情報技術
セキュリティ評価のための
コモンクライテリア

パート 3: セキュリティ保証コンポーネント

2022 年 11 月

CC:2022
改訂第 1 版

CCMB-2022-11-003

令和 5 年 9 月 翻訳第 1.0 版
独立行政法人情報処理推進機構
セキュリティセンター
セキュリティ技術評価部

目次

IPA まえがき

本書は、「IT セキュリティ評価及び認証制度」において、「認証機関が公開する評価基準」の規格として公開している Common Criteria(以下、CC という)を翻訳した文書である。

原文

Common Criteria for Information Technology Security Evaluation

Part3: Security assurance components CC:2022 Revision 1

November 2022 CCMB-2022-11-003

まえがき

本バージョンは、2017 年に CC v3.1 改訂第 5 版として発行されて以来、最初の大幅改訂となる「情報技術セキュリティ評価のためのコモンクライテリア」（CC:2022）である。

歴史的に、CC 標準は共通評価方法（CEM）とともに、IT セキュリティ分野におけるコモンクライテリア認証書の承認に関する協定（CCRA）の参加国によって開発・維持され、その後、ISO（国際標準化機構）及び IEC（国際電気標準会議）が維持する標準として公表されてきた。しかし、CC:2022 と CEM:2022 は、まず ISO/IEC 標準として開発され、その後、CCRA により CC と CEM の新バージョンとして発行されたものである。CC:2022 の ISO 版は ISO/IEC 15408-1:2022～15408-5:2022 として 5 パートで発行され、CEM:2022 の ISO 版は ISO/IEC 18045:2022 として 1 パートで発行されている。

CC:2022 は、以下のパートから構成されている。

— パート1：概説と一般モデル

— パート2：セキュリティ機能コンポーネント

— パート3：セキュリティ保証コンポーネント

— パート4（新規）：評価方法及び評価アクティビティの仕様のための枠組み

— パート5（新規）：セキュリティ要件の定義済みパッケージ

CC:2022 は、CC v3.1 が発行されて以来用いられてきた標準の新しい使用方法を、正式に規定することを目的としている。CC v3.1 が発行されて以来、新しい保証パラダイムが開発され、附属書や補遺として標準に追加されてきた。これには、評価が適合主張の範囲を超えることを禁止する完全適合の概念や、個々のセキュリティ機能を評価するために、評価アクティビティを使用して、機能に特化した保証や客観性のあるガイドラインを提供するという概念が含まれる。また、標準の前回の大幅な改訂以降、重要性が増した機能要件の形式化も含まれている。CC:2022 の発行は、これらの開発を標準そのものに完全に統合する。

CC:2022 には、新しい ISO/IEC 15408:2022 標準の編集中に提供されたパート 4 とパート 5 が CC の新しいオリジナルパートとして含まれていることを強調する価値がある。これらは、旧版 CC v3.1 R5 を大幅に強化する。パート 5 は、CC v.3.1 改訂第 5 版のパート 3 の関連する節に基づいている。

CC:2022 は、次のような具体的な変更点を取り入れている。

— 文書が再構成され、新たなパートが追加された。

  — パート4：評価方法及び評価アクティビティの仕様の方法を定義している。

  — パート5：事前に定義された保証パッケージを列挙したもので、このバージョンで新たに導入されたものもある。

— 以下の技術的な変更が導入された。

  — 用語が見直され、更新された。

— 新しい機能要件及び新しい保証要件が導入された。

— 完全適合の種別が導入された。

— 低保証のプロテクションプロファイル(PP)が削除され、直接根拠PPが導入された。

— マルチ保証評価が導入された。

— 保証の統合が導入された。

CCの全てのパートはCommon Criteria Portal (www.commoncriteriaportal.org)で見ることができる。

本書で使用されている商標は、利用者の便宜を図るための参考情報であり、推奨を意味するものではない。

法定通知

情報技術セキュリティ評価のためのコモンクライテリアの本バージョンの開発には、以下に示す政府機関が貢献した。ISO/IEC とともに、情報技術セキュリティ評価のためのコモンクライテリア、バージョン 2022 パート 1 からパート 5（「CC:2022」と呼ぶ）の著作権の共同保有者として、これらの政府機関はここに、ISO/IEC 15408 及びその派生版（それらの国での採用を含む）の改訂版において ISO/IEC に CC:2022 を複製する非排他的許可を与える。ただし、CC:2022 を適切な方法で使用、複製、配布、翻訳、変更する権利は、これらの政府機関が保有する。ISO/IEC はその見返りとして、前述の政府機関に対し、成果物である CC:2022 パート 1 からパート 5 を、彼らが適切と考えるライセンスで使用することを許可する。前述の政府機関は、文書の一部分の修正や再利用を含め、文書の利用者がテキストを再利用することを常に支援しており、今後もこの方針に従う予定である。

オーストラリア	The Australian Signals Directorate
カナダ	Communications Security Establishment
フランス	Agence Nationale de la Sécurité des Systèmes d'Information
ドイツ	Bundesamt für Sicherheit in der Informationstechnik
日本	独立行政法人情報処理推進機構(Information-technology Promotion Agency)
オランダ	Netherlands National Communications Security Agency
ニュージーランド	Government Communications Security Bureau
韓国	National Security Research Institute
スペイン	Ministerio de Asuntos Económicos y Transformación Digital and Centro Criptológico Nacional
スウェーデン	FMV, Swedish Defence Materiel Administration
英国	National Cyber Security Centre
米国	The National Security Agency and the National Institute of Standards and Technology

序説

この文書に定義されているセキュリティ保証コンポーネントは、セキュリティ保証パッケージ、プロテクションプロファイル(PP)、PPモジュール、PP構成又はセキュリティターゲット(ST)に表されているセキュリティ保証要件に対する基礎である。

これらの要件は、TOE保証要件を表現する標準的な手段を規定している。この文書は、保証コンポーネント、保証ファミリ、及び保証クラスのセットをカタログ化している。また、PP、PP構成、PPモジュール及びSTの評価基準も定義している。

この文書の対象読者には、セキュアなIT製品の消費者、開発者、評価者などが含まれる。CCパート1の5章は、CCの対象読者及び対象読者からなるグループによる標準の使用についての追加情報を提供している。これらのグループは、この文書を次のように使うことができる:

a) 消費者は、PP又はSTに記述されているセキュリティ対策方針を達成するための保証要件を表すコンポーネントを選択する際に、この文書を使用して、TOEで要求されるセキュリティ保証レベルを決定する。

b) 開発者は、TOEを構成するときに実際の又は認識された消費者のセキュリティ要件に応じ、TOEの保証要件のステートメントを解釈する際、及びTOEの保証アプローチを決定する際にこのCCパート3を参照する。

c) 評価者は、TOEの保証を確定する際、及びPPとSTを評価する際に、不可欠な評価基準のステートメントとして、この文書で定義されている保証要件を使用する。

注：この文書では、用語を他のテキストと区別するために、ボールドやイタリック体を使用している場合がある。ファミリ内のコンポーネント間の関係は、ボールド表記を用いて強調表示される。この表記では、全ての新しい要件をボールドで表示する必要がある。階層型のコンポーネントでは、前のコンポーネントの要件を超えて強化又は変更されたとき、要件がボールドで表示される。また、前のコンポーネントを超えて許可される新しい操作又は拡張操作も、ボールドで強調表示される。

