ソフトウェア出荷判定セキュリティ基準チェックリスト解説書

CSAJ Computer Software Association of Japan
一般社団法人コンピュータソフトウェア協会

contents

1 はじめに

IoT社会の到来とともにITシステムが生活や産業に関与する比率は飛躍的に増加し、ソフトウェアの重要性が今まで以上に認識されるようになりました。一方そのソフトウェアの欠陥や、バグによる障害に起因する社会的な事故・事件も増加し、信頼性や安全性の担保が喫緊の課題となってきております。

しかしながらソフトウェアはその使用目的やサイズ、使用場所、稼働環境、開発者の違い等によりさまざまな手法が使われるため、標準化の遅れている分野でもあり、研究者や業界団体などにより信頼性を高めるためのガイドラインやテスト方法が提唱されていますが、採用される割合はいまだ高くありません。
さらにはソフトウェアに存在する脆弱性を外部または内部から攻撃し、システム運用を阻害したり重要な情報を窃取したりするいわゆる「サイバー攻撃」が増加したことにより、セキュリティがソフトウェアの品質に大きな影響を与えることになりました。

このような状況を背景に、一般社団法人コンピュータソフトウェア協会　セキュリティ委員会では、WGを発足し有識者有志による議論の結果、ソフトウェア開発・出荷におけるセキュリティの要検討事項についてチェックリスト及び解説書として、一般的なアプリケーションソフトウェア開発及び出荷基準として参照されるべきセキュリティの要件を定義しました。

周知の通り、セキュリティは新たな脅威の発生や環境の変化により常に変化していくものです。ゆえに「完全」を実現することは困難ではありますが、チェックリストの活用により、開発されるソフトウェアのセキュリティ向上を目指したいと考えております。

これが多くの企業、開発に従事する方々に利用され、今後ますます日本経済の発展に必要となるIT、中でもソフトウェアの安全性向上に少しでも貢献できることを切に期待しております。

一般社団法人コンピュータソフトウェア協会
セキュリティ委員会
ソフトウェア出荷判定セキュリティ基準策定ワーキンググループ
主査　小屋晋吾

※ただし、本解説書の情報は本解説書執筆時点の情報であり、ご参照いただく時点の情報とは異なる可能性もありますのでご留意ください。

2 チェックリストの使い方

1)チェックリストの並び順および項目の意味について

チェックリストはおおむね共通フレームの開発工程順に並んでいます。チェックリストの各項目では以下の説明をしています。

【対応する開発プロセス分類】

対策を行うべき主な開発プロセスを示します。詳細については後述の共通フレームによる説明を参照してください。

【対策項目】

開発にあたり、どのような対策を講じるべきかの概略を示します。

【対策詳細】

対策項目について、理解するために必要な解説を行い、また具体的な対策があればその詳細を示しています。

【脅威シナリオ】

対策を講じなかった場合、どのような脅威が発生するかを具体的なシナリオとして示しています。

* これは一例であり、ほかにも発生する可能性のある脅威があることに注意してください。

【具体的な実装例(参考文献)】

参考文献を紹介しています。

【脅威分類＜事象/結果＞および＜手段＞】

脅威によりもたらされる結果、および脅威となる手段について、いくつかのキーワードで分類を行いました。

【共通フレームにおける分類】

セキュリティへの対策が、共通フレームのどの開発工程で要求され、定義され、実施され、その結果が文書化されるかを示し、また各工程で行うべき作業の詳細を示しました。詳細については前述の共通フレームによる説明を参照してください。

2）共通フレーム／開発プロセスについての解説

セキュリティ対策は、ソフトウェア開発の特定の工程でのみ実施すればよいという性格のものではありません。最上流の要件定義に始まり、運用・保守に至るまでのライフサイクル全体での対応が求められます。
チェックリスト中、「対策項目」より右の欄は、当該対策項目に関して、開発プロセス中の各工程で推奨される実施項目を挙げているものです。
「脅威分類＜手段＞」より右の欄は、ライフサイクル全体でのセキュリティ対策を考える際の参考として、個々の対策項目について、各工程でどういうことをすべきかを記載したものです。

