、次の事項について誠意をもって行うこと。

8.1. 脆弱性対応基本方針

(1) 脆弱性対応作業費用については別途保守契約で定める。セキュリティ保証期間中に脆弱性対応のためのパッチ適用作業やバージョンアップ作業が発生し、費用を要する場合は、その費用を予め運用費用見積に含め、提出すること。

(2) セキュリティ保証期間中に本システムが、『別紙 1 脆弱性リスト』に記載の脆弱性に対応できていないことが判明した場合、これを追加費用なしで修補（パッチの適用作業又はバージョンアップ作業）すること。

8.2. パッチ適用ポリシー

(1) ソフトウェアのパッチは、パッチリリース後 1 週間以内に適用・非適用の方針を決めること。また最終結論の方針に依らず、その判断理由について報告すること。

(2) パッチ適用を決定した場合、パッチリリース後 2 週間以内に適用作業を完遂すること。また、適用作業終了後は、パッチ適用状況（適用の成功・不成功、動作への影響有無等）を報告すること。

(3) 前項(1)、(2)を保証できない場合、『1.3. 提案時の提出物 (3)』に従って資料を作成し、提出すること。

＝解説＝

本書では『1.2. 本システムのセキュリティ保証期間について』にてセキュリティ保証期間を 5 年と例示しています。保守においてはその期間中にパッチがリリースされることと、適用できることを求めています。

ここでは、上記『8.1. 脆弱性対応基本方針 (2)』に関連し、セキュリティ保証期間を 5 年と例示している理由を以下、補足します。

通常、ハードウェアの故障率の場合、次のような横軸に時間経過、縦軸に故障率の変化をとった故障率曲線（バスタブ曲線）を描くと言われています（図 5 参照）。

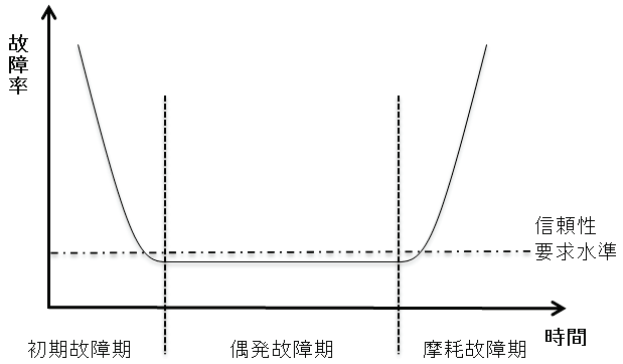

図 5 ハードウェアの故障率曲線

ハードウェアの瑕疵担保期間が 1 年とされるのは、初期不良の発生が顕著な期間（＝劣化によらない不良）を担保するためです。その後の故障は偶発とされ、保守による修補がなされるのが一般的です。

一方、ソフトウェアの故障（バグ）の発生に関しては、横軸に時間経過、縦軸に故障率の変化をとった次のような曲線を描くと言われています（図 6 参照）。

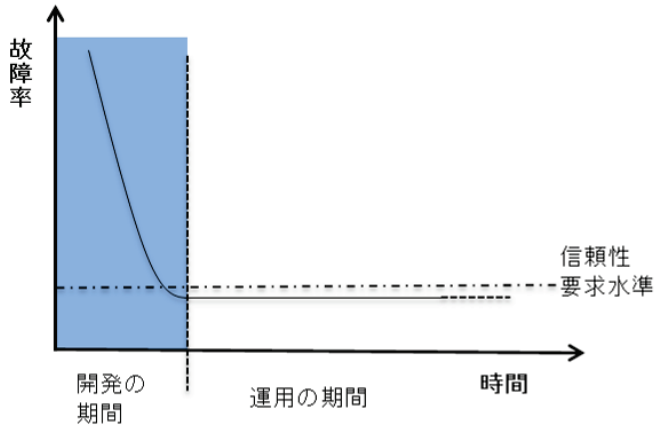

図 6 ソフトウェアの故障率(バグ)曲線

ソフトウェアの開発・保守における信頼性管理の要諦は、初期故障が出つくし、ソフトウェアの信頼性が実用に耐える水準に達したことをいかに見極めるかということに尽きますが、開発時のテストにおいて取りきれなかった残存バグ（機能要件を満たしていないもの）が運用期間中発生することもあります。