イタリック体の使用は、正確な意味を持つテキストであることを示す。セキュリティ保証要件では、この表記は評価に関連する特別な動詞に使用される。

情報技術セキュリティ評価のためのコモンクライテリア ― パート3：セキュリティ保証コンポーネント

1　適用範囲

この文書は、CCの保証要件を定義している。ここには、CCパート5に含まれる保証レベルとパッケージを構成する個々の保証コンポーネント、及びプロテクションプロファイル(PP)、PP構成、PPモジュール及びセキュリティターゲット(ST)の評価基準が含まれている。

2 規定の参照

以下の文書は、その内容の一部又は全部が本書の要求事項となるように本文中で参照されている。日付の付いている参照資料については、指定した版のみが適用される。日付のない参照資料については、(修正を含む)最新版の参照文書が適用される。

情報技術セキュリティ評価のためのコモンクライテリア、CC:2022、改訂第1版、2022年11月 — パート1: 概説と一般モデル

情報技術セキュリティ評価のためのコモンクライテリア、CC:2022、改訂第1版、2022年11月 — パート2: セキュリティ機能コンポーネント

情報技術セキュリティ評価のためのコモンクライテリア、CC:2022、改訂第1版、2022年11月 — パート4: 評価方法とアクティビティの仕様フレームワーク

情報技術セキュリティ評価のためのコモンクライテリア、CC:2022、改訂第1版、2022年11月 — パート5: セキュリティ要件の事前定義パッケージ

情報技術セキュリティ評価のための共通方法、CEM:2022、改訂第1版、2022年11月 — 評価方法

ISO/IEC IEEE 24765, *Systems and software engineering — Vocabulary*

3 用語と定義

本文書の目的のために、CCパート1、CCパート2、CCパート4、CCパート5、CEM、ISO/IEC IEEE 24765及び以下で使用された用語及び定義を適用する。

ISOとIECは、標準化で使用する用語データベースを以下のアドレスで管理している。

— ISO Online browsing platform: https://www.iso.org/obp

— IEC Electropedia: https://www.electropedia.org/

3.1
受入れ手続き(acceptance procedure)
新たに作成又は変更された*構成要素*(3.3)を評価対象(TOE)の一部として受け入れるか、又はそれらをライフサイクルの次のステップに移すために実行される手続き。

注1：これらの手続きによって、受け入れ責任のある役割又は個人、及び受け入れを決定するために適用される基準が識別される。

注2：受入れ状況にはいくつかのタイプがあり、その一部は重複してもよい。

a) 特に統合プロセスの一部として、構成管理システムへの要素の最初の受け入れ。

b) TOEの構成の各段階で、構成要素の次のライフサイクルフェーズへの進行。

例：モジュール、サブシステム、完成したTOEの品質管理[i]

c) 構成要素の転送後。

例：異なる*開発*(3.15)サイト間でのTOE又は準備製品の部分[ii]

d) 消費者へのTOEの*配付*(3.14)後。

e) TOEの統合後。

例：他の製造者のソフトウェア、ファームウェア、及びハードウェアコンポーネントのTOEへの組み込み[iii]

3.2
アクション(action)
CCパート3の評価者又は開発者のアクションエレメント。

注1：これらのアクションは、評価者アクションとして明示的に記述されているか、又はCCパート3の保証コンポーネント内の開発者アクション(暗黙の評価者アクション)から暗黙的に導き出される。

3.3
構成要素(configuration item)
評価対象(TOE)*開発*(3.15)中に、構成管理のために指定され、構成管理プロセスにおいて単一のエンティティとして扱われる、ハードウェア、ソフトウェア、又はその両方の項目又は集合。

注1：これらは、TOEの一部又はTOEの開発に関連するオブジェクト、例えば評価証拠資料や開発ツール、かもしれない。構成管理要素は、構成管理システムに直接格納されるか(例えばファイル)、又はそれらのバージョンと共に参照によって格納されてもよい(例えばハードウェア部品)。

3.4
構成リスト(configuration list)
特定の製品の全ての*構成要素*(3.3)をリストする*構成管理出力*(3.8)文書。完全な製品の特定バージョンに関連する各構成管理要素の正確なバージョンを伴う。

注1：このリストによって、製品の評価済みバージョンに属する要素と、製品の別のバージョンに属するその要素の別バージョンとを区別することが可能となる。最終的な構成リスト[iv]は、特定の製品の特定バージョンに対する固有の文書である(このリストは*構成管理ツール*(3.12)内の電子文書にすることができる。その場合は、システムの出力ではなく、システムの特定のビュー又はシステムの一部として参照できる。ただし、実際に評価で使用される場合は、おそらく評価証拠資料の一部として構成リストが配付される。)構成リストは、ALC_CMCの構成管理要件下にある要素を定義する。

3.5
構成管理(configuration management)
CM
以下の技術、管理上の指針及び調査に適用される原則:*構成要素*(3.3)の機能的及び物理的特性を識別して文書化する、それらの特性の変更を制御する、変更処理及び実施状況を記録及び報告する、特定の要求への適合を検証する。

[出典： ISO/IEC IEEE 24765:2017, 3.779 1]

3.6
構成管理証拠資料(configuration management documentation)
CM証拠資料(CM documentation)
構成管理出力(3.8)、*構成リスト*(3.4)[iv]、*構成管理システム記録*(3.11)、*構成管理計画*(3.9)及び*構成管理用法証拠資料*(3.13)を含む証拠資料。

3.7
構成管理証拠(configuration management evidence)
構成管理システムの正しい運用を確信するために使用される全てのもの。

例：*構成管理出力*(3.8)、開発者が提供する根拠、評価者がサイト訪問中に行った観察、実験又はインタビュー。

3.8
構成管理出力(configuration management output)
構成管理システムによって生成又は実施された、構成管理に関連する結果。

注1：これらの構成管理関連結果は、文書(例えば、データが出力された用紙、*構成管理システム記録*(3.11)、ロギングデータ、ハードコピー、電子出力データ)、及びアクション(例えば、構成管理の指示を満たすための手動による措置)として発生する。このような構成管理出力の例には、*構成リスト*(3.4)、*構成管理計画*(3.9)、及び/又は製品ライフサイクルの間のふるまいがある。

3.9
構成管理計画(configuration management plan)
評価対象(TOE)に対し構成管理システムがどのように使用されるかの記述。

注1：構成管理計画を発行する目的は、スタッフメンバがそれぞれの責務を明確に把握できるようにすることである。構成管理システム全体の観点では、構成管理計画を出力文書とみなすことができる(構成管理システムのアプリケーションの一部として生成することができるため)。具体的なプロジェクトの観点では、構成管理計画は用法文書である。なぜなら、プロジェクトチームのメンバが、プロジェクトの期間中に実行しなければならないステップを理解するために使用するからである。構成管

理計画は、特定の製品に対するシステムの用法を定義する。別の製品に対して、同じシステムが異なる範囲で用いられてもよい。構成管理計画は、TOEの*開発*(3.15)中に使用される会社の構成管理システムの出力を定義し、記述する。

例：構成管理計画の構造及び内容は、ISO 10007:2017の附属書Aに示されている。

3.10
構成管理システム(configuration management system)
製品のライフサイクルにおいて、開発者がその製品の設定を開発及び保守するために使用する手続きとツールのセット(それらの証拠資料も含む)。