本チェックリストに記載の開発プロセスは、下記文献を参考にしました。

- 『共通フレーム 2013 ～経営者、業務部門とともに取組む「使える」システムの実現～』
- 監修・発行 独立行政法人 情報処理推進機構、ISBN:978-4-905318-19-4

昨今では、組織の開発標準としてソフトウェア開発プロセスを定義し、運用している開発組織が多いと思われます。ただ、組織のプロセス定義や開発標準で用いられる工程の捉え方や工程名称は、それぞれの開発組織の考え方や文化的背景が大きく反映されています。その結果、異なる開発組織のプロセスを取り上げてみると、工程名称が同じでも実施事項が異なったり、同じ事項を実施する工程が異なっていたり、ということがよくあります。また、アジャイル開発プロセスのように、ウォーターフォール型のプロセスとはそもそも考え方の異なるプロセスを採用する開発組織も増えてきています。

本チェックリストは、さまざまな開発組織で使っていただくことを想定しています。そのため、汎用的なソフトウェア開発プロセスとして「共通フレーム」中の「テクニカルプロセス」を取り上げて、各工程の実施内容を記述することとしました。

下表（表2-1）に、「共通フレーム：テクニカルプロセス」における各工程の概略説明を記します（参考：前掲書）。チェックリスト「対策項目」より右の欄を見る際は、見出しにある工程を、読者の組織における工程に読み替えてください。

表 2-1

工程 （チェックリストの見出し）	説明 ※丸括弧内は、対応する共通フレームでのプロセス名称
企画	（企画プロセス） 目的、目標を達成するために必要なシステムに関する要件の集合とシステム化の方針、および、システムを実現するための実施計画を構想する。
要件定義	（要件定義プロセス） 利用者および他の利害関係者が必要とするサービスを提供できるよう、システムに対する要件を定義する。
ソフトウェア要件定義	（ソフトウェア要件定義プロセス） システム中のソフトウェア構成要素に関わる要件を定義する。
方式設計	（ソフトウェア方式設計プロセス） ソフトウェア要件の検証可能な実現方式を検討する（ソフトウェアのアーキテクチャや構成、構成要素間のインターフェイス等）。
詳細設計（コーディング）	（ソフトウェア詳細設計プロセス） ソフトウェア要件やソフトウェア実現方式に対し、コーディングやテストが実施できるよう詳細を検討する。コーディング作業はこの工程に含まれる。
構築（単体ビルド）	（ソフトウェア構築プロセス） ソフトウェア設計を適切に反映した、実行可能なソフトウェア構成単位を作成する（モジュール、ライブラリ、パッケージ等）。ソフトウェア構成単位のテスト（単体テスト）はこの工程に含まれる。
結合	（ソフトウェア結合プロセス） ソフトウェア構成単位を組合せ、結合されたソフトウェア構成要素を作成する。結合テストはこの工程に含まれる。
適格性確認テスト	（ソフトウェア適格性確認テストプロセス） 結合されたソフトウェア製品が、要件を満たしていることをテストにより確認する。いわゆる（ソフトウェア構成要素ごとの、またソフトウェア全体の）システムテスト、総合テストはこの工程に該当する。
運用	（運用プロセス） 意図された環境でソフトウェア製品を運用する。また、システムおよびソフトウェア製品の顧客に対する支援を提供する。
支援	（支援プロセス） ソフトウェア開発プロセスの実行に関連し、プロジェクトの成功及び品質に寄与する次のプロセスをいう。文書化管理、品質保証、検証、妥当性確認、共同レビュー、監査、問題解決