そのような残存バグも、およそ全ての機能を使いきるのが導入後 1 年以内（例えば、年度末の処理等も含め、全ての機能が業務上実行されるのにかかる年月）であるという傾向から、その 1 年の間に洗い出すことができるとし、業界慣習的に瑕疵担保期間は 1 年以内とされています。

しかしながら、ここで示されているソフトウェアのバグは、ソフトウェアの機能要件に係る内容のため 1 年以内が想定されているのであり、非機能要件であるセキュリティについては、そのバグ（脆弱性）はソフトウェア設計における想定外の操作をした場合に偶発的にしか発見されません。

例えば、第三者による脆弱性診断を実施する、あるいは第三者から脆弱性の存在を指摘される、そのような偶発がない場合の脆弱性件数は次ページ図 7（左のグラフ）、偶発があった場合は次ページ図 8（右のグラフ）のような傾向となります。いずれも、時間経過を横軸、脆弱性の残存数（ソフトウェアに内在しているもの）を縦軸としています。

脆弱性は通常の操作では発見できず、偶発的な事象からでしかその存在を確認できないことから、機能の不備とは異なりリリース後数年を経てから第三者が脆弱性を発見・指摘するケースも珍しくありません。

※なお、当然ながら内在する脆弱性数は、当該ソフトウェア開発者の設計・コーディング作業品質、出荷時の脆弱性検査品質等により上下し、運用当初から信頼性要求水準を十分クリアすることもあることを付記します。

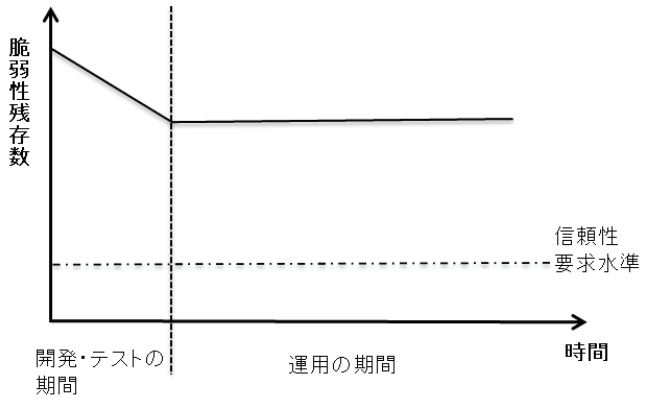

図 7 第三者による脆弱性指摘がない場合

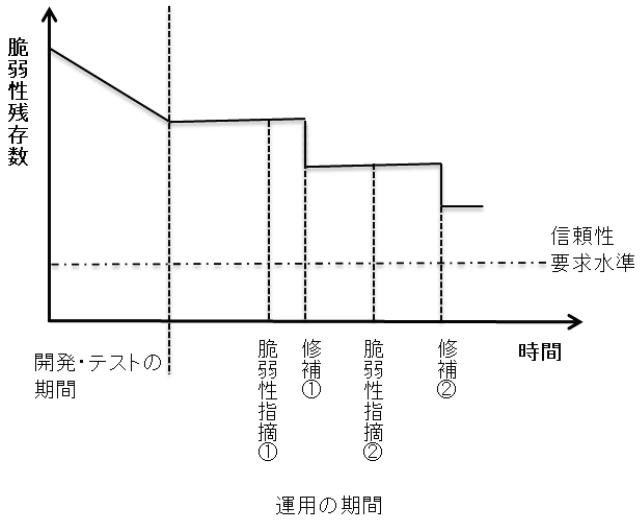

図 8 第三者による脆弱性指摘があった場合

このような背景及び理由から、通常のハードウェア瑕疵担保期間、ソフトウェア機能要件に係る瑕疵担保期間とは別に「セキュリティ実装方針における不備・欠落」「実装状況報告書の不備・欠落」があった場合を考慮し、ソフトウェアの脆弱性対応に関しては Web サイトの「セキュリティ保証期間」を定義し、同期間内は予め対処を約束していた脆弱性について追加費用なしに修補することを求めることとしました。そして、セキュリティ保証期間は原則としてサイトの稼働予定期間と同じとしています。