注1：構成管理システムでは、厳格性の度合い及び機能は様々である。上位レベルでは、構成管理システムで欠陥修正、変更管理、及びその他の追跡メカニズムを自動化することができる。

3.11
構成管理システム記録(configuration management system record)
重要な構成管理アクティビティを文書化する構成管理システムの運用中に生成される出力。

例：構成管理要素変更管理用紙及び構成管理要素アクセス許可用紙。

3.12
構成管理ツール(configuration management tool)
構成管理システムを実現又はサポートする手動操作の、又は自動化されたツール。

例：評価対象(TOE)の部分のバージョンを管理するツール。

3.13
構成管理用法証拠資料(configuration management usage documentation)
例えば、ハンドブック、規則、及び/又はツールと手続きの証拠資料などを使用して、構成管理システムがどのように定義され、適用されるかを記述する構成管理システムの一部。

3.14
配付(delivery)
完成した評価対象(TOE)の*製造*(3.24)環境から顧客の下への移送。

注1：この製品ライフサイクルのフェーズには、*開発*(3.15)サイトでのパッケージングと保管を含むことができるが、未完成のTOEやTOEの部分を異なる開発者間又は異なる開発サイト間で移送する処理は含まれない。

3.15
開発(development)
評価対象(TOE)の実装表現の生成に関係する製品ライフサイクルのフェーズ。

注1：ALC: ライフサイクルサポート要件全般では、開発及び関連用語(開発者、開発する)が、より一般的な意味で開発と*製造*(3.24)を含むように意図されている。

3.16
遭遇した潜在的脆弱性(encountered potential vulnerability)
評価者が評価アクティビティを実行中に識別した、セキュリティ機能要件(SFR)の侵害に使用される可能性のある評価対象(TOE)の潜在的な弱点。

3.17
評価用提供物件(evaluation deliverable)

1つ又は複数の評価又は評価監督アクティビティを実行するために評価者又は評価監督機関がスポンサー又は開発者に要求する資源。

3.18
悪用可能脆弱性(exploitable vulnerability)
評価対象(TOE)の運用環境でセキュリティ機能要件(SFR)を侵害するために使用されることがあるTOEの弱点。

3.19
設置(installation)
評価対象(TOE)をその運用環境に組み入れ、運用状態にする人間の利用者によって実行される手続き。

注1：この操作は、TOEを受領して受け入れた後に通常は1回のみ実行される。TOEはセキュリティターゲット(ST)により許可された設定にすることが期待されている。同様のプロセスを開発者が実行しなければならない場合、それらのプロセスは、ALC: ライフサイクルサポートクラス全体を通じて「生成」(generation)と表される。TOEに、定期的に繰り返す必要のない初期立ち上げが必要である場合、そのプロセスは設置として分類される。

3.20
ライフサイクルモデル(life-cycle model)
製品の*開発*(3.15)、運用、保守に関わるプロセス、活動、タスクを含む枠組みで、システムの要件定義から使用終了までのライフサイクルに関わるもの。

[出典：ISO/IEC IEEE 24765:2017 3.2219 2]

3.21
運用(operation)
<TOEライフサイクル> 評価対象(TOE)の使用フェーズで、これには、*配付*(3.14)及び*準備*(3.23)後のTOEの通常の使用、管理、及び保守が含まれる。

3.22
潜在的脆弱性(potential vulnerability)
疑われるが、確認されていない弱点。

注1：疑いは、セキュリティ機能要件(SFR)を侵害するような仮定される攻撃経路より生じる。

3.23
準備(preparation)
配付された評価対象(TOE)の顧客による受け入れと*設置*(3.19)で構成される、製品のライフサイクルフェーズにおけるアクティビティ。

注1：準備には起動、初期化、立ち上げ、運用可能な状態へのTOEの移行などを含む。

3.24
製造(production)
実装表現からTOEの実装へ、つまり顧客に*配付*(3.14)できる状態へ変換することからなるライフサイクルフェーズ。

注1：このフェーズは、TOEの製造、統合、生成、内部転送、保管、及びラベル付けで構成することができる。

3.25
残存脆弱性(residual vulnerability)
評価対象(TOE)の運用環境では悪用できないが、TOEの運用環境において予想を超える攻撃能力を持つ攻撃者が、SFRを侵害するために使用することがある弱点。

3.26
サブアクティビティ(sub-activity)
CCパート3の保証コンポーネントの適用。

注1：評価は、保証ファミリの単一の保証コンポーネントに対して行われるために、保証ファミリは、CCで明示的に取り扱われていない。

3.27
暴露期間(time period to exposure)
ある要素がITシステムに参加していて攻撃される可能性がある時間間隔。

3.28
脆弱性(vulnerability)
ある環境のセキュリティ機能要件(SFR)を侵害するために使用されることがあるTOEの弱点。

3.29
機会の期間(window of opportunity)
攻撃者が評価対象(TOE)にアクセスできる期間。

4 概要

5章では、この文書のセキュリティ保証要件で使われるパラダイムを記述している。

6章では、保証クラス、ファミリ、コンポーネント、及び評価保証レベルの提示構造とそれらの関係、及び統合保証パッケージ(CAP)の構造を記述している。また、7章から15章に記述されている保証クラスとファミリの特性も記述している。

7節から15節では、この文書の保証クラスを詳細に定義している。

附属書Aでは、開発クラスの背景にある概念について詳しく説明し、例を示している。

附属書Bでは、統合TOE評価と統合クラスの背景にある概念を説明している。

附属書Cでは、保証コンポーネント間の依存性を要約している。

5 保証のパラダイム

5.1 一般

この章の目的は、保証に対するCCのアプローチの基礎となるアプローチを文書化することである。この章を理解することにより、読者は、この文書の保証要件の合理的根拠を理解できる。

5.2 CCのアプローチ

CCのアプローチは、セキュリティへの脅威及び組織のセキュリティ方針のコミットメントを明確に表現し、提案するセキュリティ管理策が意図する目的に対して明らかに十分であることである。

そこで、脆弱性の可能性、脆弱性を実行させる能力(意図的悪用又は意図しない誘発)、及び脆弱性が実行されることにより引き起こされる損害の範囲を軽減する手段が採用されるべきである。さらに、脆弱性のその後の識別、及び脆弱性が悪用又は誘発されることの排除、緩和、及び/又は通知を容易にする手段が採用されるべきである。

5.3 保証アプローチ

5.3.1 一般

CCのアプローチは、信頼されるべきIT製品の評価に基づいて保証を提供することである。評価は、保証を提供する伝統的な手段であり、先行する評価基準書の基礎である。既存のアプローチと調和を取るために、CCは、同じアプローチを採用している。CCは、適用範囲、深さ、及び厳格性を一層強調することにより、専門の評価者による、証拠資料及び結果としてのIT製品の有効性を測定することを提案している。

CCは、保証を得るための他の手段の相対的利点を排除しておらず、またそれらについての注釈も行っていない。保証を得るための別のアプローチに関する調査が継続されている。成熟した別のアプローチがこれらの調査アクティビティから明らかになれば、それらをこのCCに含めることが検討される。現在のCCは、将来それらを取り入れることができるように構成されている。

5.3.2 脆弱性の重要性

不正利益の取得及び善意ではあるがセキュアでない行為のいずれかであれ、セキュリティ方針を侵害する機会を積極的に利用しようとする脅威エージェントが存在すると想定される。脅威エージェントは、意図せずにセキュリティの脆弱性を誘発し、組織に損害を与えることがある。機密に関わる情報を処理する必要性と、十分に信頼された製品の可用性の欠如のために、ITの障害をもたらす重大なリスクが存在する。したがって、ITセキュリティの違反が重大な損失をもたらすことがある。