3）職務別の使い方

- ①共通の認識を持つためソフトウェア開発に携わる関係者（企画・開発・品質保証・テクニカルサポート）はチェックリストの内容を一読してください。

- ②どの項目を適用するかは、該当する製品/サービスに必要なセキュリティレベルを考慮して、関係部署それぞれ視点で必要な項目のリストアップを行い、合意を取るようにしてください。

  - a.企画段階：企画・要件定義段階で要求事項としてリストアップしてください。当たり前として要求しないと後工程で漏れる可能性があります（特に開発が別会社の場合）。
  - b.開発段階：企画・要件定義より技術的に必要と思われる項目をリストアップしてください。また、どのプロセスで行うべきかの指針を共通フレームワークベースで示しています。
  - c.品質保証段階：品質保証の視点から必要と思われる項目をリストアップしてください。また、リストアップされた項目の適格性確認テストが実施されていることを確認してください。
  - d.テクニカルサポート段階：予防的見地から必要と思われる項目をリストアップしてください。

- ③該当する製品/サービスの出荷判断時に本チェックリストでリストアップした項目についての試験結果を確認し、リリースの判断項目の一助としてください。

4）別表「ソフトウェアのセキュリティ」について

近年、インターネットが生活や業務に欠かせないインフラの一部になり、それに伴いソフトウェアの品質保証における脆弱性対策が重要な位置を占めるようになってきています。一方で、脆弱性対策が品質の要求事項として、企画・要件定義段階で明確に定義されているケースは少なく、脆弱性対策は技術者の「暗黙の前提」、「当然の配慮」とする開発体制を多数見受けます。
しかし、既知の脆弱性は明確に対策を要件として定義でき、（定義されていない要件が実装に反映されないのは当然とすれば）インシデント発生の原因を単なるコーディングミスとすべきではないと考えられます。つまり、脆弱性を許容してしまった一連のソフトウェア制作プロセスに「内在する瑕疵」として位置づけ、原因究明と再発防止の体制が検討されるべきです。

他方、脆弱性を作り込まないマネジメント体制という観点での議論は少なく、多くの開発者が手探りで脆弱性対策という品質の作り込みに追われているのも事実といえます。そこで、当WGでは技術的な対策手法を例示するだけでなく、管理的な対策を例示することで、脆弱性を作り込まない組織・マネジメント体制に寄与するドキュメントの作成を試みました。

別表「ソフトウェアのセキュリティ」は、経済産業省が2015年に発表したスマートメーター制度検討会のセキュリティ検討ワーキンググループ報告書「別添統一的なガイドラインの標準対策要件に盛り込むべき事項」を参考に、当WGが独自に改変したものです。
セキュリティ仕様、ソフトウェア取得、アップデート、認証の項目に分かれており、一見、ごく常識的な文書化やチェックを求めるものでしかありませんが、実際に、このような項目に従って、脆弱性を作り込まない組織体制を維持するには、一定のコストとチームメンバーの見識の高さが求められるといえます。協力パートナーを含めた体制の見直しの一助として参考いただければ幸いです。

- 「スマートメーター制度検討会セキュリティ検討ワーキンググループ」報告書別添(PDF形式：324KB)
- http://www.meti.go.jp/press/2015/07/20150710001/20150710001-3.pdf

3 参考文献

■IPAセキュアプログラミング講座Webアプリケーション編第5章暴露対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/web.html

■DDoS攻撃のソリューション(ARBOR)
http://jp.arbornetworks.com/ddos%E6%94%BB%E6%92%83%E9%98%B2%E5%BE%A1/

■DDOS 攻撃を撃退するには(Akamai)
https://www.akamai.com/jp/ja/resources/protect-against-ddos-attacks.jsp

■チーム開発実践入門（技術評論社）
p39 理想的なプロジェクトとは（PDF)

■Gitが、おもしろいほどわかる基本の使い方33(MdN)
p12 Gitを使ったバージョン管理（PDF)