また、本書ではソフトウェアアーキテクチャの寿命及びソフトウェア製品のサポートライフサイクルも加味し、これを 5 年間と例示しました。

第5章 他の要件の参考情報、参考文献

(1) 他のセキュリティ要件に関する情報

内閣官房情報セキュリティセンター（NISC） 政府機関の情報セキュリティ対策のための統一管理基準及び解説書(平成 23 年 4 月 21 日)、
NISC 政府機関の情報セキュリティ対策のための統一技術基準及び解説書(平成 23 年 4 月 21 日)
http://www.nisc.go.jp/conference/seisaku/index.html#seisaku25
http://www.nisc.go.jp/materials/index.html

経済産業省「情報システム信頼性向上のための取引慣行・契約に関する研究会」最終報告書 ～情報システム・モデル取引・契約書～ の公表について（平成 19 年 4 月 13 日）
http://www.meti.go.jp/policy/it_policy/keiyaku/

NISC ソフトウェア開発における情報セキュリティ対策実施規程 （雛形）（平成 18 年 2 月）
http://www.nisc.go.jp/active/general/pdf/dm6-03-051_sample.pdf

総務省 個人情報の取扱いに関する特記仕様書(雛形) （平成 21 年 3 月 27 日）
http://www.soumu.go.jp/menu_news/s-news/02gyosei07_000006.html

経済産業省 平成 22 年度ソフトウェア開発管理基準に関する調査報告書（ソフトウェアメトリックス調査）の公表について（平成 23 年 2 月 28 日）
http://www.meti.go.jp/policy/mono_info_service/joho/softwaremetrics/2010/index.html

独立行政法人情報処理推進機構(IPA) セキュリティ要件確認支援ツールの公開 ～さまざまな情報システムにおける適切なセキュリティ要件定義を容易に～（平成 23 年 8 月 17 日）
http://www.ipa.go.jp/about/press/20110817.html

(2)　参考文献

IPA　「地方公共団体のための脆弱性対応ガイド」などを公開
～「情報システム等の脆弱性情報の取扱いに関する研究会」の 2011 年度活動成果～(平成 24 年 3 月 26 日)
http://www.ipa.go.jp/security/fy23/reports/vuln_handling/index.html

経済産業省 情報システムの信頼性向上のための取引慣行・契約に関する研究会 (2007)『「情報システム信頼性向上のための取引慣行・契約に関する研究会」最終報告書 ～情報システム・モデル取引・契約書～ (受託開発 (一部企画を含む)、保守運用) 〈第一版〉』経済産業省商務情報政策局情報処理振興課
http://www.meti.go.jp/policy/it_policy/keiyaku/model_keiyakusyo.pdf

経済産業省 情報システムの政府調達に係るSLA導入研究会(2004)『情報システムに係る政府調達へのSLA導入ガイドライン』
http://www.meti.go.jp/policy/it_policy/tyoutatu/sla-guideline.pdf

IPA　『安全なウェブサイトの作り方 (改訂第 5 版第 2 刷)』(平成 24 年 3 月 30 日)
http://www.ipa.go.jp/security/vuln/websecurity.html

LASDEC　ウェブ健康診断 診断仕様 (平成 22 年度版) の公開 (平成 23 年 5 月 19 日)
https://www.lasdec.or.jp/cms/12,1284.html#siyou-h22

徳丸浩 (2011)『体系的に学ぶ 安全な Web アプリケーションの作り方　脆弱性が生まれる原理と対策の実践』ソフトバンククリエイティブ

岡村久道(2011)『情報セキュリティの法律[改定版]』商事法務

碓井光明(2005)『公共契約法精義』信山社出版

地方公共団体における情報システム
セキュリティ要求仕様モデルプラン
（Web アプリケーション）解説書

2012 年 10 月 22 日公開

［技術監修・委員］
独立行政法人 産業技術総合研究所　高木　浩光

［執筆協力・技術アドバイザー・委員］
HASH コンサルティング株式会社　　徳丸　浩

［執筆協力・委員会事務局］
京セラコミュニケーションシステム株式会社
間嶋　英之
小関　直樹
原田　隆正

［編集・委員会事務局］
財団法人 地方自治情報センター
自治体セキュリティ支援室 （担当：百瀬、古家）
〒102-8419　東京都千代田区一番町 25 番
（全国町村議員会館内）
TEL. 03-5214-8040
Mail. lasc_info@lasdec.asp.lgwan.jp （LGWAN）
lasc@lasdec.or.jp （Internet）