ITセキュリティの違反は、ビジネスでのITの適用時に、脆弱性の意図的悪用又は意図しない誘発によって引き起こされる。

IT製品で生じる脆弱性を阻止する手順を踏むべきである。可能な限り、脆弱性には次のように対処するべきである:

a) 排除：全ての実行可能な脆弱性を明らかにし、排除又は無効にする有効な手順を踏むべきである。

b) 最小化：脆弱性の実行による潜在的な影響を、容認できる残存レベルにまで軽減するための有効な手順を踏むべきである。

c) 監視：残存脆弱性を実行させる試みを検出し、損失を抑える手順を踏むことができるようにする有効な手順を踏むべきである。

5.3.3 脆弱性の原因

脆弱性は、以下の障害により起きることがある。

a) 要件：IT製品は、必要とされる全ての機能と特徴を所有しているが、なお、セキュリティに関してその製品を不適切又は無効にする脆弱性を含む。

b) 設計：IT製品の設計に問題がある。セキュアな製品、システム又はアプリケーションを構築するには、機能要件の実装だけでなく、その製品、システム、又はアプリケーションが実施する特定のセキュリティ特性を効果的に実施できるアーキテクチャも必要になる。製品、システム又はアプリケーションが意図した運用環境で直面する可能性がある攻撃に耐える能力は、これらの攻撃を禁止するか、もし禁止できない場合はそのような攻撃の検出やその被害の制限を可能にするアーキテクチャに大きく依存する。

c) 開発：IT製品がその仕様を満たしていない、及び/又は開発上の標準が不十分であるか、設計上の選択が不適切であるために脆弱性が導入される。

d) 配付、設置及び設定：IT製品は、製品の配付、設置及び設定の際に脆弱性が導入された。

e) 運用：IT製品は正しい仕様に従って正しく構成されているが、運用の管理が不適切であるために脆弱性が導入された。

f) 保守：IT製品が新しい脆弱性が導入されるような方法で保守されている。

5.3.4 CC 保証

保証は、実証されていない主張、これまでの関連する経験、又は特別の経験などのソースを参照することで得られる。ただし、このCCは、能動的な調査や仕様に基づいたアプローチを通して保証を提供する。能動的な調査とは、セキュリティ特性を決定するためのIT製品の評価である。

5.3.5 評価を通した保証

評価は、保証を得るための伝統的な手段であり、CCのアプローチの基礎となっている。評価技法には次のものが含まれるが、必ずしもこれだけに限定されない。

a) プロセス及び手続きの分析とチェック。

b) プロセス及び手続きが適用されていることのチェック。

c) TOE設計表現の間の対応分析。

d) 要件に対するTOE設計表現の分析。

e) 証拠書類の検証。

f) ガイダンス文書の分析。

g) 開発された機能テストと提供された結果の分析。

h) 独立機能テスト。

i) 脆弱性(欠陥仮説法を含む)の分析。

j) 侵入テスト。

k) 配付手続きの分析。

l) 保守手続きの分析。

5.4 CC評価保証の尺度

CCのアプローチは、評価のための労力が大きいほど、大きな保証結果が得られること、及び必要最小限の労力で必要な保証を提供することが目標であることを主張している。労力のレベルは、次のことに基づいて増加する。

a) 適用範囲：IT製品のより多くの部分が対象になると、労力は大きくなる。

b) 深さ：詳細な設計や詳細な実装を使用すると、労力は大きくなる。

c) 厳格性：より構造化された形式的な方法で適用されると、労力は大きくなる。

6 セキュリティ保証コンポーネント

6.1 一般

6.2から6.6までの節では、保証クラス、ファミリ、及びコンポーネントを表す際に使用される構造について記述する。

図1は、この文書に定義されているセキュリティ保証要件(SAR)を示している。SARの最も抽象的なセットは、クラスと呼ばれることに注意のこと。各クラスには保証ファミリが含まれ、保証ファミリには保証コンポーネントが含まれ、保証コンポーネントには保証エレメントが含まれる。クラスとファミリは、SARを分類するための分類方法を提供するために使われる。一方、コンポーネントは、PP/STでSARを特定するために使われる。

6.2 保証クラスの構造

6.2.1 一般

図1は、保証クラスの構造を示す。

6.2.2 クラス名

各保証クラスには一意の名前が割り当てられる。この名前は、保証クラスが扱うトピックを示す。

保証クラス名の一意の短い形式も提供される。これは、保証クラスを参照するための主な手段である。採用された規則では、「A」の次にクラス名に関係する2文字が続く。

6.2.3 クラスの概説

各保証クラスには、クラスの構成が記述され、クラスの意図を扱う補足説明を含む概説の節がある。

6.2.4 保証ファミリ

各保証クラスには、少なくとも1つの保証ファミリが含まれる。保証ファミリの構造については、次の節で記述する。

図1は、保証ファミリの構造を示す。

図 1 — 保証クラス/ファミリ/コンポーネント/エレメントの階層

6.3 保証ファミリの構造

6.3.1 ファミリ名

各保証ファミリには一意の名前が割り当てられる。この名前は、保証ファミリが扱うトピックについての記述情報を提供する。各保証ファミリは、同じ意図を持つ他のファミリが含まれている保証クラスの中に置かれる。

保証ファミリ名の一意の短い形式も提供される。これは、保証ファミリを参照するために使われる主な手段である。採用されている規則では、クラス名の短い形式が使われ、その後に下線文字が続き、次にファミリ名に関係する3文字が続く。

6.3.2 目的

保証ファミリの目的の節は、保証ファミリの意図を表す。

この節は、ファミリが扱うように意図されているCC保証のパラダイムに特に関係する目的を記述する。保証ファミリの記述は、全般的なレベルにとどめている。目的に必要な特別な詳細は、特定の保証コンポーネントに組み込まれる。

6.3.3 コンポーネントのレベル付け

各保証ファミリには、1つ又は複数の保証コンポーネントが含まれる。保証ファミリのこの節では、使用可能なコンポーネントについて記述し、それらの区別を説明する。主な目的は、保証ファミリが、PP/STに対するSARの必要な、又は有用な部分であることが決定された後に、これらの保証コンポーネントを区別することである。

複数のコンポーネントが含まれている保証ファミリは、レベル付けが行われ、コンポーネントにレベルを付ける方法の根拠が示される。この根拠は、適用範囲、深さ、及び/又は厳格性に関して示される。

6.3.4 適用上の注釈

保証ファミリに適用上の注釈の節が存在する場合、その節には保証ファミリの追加情報が含まれる。これは、保証ファミリの利用者(例えば、PPとSTの作成者、TOEの設計者、評価者)が特に関心を持つ情報である。表現は非形式的であり、例えば、使用上の制約及び特別の注意が必要となる領域に関する警告が扱われる。