■RFC2827 （IPA 独立行政法人 情報処理推進機構）
https://www.ipa.go.jp/security/rfc/RFC2827JA.html

■RFC 3704 （IPA 独立行政法人 情報処理推進機構）
https://www.ipa.go.jp/security/rfc/RFC3704JA.html

■イングレスフィルタリングとは（日本ネットワークインフォメーションセンター）
https://www.nic.ad.jp/ja/basics/terms/ingress-filtering.html

■IPAセキュアプログラミング講座 > C/C++言語編 > 脆弱性回避策とソフトウェア開発工程 > 脅威モデリング
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/c101.html

■いまさら聞けないサーバー証明書(サイバートラスト)
https://www.cybertrust.ne.jp/sureserver/basics/ssl_movie.html

■暗号技術入門第3版（SBクリエイティブ）
p364 第14章SSL/TLSセキュアな通信のために

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p441盗聴・改ざん対策（PDF)

■.NETで難読化を試してみる（とある技術者の劣等感）
http://ouranos.sakura.ne.jp/wordpress/2012/05/17/net%E3%81%A7%E9%9B%A3%E8%AA%AD%E5%8C%96%E3%82%92%E8%A9%A6%E3%81%97%E3%81%A6%E3%81%BF%E3%82%8B-%E7%AC%AC1%E5%9B%9E/

■不正な解析から知的財産を守る.NETアプリ「難読化」再入門 （CodeZine）
https://codezine.jp/article/detail/5444

■オープンソースソフトウェアを利用した出荷済製品のセキュリティ確保の活動（富士通）
https://www.fujitsu.com/jp/documents/about/resources/reports/securityreport/2015-securityreports/security-2015-08.pdf

■IPAテクニカルウォッチ「ウェブサイトにおける脆弱性検査手法の紹介（ソースプログラム検査編）」
https://www.ipa.go.jp/security/technicalwatch/20140306.html

■IPA MyJVN（フィルタリング条件設定）
http://jvndb.jvn.jp/apis/myjvn/mjcheck.html

■SLAの考え方 - 総務省
http://www.soumu.go.jp/main_sosiki/joho_tsusin/top/local_support/pdf/cio_text18_t_18.pdf

■情報システムに係る政府調達へのＳＬＡガイドライン
（独立行政法人情報処理推進機構、平成16年）
http://www.meti.go.jp/policy/it_policy/tyoutatu/sla-guideline.pdf

■Web Application Firewall （WAF）読本(IPA)
https://www.ipa.go.jp/files/000017312.pdf

■ウェブサイトにキャプチャを導入する方法【reCAPTCHAの使い方】(Syncer)
https://syncer.jp/how-to-introduction-recaptcha

■reCAPTCHA: Easy on Humans, Hard on Bot
https://www.google.com/recaptcha/intro/index.html

■インフラ/ネットワークエンジニアのためのネットワーク技術＆設計入門（SBクリエイティブ）
p381 Syslogで障害を検知する

■IPA コンピュータセキュリティログ管理ガイド（NIST SP800-92）
https://www.ipa.go.jp/files/000025363.pdf

■IPAセキュアプログラミング講座 C/C++言語編 > 不測の事態対策 > ログ記録による証跡確保とログ自体の漏えい対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/c301.html

■インフラ/ネットワークエンジニアのためのネットワーク技術＆設計入門（SBクリエイティブ）
p340 高可用性設計

■暗号サービス（MSDN）
https://msdn.microsoft.com/ja-jp/library/92f9ye3s(v=vs.110).aspx

■RedHat Enterprise Linux セキュリティガイド 第3章 暗号化
https://access.redhat.com/documentation/ja-JP/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-Security_Guide-Encryption.html

■暗号技術入門 第3版（SBクリエイティブ）
p299 パスワードを元にした暗号

■経済産業省 「スマートメーター制度検討会セキュリティ検討ワーキンググループ」報告書別添「統一的なガイドラインの標準対策要件に盛り込むべき事項」
http://www.meti.go.jp/press/2015/07/20150710001/20150710001-3.pdf

■IPA NIST Special Publication 800-63. Version 1.0.2. 電子認証に関するガイドライン
https://www.ipa.go.jp/files/000025342.pdf

■オンライン本人認証方式の実態調査報告書
https://www.ipa.go.jp/files/000040778.pdf

■IPA セキュアプログラミング講座 > C/C++言語編 > 著名な脆弱性対策 > フォーマット文字列攻撃対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/c906.html

■IPAセキュアプログラミング講座 > C/C++言語編 > 著名な脆弱性対策 > バッファオーバーフロー
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/c901.html

■IPAセキュアプログラミング講座 > C/C++言語編 > ファイル対策 > シンボリックリンク攻撃対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contents/c802.html

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p119 SQL呼び出しに伴う脆弱性 （PDF エラーメッセージからの情報漏えい）

■HTTPの教科書（翔泳社）
p264 SQLインジェクション〃

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p160 推測可能なセッションID
p164 URL埋め込みのセッションID
p171 セッションIDの固定化〃

■インジェクション攻撃 OSコマンドインジェクション （ThinkIT）
https://thinkit.co.jp/cert/tech/7/5/4.htm

■SSIインジェクション（Web Application Security Consortium:脅威の分類）
http://projects.webappsec.org/f/WASC_TC-1.0.jpn.pdf

■攻撃者が”嫌う”セキュリティ対策とは何か？（yohgaki's blog）
http://blog.ohgaki.net/software-defense-attacker-hates

■安全なWebアプリケーションの作り方 （SBクリエイティブ）
p68 Webアプリケーションの機能と脆弱性の対応

■インジェクション攻撃 OSコマンドインジェクション （ThinkIT）
https://thinkit.co.jp/cert/tech/7/5/4.htm

■SSIインジェクション（Web Application Security Consortium:脅威の分類）
http://projects.webappsec.org/f/WASC_TC-1.0.jpn.pdf

■攻撃者が”嫌う”セキュリティ対策とは何か？（yohgaki's blog）
http://blog.ohgaki.net/software-defense-attacker-hates

■IPA セキュアプログラミング講座 > Webアプリケーション編 > 入力対策 > コマンド注入攻撃対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/web.html

■RFC 2616　14.38 Server
Note: Revealing the specific software version of the server might allow the server machine to become more vulnerable to attacks against software that is known to contain security holes. Server implementors are encouraged to make this field a configurable option.
注意: サーバのソフトウェアバージョンを明らかにする事で、セキュリティホールを持っている事がわかっているソフトウェアを使うサーバのマシンは攻撃を受けやすくなるかもしれない。サーバの開発者は、このフィールドをオプションとして設定を変更できるようにする事が推奨される。
http://www.spencernetwork.org/reference/rfc2616-ja-HTTP1.1.txt

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p118 エラーメッセージからの情報漏えい

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p313 パスワードに関するアプリケーション要件

■本当は怖いパスワードの話（@IT）
https://www.atmarkit.co.jp/ait/articles/1110/06/news154.html

■オンライン手続におけるリスク評価及び電子署名・認証ガイドライン（各府省情報化統括責任者（CIO）連絡会議）
https://www.kantei.go.jp/jp/singi/it2/guide/guide_line/guideline100831.pdf

■HTTPの教科書 （翔泳社）
p283 不適切なエラーメッセージ処理

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p119 SQL呼び出しに伴う脆弱性

■HTTPの教科書（翔泳社）
p264 SQLインジェクション

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p446 frame、iframeを使わない
p434 なりすまし対策

■知らぬ間にプライバシー情報の非公開設定を公開設定に変更されてしまうなどの「クリックジャッキング」に関するレポート
https://www.ipa.go.jp/files/000026479.pdf