6.3.5 保証コンポーネント

各保証ファミリには、少なくとも1つの保証コンポーネントが含まれる。保証コンポーネントの構造については、次の節で説明する。

6.4 保証コンポーネント構造

6.4.1 一般

図2は、保証コンポーネント構造を示す。

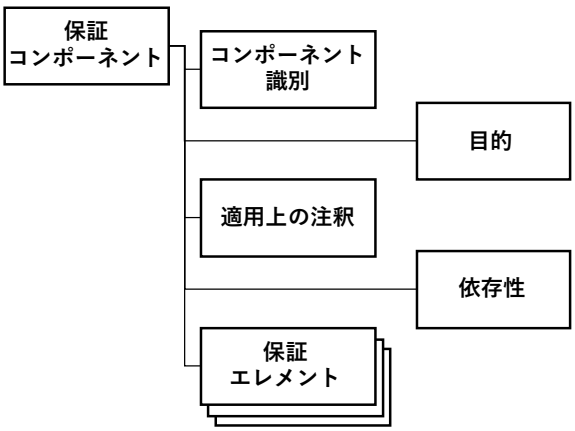


図 2 — 保証コンポーネント構造

ファミリ内のコンポーネント間の関係は、ボールド表記を用いて強調表示される。新規、及び階層内でこれまでのコンポーネントの要件を越えて強化又は修正されている要件のこれらの部分は、ボールドで表示される。

6.4.2 コンポーネント識別

コンポーネント識別の節は、コンポーネントを識別、分類、登録、及び参照するために必要な記述情報を提供する。

各保証コンポーネントには一意の名前が割り付けられる。この名前は、保証コンポーネントが扱うトピックについての記述情報を提供する。各保証コンポーネントは、セキュリティの目的を共有する保証ファミリの中に置かれる。

保証コンポーネント名の一意の短い形式も提供される。これは、保証コンポーネントを参照するために使われる主な手段である。使用される規則では、ファミリ名の短い形式が使用され、次にピリオドが続き、次に数字が続く。各ファミリ内のコンポーネントに対する数字は、1から順に割り付けられる。

6.4.3 目的

保証コンポーネントに目的の節が存在する場合、その節には特定の保証コンポーネントの特定の目的が含まれる。この節を持つ保証コンポーネントについて、コンポーネントの特定の意図を示し、目的のさらに詳細な説明を提供する。

6.4.4 適用上の注釈

保証コンポーネントの適用上の注釈の節が存在する場合には、コンポーネントを容易に使用するための追加情報が含まれる。

6.4.5 依存性

保証コンポーネントの間の依存性は、コンポーネントが自己完結型ではなく、他のコンポーネントの存在に依存するときに起きる。

各保証コンポーネントは、他の保証コンポーネントに対する依存性の完全なリストを提供する。コンポーネントによっては、「依存性なし」を示してもよい。これは、識別される依存性は存在しないことを示す。依存されているコンポーネントは、他のコンポーネントに依存してもよい。

依存性リストは、必要とされる最小限の保証コンポーネントのセットを識別する。依存性リストで、あるコンポーネントの下位階層にあるコンポーネントが、依存性を満たすために使用される場合もある。

特別の状況では、示された依存性が適用できない場合がある。PP、PPモジュール、PP構成又はSTの作成者が、特定の依存性を適用できない理由の根拠を提供することで、その依存性を満たさないことを選択してもよい。

6.4.6 保証エレメント

各保証コンポーネントには、保証エレメントのセットが提供される。保証エレメントは、それ以上分割しても意味のある評価結果が得られないセキュリティ要件である。これは、CCで認められている最小のセキュリティ要件である。

各保証エレメントは、保証エレメントの以下の3つのセットの1つに属するものとして識別される。

a) 開発者アクションエレメント: 開発者が行わなければならないアクティビティ。このアクションのセットは、次に続くエレメントのセットで参照されている証拠資料によってさらに評価付けされる。開発者アクションの要件は、エレメント番号の後に「D」の文字を追加することによって識別される。

b) 証拠の内容・提示エレメント: 必要とされる証拠、証拠が示さなければならないもの、及び証拠が伝えなければならない情報。証拠の内容・提示の要件は、エレメント番号の終わりに「C」の文字を追加することによって識別される。

c) 評価者アクションエレメント: 評価者が行わなければならないアクティビティ。このアクションのセットには、証拠の内容・提示エレメントに記述されている要件が満たされていることの確認が明示的に含まれる。また、開発者がすでに行っているものに加えて実行しなければならない明示的なアクションと分析も含まれる。暗黙の評価者アクションも、証拠の内容・提示要件に示されていない開発者のアクションエレメントの結果として実行される。評価者アクションの要件は、エレメント番号の終わりに「E」の文字を追加することにより識別される。

開発者アクションと証拠の内容・提示は、PP、PPモジュール、PP構成又はSTのSFRを満たしているTOEに保証を示す開発者の責任を表すために使用される保証要件を定義する。

評価者アクションは、評価の2つの側面での評価者の責任を定義する。第1の側面は、「ACE: プロテクションプロファイル構成評価」、「APE: プロテクションプロファイル評価」、及び「ASE: セキュリティターゲット評価」の章のACEクラス、APEクラスとASEクラスに従った、該当するPP、PPモジュール、PP構成又はSTの妥当性の確認である。第2の側面は、TOEのそのSFRとSARに対する適合性の検証である。PP、PPモジュール、PP構成又はSTが妥当であり、要件がTOEによって満たされていることを実証することにより、評価者は、定義されたセキュリティの課題をTOEがその運用環境で解決するという信頼の基礎を提供できる。

開発者アクションエレメント、証拠の内容・提示エレメント、及び明示的評価者アクションエレメントは、TOEのSTにおいてなされるセキュリティ主張の検証に費やされなければならない評価者の労力を識別している。

6.5 保証エレメント

各エレメントは、満たす必要がある要件を表す。要件のこれらのステートメントは、明確かつ簡潔で、曖昧でないことが意図されている。したがって、重文は存在せず、分離可能な要件はそれぞれ個別のエレメントとして記述される。

6.6 コンポーネントの分類

この文書には、関係する保証に基づいてグループ化されたファミリのクラスとコンポーネントが含まれている。各クラスの冒頭に、クラス内のファミリと各ファミリのコンポーネントを表す図が示される。

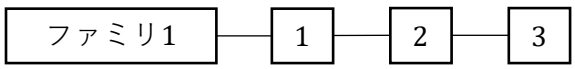


図 3 — サンプルクラスのコンポーネント構成図

図 3 には、単一のファミリを含んだクラスが示されている。このファミリには、直線的に階層化された 3 つのコンポーネントが含まれている(つまり、コンポーネント 2 は、特定のアクション、特定の証拠、あるいはアクション又は証拠の厳格性の観点から、コンポーネント 1 以上を必要とする)。この文書の保証ファミリは全て直線的に階層化されているが、将来追加される保証ファミリにとって直線性は必須の基準ではない。

7 APEクラス: プロテクションプロファイル(PP)評価

7.1 一般

PPの評価は、PPが信頼でき内部的に一貫していること、及びPPが1つ又は複数のPP又はパッケージに基づいている場合に、それらのPPやパッケージをPPが正しく具体化していることを実証するために必要である。これらの特性は、PPが、ST又は他のPPを記述するための基礎として使用するのに適しているために必要である。