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p88 クロスサイト・スクリプティング

■HTTPの教科書（翔泳社）
p258 クロスサイト・スクリプティング

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p141 クロスサイト・リクエストフォージェリ

■IPA セキュアプログラミング講座 > Webアプリケーション編 > セッション対策 > リクエスト強要（CSRF）対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/web.html

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p232 ファイルアクセスにまつわる問題

■IPA セキュアプログラミング講座 > Webアプリケーション編 > 暴露対策 > Webサーバからのファイル流出対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/web.html

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p425 Webサーバーへの攻撃経路と対策

■IPA セキュアプログラミング講座 >Webアプリケーション編 > 開発工程と脆弱性対策
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/web.html
https://www.ipa.go.jp/security/awareness/vendor/programmingv2/contets/w002_img.html

■Coverity Security Advisor
https://www.coverity.com/

■Fortify SCA(HP Fortify Static Code Analyzer)
https://www.fortify.com/products/hpfssc/source-code-analyzer.html

■IBM Security AppScan
https://www.ibm.com/software/products/ja/appscan

■Valgrind
https://valgrind.org/

■安全なWebアプリケーションの作り方（SBクリエイティブ）
p448 マルウェア対策

■感染チェックツールのご紹介（ACTIVE）オンラインスキャンによるクロスチェック
https://www.active.go.jp/security/flow/complete.html

■マイクロソフト セキュリティアドバイザリ SSL 3.0 の脆弱性により、情報漏えいが起こる
https://technet.microsoft.com/ja-jp/library/security/3009008.aspx

■redhat httpd における POODLE SSLv3.0 脆弱性問題の解決方法
https://access.redhat.com/ja/solutions/1232613

■JVNDB-2013-005768 NTP の ntpd の ntp_request.c 内の monlist 機能におけるサービス運用妨害（DoS）の脆弱性
http://jvndb.jvn.jp/ja/contents/2013/JVNDB-2013-005768.html

■JPNIC オープンリゾルバ(Open Resolver)に対する注意喚起
https://www.nic.ad.jp/ja/dns/openresolver/

■国立研究開発法人情報通信研究機構 日本標準時G
https://www2.nict.go.jp/aeri/sts/tsp/PubNtp/

■インターネットマルチフィード 時刻情報提供サービス for Public
http://www.jst.mfeed.ad.jp/service/02.html

■難読化していないAndroidアプリケーションは脆弱性か （徳丸浩の日記）
http://blog.tokumaru.org/2012/02/is-obfuscation-of-android-application.html

■ロケールの影響を受ける動作（Oracle）
https://docs.oracle.com/cd/E26924_01/html/E27144/glmde.html

■言語パックとは（Microsoft）
https://technet.microsoft.com/ja-jp/library/cc766472(v=ws.10).aspx

■Windows, Inernet Explorerセキュリティのいま（Microsoft）
http://www.jnsa.org/seminar/pki-day/2015/data/2-2_muraki.pdf

■TLS/SSL の設定（Microsoft）
https://msdn.microsoft.com/ja-jp/library/dn786418(v=ws.11).aspx

■SSL 3.0 の脆弱性により、情報漏えいが起こる
https://technet.microsoft.com/ja-jp/library/security/3009008.aspx

■暗号技術入門 第3版（SBクリエイティブ）
p241 デジタル署名の利用例

■EVプログラム署名の必要性（サイバートラスト）
https://www.cybertrust.ne.jp/digicert-ev-code-signing/about-codesigning.html

■Availability and description of the File Checksum Integrity Verifier utility（MS）
https://support.microsoft.com/en-us/kb/841290

■CRYPTREC の暗号アルゴリズム仕様書について
https://www.cryptrec.go.jp/report/c13_kentou_gijj01_5.pdf

■FIPS PUB 180-4　Secure Hash Standard　(SHS)
http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

■脆弱性対策情報データベース JVN iPedia
http://jvndb.jvn.jp/

■MyJVN 脆弱性対策情報収集ツール
http://jvndb.jvn.jp/apis/myjvn/sysad.html

■MyJVN API とは
http://jvndb.jvn.jp/apis/index.html

■IPA 「システム・リファレンス・マニュアル（ＳＲＭ)」の作成（経営目標実現のためのＩＴ課題解決へのヒント）保守・運用編
https://www.ipa.go.jp/about/jigyoseika/04fy-pro/chosa/srm/srm4.pdf
https://www.ipa.go.jp/about/jigyoseika/05fy-pro/chosa/2005-srm2.pdf

4 終わりに

ソフトウェアの出荷基準について、一般的に取られている対策についてまとめました。
この出荷基準を活用いただき、CSAJ会員企業で出荷されるソフトウェアの品質が高まり、ユーザーから信頼されることを期待いたしております。

セキュリティリスクは、ソフトウェアに対策を施してもハードウェア、ネットワーク機器や、ネットワーク構成などに問題がある場合は、無意味となります。攻撃者は、一番対策があまい個所を狙って攻撃をするものなので、全般的な対応が必要となります。

セキュリティ対策には、完全な対策はありません。完全な対策を追求するとコストが膨らみ過ぎ、ユーザーに過剰なコストを要求することになります。保全される利益とコストのバランスを考え、取るべき対策を検討していくことが望まれます。

そのためには、日ごろからセキュリティに関する知識・情報を確認し知識をアップデートする必要があります。本書がその一助になればと考えております。

5 ソフトウェア出荷判定セキュリティ基準策定ワーキンググループ

主　査★小屋　晋吾（トレンドマイクロ株式会社　執行役員 統合政策担当／セキュリティ委員会副委員長）

メンバ★三舛畑　達（アクセルユニバース株式会社　システム部 マネージャー）
★板東　直樹（アップデートテクノロジー株式会社　代表取締役社長）
★福嶌　淳也（オー・エイ・エス株式会社　製品開発ソリューション部 課長）
★伊藤　裕紀（サイバートラスト株式会社　品質保証本部）
松尾　武俊（サイバートラスト株式会社　営業本部長）
★明尾　洋一（サイボウズ株式会社　グローバル開発本部 品質保証部 部長）
唐澤　正樹（株式会社チェプロ　サポート統括）
★太田　浩二（トレンドマイクロ株式会社　プロダクトマーケティング本部）
★望月　信昭（日本ナレッジ株式会社）
★上符　仁司（ピー・シー・エー株式会社　品質管理本部　システム検証部 次長）
★高田　寿久（株式会社フォーラムエイト　システム開発Group長）
★太田　猛　（株式会社ユビキタス　研究開発本部　品質保証グループ）
★太田　純　（リコージャパン株式会社 ICT技術本部 ドキュメントSIセンター ソリューション開発室 ）

事務局　戸島　拓生（一般社団法人コンピュータソフトウェア協会　業務課）
鈴木　啓紹（一般社団法人コンピュータソフトウェア協会　業務課）

（社名五十音順、敬称略、★は執筆メンバ）

◆意見募集協力企業
・JBアドバンスト・テクノロジー株式会社
・ネクストウェア株式会社
・弥生株式会社

● ワーキンググループでは今後もチェックリスト及び解説書の更新作業を検討しております。更新時の改訂作業には広く知見を必要としておりますので会員企業からのご協力をお願い致します。

項目	Ver	日付	備考
ソフトウェア出荷判定セキュリティ基準チェックリスト-解説書-	1.0.0	2016/07/13	

本解説書の対応チェックリストver：1.0.0

ソフトウェア出荷判定セキュリティ基準
チェックリスト解説書

2016年7月13日　第1.0.0版

Computer Software Association of Japan
一般社団法人コンピュータソフトウェア協会

〒107-0052
東京都港区赤坂1-3-6
赤坂グレースビル4階
TEL：03-3560-8440
FAX：03-3560-8441
http://www.csaj.jp/