7章は、CCパート1の附属書B及びDとともに使用されるべきである。これらの附属書は、ここでの概念を明確にし、多くの例を提供する。

図4は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。

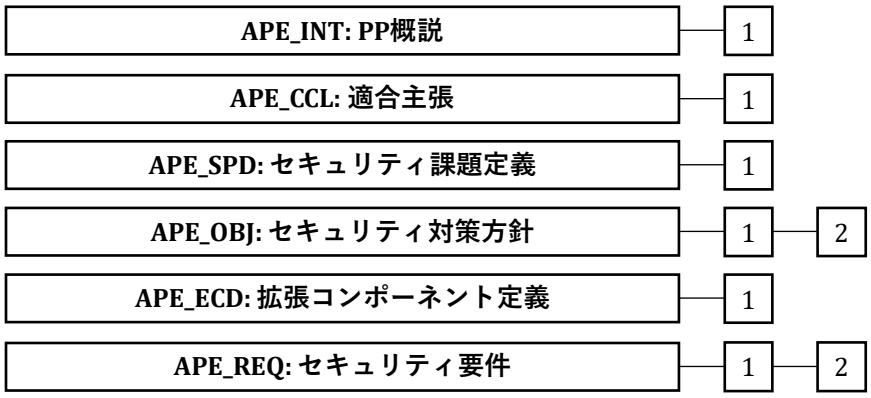


図 4 — APE: プロテクションプロファイル(PP)評価クラスのコンポーネント構成

7.2 PP 概説(APE_INT)

7.2.1 目的

このファミリの目的は、TOEを順序立てて記述することである。

PP概説の評価は、PPが正しく識別されていること、及びPP参照とTOE概要が相互に一貫していることを実証するために必要である。

7.2.2 APE_INT.1 PP概説

依存性：なし

開発者アクションエレメント:

APE_INT.1.1D

開発者は、PP概説を提供しなければならない。

内容・提示エレメント:

APE_INT.1.1C

PP概説は、PP参照とTOE概要を含めなければならない。

APE_INT.1.2C

PP参照は、PPを一意に識別しなければならない。

APE_INT.1.3C

TOE概要は、TOEの使用法及び主要なセキュリティ機能の特徴を要約しなければならない。

APE_INT.1.4C

TOE概要は、TOE種別を識別しなければならない。

APE_INT.1.5C

TOE概要は、TOEが利用できるTOE以外のハードウェア/ソフトウェア/ファームウェアを識別しなければならない。

評価者アクションエレメント:

APE_INT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

7.3 適合主張(APE_CCL)

7.3.1 目的

このファミリの目的は、適合主張の有効性を決定することである。さらに、このファミリは、ST及び他のPPがPPに対する適合を主張する方法を特定する。

7.3.2 APE_CCL.1 適合主張

依存性： APE_INT.1 PP概説
APE_ECD.1 拡張コンポーネント定義
APE_REQ.1 直接根拠PPモジュールのセキュリティ要件

開発者アクションエレメント:

APE_CCL.1.1D

開発者は、適合主張を提供しなければならない。

APE_CCL.1.2D

開発者は、適合主張根拠を提供しなければならない。

APE_CCL.1.3D

開発者は、適合ステートメントを提供しなければならない。

内容・提示エレメント:

APE_CCL.1.1C

適合主張は、PPが適合を主張するCCの版を識別しなければならない。

APE_CCL.1.2C

適合主張は、CCパート2に対するPPの適合をCCパート2適合又はCCパート2拡張のいずれかとして記述しなければならない。

APE_CCL.1.3C

適合主張は、PPの適合を「CCパート3適合」又は「CCパート3拡張」のいずれかとして記述しなければならない。

APE_CCL.1.4C

適合主張は、拡張コンポーネント定義と一貫していなければならない。

APE_CCL.1.5C

適合主張は、PPが適合を主張するPP及びパッケージを全て識別しなければならない。

APE_CCL.1.6C

適合主張は、機能パッケージに対するPPの適合をパッケージ適合、パッケージ追加又はパッケージ調整のいずれかとして記述しなければならない。

APE_CCL.1.7C

適合主張は、保証パッケージに対するPPの適合をパッケージ適合又はパッケージ追加のいずれかとして記述しなければならない。

APE_CCL.1.8C

適合主張は、他のPPに対するPPの適合をPP適合として記述しなければならない。

APE_CCL.1.9C

適合主張根拠は、TOE種別が、適合が主張されているPP内のTOE種別と一貫していることを実証しなければならない。

APE_CCL.1.10C

適合主張根拠は、セキュリティ課題定義のステートメントが、適合が主張されているPP及び機能パッケージ内のセキュリティ課題定義のステートメントと一貫していることを実証しなければならない。

APE_CCL.1.11C

適合主張根拠は、セキュリティ対策方針のステートメントが、適合が主張されているPP及び機能パッケージ内のセキュリティ対策方針のステートメントと一貫していることを実証しなければならない。

APE_CCL.1.12C

適合主張根拠は、セキュリティ要件のステートメントが、適合が主張されているPP及び機能パッケージ内のセキュリティ要件のステートメントと一貫していることを実証しなければならない。

APE_CCL.1.13C

適合ステートメントは、PPに対する任意のPP/STに必要とされる適合を、完全適合、正確適合又は論証適合のいずれかとして記述しなければならない。

APE_CCL.1.14C

完全適合PPの場合、適合ステートメントには、評価されているPPと組み合わせて、完全適合を主張することが許されるPPのセットを(もしあれば)識別する、併用許可ステートメントが含まれなければならない。

APE_CCL.1.15C

完全適合PPの場合、適合ステートメントには、PP構成において評価されているPPと組み合わせて、使用することが許可されるPPモジュールのセットを(もしあれば)識別する、併用許可ステートメントが含まれなければならない。

APE_CCL.1.16C

適合ステートメントは、評価されているPPで使用されなければならない派生した評価方法と評価アクティビティのセットを(もしあれば)識別しなければならない。このリストには以下のものが含まれなければならない。

— 評価されているPPに特定された評価方法及び評価アクティビティ

— 評価されているPPが適合を主張するPPの適合ステートメントで特定された評価方法及び評価アクティビティ

— 評価されているPPが適合を主張するパッケージのセキュリティ要件の節で特定された評価方法と評価アクティビティ

評価者アクションエレメント:

APE_CCL.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

7.4 セキュリティ課題定義(APE_SPD)

7.4.1 目的

PPのこの部分は、TOE及びTOEの運用環境によって対処されるセキュリティ課題を定義する。

セキュリティ課題定義の評価は、TOE及びTOEの運用環境によって対処されることが意図されているセキュリティ課題が明確に定義されていることを実証するために必要である。

7.4.2 APE_SPD.1 セキュリティ課題定義

依存性：なし

開発者アクションエレメント:

APE_SPD.1.1D

開発者は、セキュリティ課題定義を提供しなければならない。

内容・提示エレメント:

APE_SPD.1.1C

セキュリティ課題定義は、*脅威*を記述しなければならない。

APE_SPD.1.2C

全ての*脅威*は、*脅威*エージェント、資産、及び有害なアクションの観点から記述しなければならない。

APE_SPD.1.3C

セキュリティ課題定義は、組織のセキュリティ方針(OSP)を記述しなければならない。

APE_SPD.1.4C

セキュリティ課題定義は、TOEの運用環境についての前提条件を記述しなければならない。

評価者アクションエレメント:

APE_SPD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

7.5 セキュリティ対策方針(APE_OBJ)

7.5.1 目的

セキュリティ対策方針は、セキュリティ課題定義(APE_SPD)ファミリを通して定義されるセキュリティ課題に対して意図される対応の簡潔なステートメントである。

セキュリティ対策方針の評価は、セキュリティ対策方針が適切かつ完全にセキュリティ課題定義を扱うこと、及びTOEとその運用環境の間でのこの課題に対する分担が明確に定義されていることを実証するために必要である。

7.5.2 コンポーネントのレベル付け

このファミリのコンポーネントは、運用環境のセキュリティ対策方針のみを記述しているのか、又はTOEのセキュリティ対策方針も含めて記述しているのかによって、レベル付けされている。

7.5.3 APE_OBJ.1 運用環境のセキュリティ対策方針

依存性：なし

開発者アクションエレメント:

APE_OBJ.1.1D

開発者は、運用環境のセキュリティ対策方針のステートメントを提供しなければならない。

APE_OBJ.1.2D

開発者は、運用環境のセキュリティ対策方針根拠[v]を提供しなければならない。

内容・提示エレメント:

APE_OBJ.1.1C

セキュリティ対策方針のステートメントは、運用環境のセキュリティ対策方針を記述しなければならない。

APE_OBJ.1.2C

セキュリティ対策方針根拠は、運用環境の各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威、そのセキュリティ対策方針によって実施されるOSP、及びそのセキュリティ対策方針によって充足される前提条件にまで、さかのぼって追跡しなければならない。

APE_OBJ.1.3C

セキュリティ対策方針根拠は、運用環境のセキュリティ対策方針が全ての前提条件を充足することを実証しなければならない。

評価者アクションエレメント:

APE_OBJ.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

7.5.4 APE_OBJ.2 セキュリティ対策方針

依存性：APE_SPD.1 セキュリティ課題定義

開発者アクションエレメント:

APE_OBJ.2.1D

開発者は、セキュリティ対策方針のステートメントを提供しなければならない。

APE_OBJ.2.2D

開発者は、セキュリティ対策方針根拠を提供しなければならない。

内容・提示エレメント:

APE_OBJ.2.1C

セキュリティ対策方針のステートメントは、TOEのセキュリティ対策方針及び運用環境のセキュリティ対策方針を記述しなければならない。

APE_OBJ.2.2C

セキュリティ対策方針根拠は、TOEの各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威及びそのセキュリティ対策方針によって実施されるOSPまでさかのぼって追跡しなければならない。

APE_OBJ.2.3C

セキュリティ対策方針根拠は、運用環境の各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威、そのセキュリティ対策方針によって実施されるOSP、及びそのセキュリティ対策方針によって充足される前提条件にまで、さかのぼって追跡しなければならない。

APE_OBJ.2.4C

セキュリティ対策方針根拠は、セキュリティ対策方針が全ての脅威に対抗することを実証しなければならない。

APE_OBJ.2.5C

セキュリティ対策方針根拠は、セキュリティ対策方針が全てのOSPを実施することを実証しなければならない。

APE_OBJ.2.6C

セキュリティ対策方針根拠は、運用環境のセキュリティ対策方針が全ての前提条件を充足することを実証しなければならない。

評価者アクションエレメント:

APE_OBJ.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

7.6 拡張コンポーネント定義(APE_ECD)

7.6.1 目的

拡張セキュリティ要件は、CCパート2又はこの文書のコンポーネントではなく、拡張コンポーネント、つまりPPの作成者によって定義されるコンポーネントに基づく要件である。

拡張コンポーネント定義の評価は、拡張コンポーネントが明確で曖昧さがないこと、及びそれらが必要であること、つまり既存のCCパート2又はこの文書のコンポーネントを使用して明確には表現できないことを決定するために必要である。

7.6.2 APE_ECD.1 拡張コンポーネント定義

依存性：なし

開発者アクションエレメント:

APE_ECD.1.1D

開発者は、セキュリティ要件のステートメントを提供しなければならない。

APE_ECD.1.2D

開発者は、拡張コンポーネント定義を提供しなければならない。

内容・提示エレメント:

APE_ECD.1.1C

セキュリティ要件のステートメントは、全ての拡張セキュリティ要件を識別しなければならない。

APE_ECD.1.2C

拡張コンポーネント定義は、各拡張セキュリティ要件に対応する拡張コンポーネントを定義しなければならない。

APE_ECD.1.3C

拡張コンポーネント定義は、各拡張コンポーネントが既存のCCコンポーネント、ファミリ、及びクラスにどのように関連するかを記述しなければならない。

APE_ECD.1.4C

拡張コンポーネント定義は、提示モデルとして既存のCCコンポーネント、ファミリ、クラス、及び方法を使用しなければならない。

APE_ECD.1.5C

拡張コンポーネントは、エレメントに対する適合又は非適合を実証できるように、評価可能で客観的なエレメントで構成されていなければならない。

評価者アクションエレメント:

APE_ECD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

APE_ECD.1.2E

評価者は、拡張コンポーネントが既存のコンポーネントを使用して明確には表現できないことを*確認しなければならない*。

7.7 セキュリティ要件(APE_REQ)

7.7.1 目的

SFRは、TOEに期待されるセキュリティのふるまいについての、明確で曖昧さがなく十分に定義された記述となる。SARは、TOEで保証を得るために採用される期待されるアクティビティについての、明確で曖昧さがなく十分に定義された記述となる。

セキュリティ要件の評価は、それらの要件が明確で曖昧さがなく十分に定義されていることを保証するために必要である。

7.7.2 コンポーネントのレベル付け

このファミリのコンポーネントは、SFRがSPDから導き出されているのか、又はSFRがTOEのセキュリティ対策方針から導き出されているのかによって、レベル付けされている。

7.7.3 APE_REQ.1 直接根拠PPモジュールのセキュリティ要件

依存性： APE_ECD.1 拡張コンポーネント定義

APE_OBJ.1 運用環境のセキュリティ対策方針

開発者アクションエレメント:

APE_REQ.1.1D

開発者は、セキュリティ要件のステートメントを提供しなければならない。

APE_REQ.1.2D

開発者は、セキュリティ要件根拠を提供しなければならない。

内容・提示エレメント:

APE_REQ.1.1C

セキュリティ要件のステートメントは、SFR及びSARを記述しなければならない。

APE_REQ.1.2C

SFR及びSARで使用される全てのサブジェクト、オブジェクト、操作、セキュリティ属性、外部のエンティティ及びその他の用語が定義されなければならない。

APE_REQ.1.3C

セキュリティ要件のステートメントは、セキュリティ要件の全ての操作を識別しなければならない。

APE_REQ.1.4C

全ての操作は正しく実行しなければならない。

APE_REQ.1.5C

セキュリティ要件の各依存性が満たされていなければならない。また、満たされない依存性がある場合は、セキュリティ要件根拠によってそのことが正当化されなければならない。

APE_REQ.1.6C

セキュリティ要件根拠は、各SFRを、そのSFRによって対抗される脅威及びそのSFRによって実施されるOSPにまでさかのぼって追跡しなければならない。

APE_REQ.1.7C

セキュリティ要件根拠は、SFRが（運用環境のセキュリティ対策方針と合わせて）TOEの全ての脅威に対抗していることを実証しなければならない。

APE_REQ.1.8C

セキュリティ要件根拠は、SFRが（運用環境のセキュリティ対策方針と合わせて）TOEのOSPの全てを実施することを実証しなければならない。

APE_REQ.1.9C

セキュリティ要件根拠は、なぜSARが選ばれたかを説明しなければならない。

APE_REQ.1.10C

セキュリティ要件のステートメントは、内部的に一貫していなければならない。

評価者アクションエレメント:

APE_REQ.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

7.7.4 APE_REQ.2 導出されたセキュリティ要件

依存性： APE_OBJ.2 セキュリティ対策方針

APE_ECD.1 拡張コンポーネント定義

開発者アクションエレメント:

APE_REQ.2.1D

開発者は、セキュリティ要件のステートメントを提供しなければならない。

APE_REQ.2.2D

開発者は、セキュリティ要件根拠を提供しなければならない。

内容・提示エレメント:

APE_REQ.2.1C

セキュリティ要件のステートメントは、SFR及びSARを記述しなければならない。

APE_REQ.2.2C

SFR及びSARで使用される全てのサブジェクト、オブジェクト、操作、セキュリティ属性、外部のエンティティ及びその他の用語が定義されなければならない。

APE_REQ.2.3C

セキュリティ要件のステートメントは、セキュリティ要件の全ての操作を識別しなければならない。

APE_REQ.2.4C

全ての操作は正しく実行しなければならない。

APE_REQ.2.5C

セキュリティ要件の各依存性が満たされていなければならない。また、満たされない依存性がある場合は、セキュリティ要件根拠によってそのことが正当化されなければならない。

APE_REQ.2.6C

セキュリティ要件根拠は、各SFRを、そのSFRによって実施されるTOEのセキュリティ対策方針にまでさかのぼって追跡しなければならない。

APE_REQ.2.7C

セキュリティ要件根拠は、SFRがTOEのセキュリティ対策方針の全てを満たすことを実証しなければならない。

APE_REQ.2.8C

セキュリティ要件根拠は、なぜSARが選ばれたかを説明しなければならない。

APE_REQ.2.9C

セキュリティ要件のステートメントは、内部的に一貫していなければならない。

評価者アクションエレメント:

APE_REQ.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8 ACEクラス: プロテクションプロファイル構成評価

8.1 一般

PP構成の評価は、PP構成が信頼でき一貫していることを実証するために必要である。これらの特性は、PP構成が、STを記述するための基礎として使用するのに適しているために必要である。

ACEクラスは、少なくとも1つのPPと1つの他のコンポーネント(PP及び/又はPPモジュール)から構成されるPP構成の評価に対して定義される。PPの評価については、APEクラスで扱われる。ACEクラスでは、以下の要件を定義している：

— PPモジュール基盤の枠組みで、PPモジュールの評価をすること(ACE_INT.1、ACE_CCL.1、ACE_SPD.1、ACE_OBJ.1又は2、ACE_REQ.1又は2、及びACE_MCO.1)。

— PP構成に属する全てのPPとPPモジュールの整合性を評価すること(ACE_CCO.1を参照)。

8章は、CCパート1付属書Cと共に使用されるべきである。

図 5 — ACE: プロテクションプロファイル構成評価クラスのコンポーネント構成

8.2 PP モジュール概説(ACE_INT)

8.2.1 目的

このファミリの目的は、TOEを順序立てて記述することである。

PPモジュール概説の評価は、PPモジュールが正しく識別されていること、及びPPモジュール参照とTOE概要が相互に一貫していることを実証するために必要である。

8.2.2 ACE_INT.1 PPモジュール概説

依存性：なし

開発者アクションエレメント:

ACE_INT.1.1D

開発者は、PPモジュール概説を提供しなければならない。

内容・提示エレメント:

ACE_INT.1.1C

PPモジュール概説は、PPモジュール参照、PPモジュール基盤の識別及びTOE概要を含めなければならない。

ACE_INT.1.2C

PPモジュール参照は、PPモジュールを一意に識別しなければならない。

ACE_INT.1.3C

PPモジュール基盤の識別は、少なくとも一つのPP、場合によってはPPモジュールが依存する他のPP及びPPモジュールで構成されなければならない。

ACE_INT.1.4C

PPモジュール基盤の識別は、PPモジュール基盤の依存構造を記述しなければならない。

ACE_INT.1.5C

PPモジュール概説には、代替のPPモジュール基盤と同数のTOE概要が含まれていなければならない。

ACE_INT.1.6C

TOE概要は、TOEの使用法及び主要なセキュリティ機能の特徴を要約しなければならない。

ACE_INT.1.7C

TOE概要は、TOE種別を識別しなければならない。

ACE_INT.1.8C

TOE概要は、TOEが利用できるTOE以外のハードウェア/ソフトウェア/ファームウェアを識別しなければならない。

ACE_INT.1.9C

TOE概要には、PPモジュール基盤に定義されているTOEとの相違点を記述しなければならない。

評価者アクションエレメント:

ACE_INT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8.3 PPモジュール適合主張(ACE_CCL)

8.3.1 目的

このファミリの目的は、適合主張及び適合ステートメントの有効性を決定することである。PPモジュールは、いかなるPP、PP構成、又は他のPPモジュールにも適合を主張できない。

8.3.2 ACE_CCL.1 PPモジュール適合主張

依存性： ACE_INT.1 PPモジュール概説

ACE_ECD.1 PPモジュール拡張コンポーネント定義

ACE_REQ.1 PPモジュールの主張したセキュリティ要件又はACE_REQ.2 PPモジュールの導出されたセキュリティ要件

開発者アクションエレメント:

ACE_CCL.1.1D

開発者は、適合主張を提供しなければならない。

ACE_CCL.1.2D

開発者は、適合ステートメントを提供しなければならない。

内容・提示エレメント:

ACE_CCL.1.1C

適合主張は、PPモジュールが適合を主張するCCの版を識別しなければならない。

ACE_CCL.1.2C

適合主張は、CCパート2に対するPPモジュールの適合をCCパート2適合又はCCパート2拡張のいずれかとして記述しなければならない。

ACE_CCL.1.3C

適合ステートメントは、(PP構成の一部として)PPモジュールに対するSTに要求される適合種別を、完全適合、正確適合又は論証適合のいずれかとして記述しなければならない。

ACE_CCL.1.4C

CC適合主張は、CCパート3[vi]に対するPPモジュールの適合を「CCパート3適合」又は「CCパート3拡張」のいずれかとして記述しなければならない。

ACE_CCL.1.5C

適合主張は、拡張コンポーネント定義と一貫していなければならない。

ACE_CCL.1.6C

適合主張は、PPモジュールが適合を主張する機能パッケージを全て識別しなければならない。

ACE_CCL.1.7C

適合主張は、機能パッケージに対するPPモジュールの適合をパッケージ適合、パッケージ追加又はパッケージ調整のいずれかとして記述しなければならない。

ACE_CCL.1.8C

適合主張は、PPモジュールが適合を主張する保証パッケージを全て識別しなければならない。

ACE_CCL.1.9C

適合主張は、保証パッケージに対するPPモジュールの適合をパッケージ適合又はパッケージ追加のいずれかとして記述しなければならない。

ACE_CCL.1.10C

完全適合の場合、PPモジュールの適合ステートメントは、評価されているPPモジュールと組み合わせて、完全適合の主張を許可されているPP及びPPモジュールのセット(PPモジュール基盤に含まれるPP及びPPモジュールを除く)を識別する併用許可ステートメントを含まなければならない。

ACE_CCL.1.11C

適合ステートメントは、評価されているPPモジュールで使用されなければならない、CEMから派生した評価方法と評価アクティビティのセットを識別することができる。このリストには、PPモジュールで特定されている評価方法と評価アクティビティだけでなく、PPモジュール基盤及び/又はパッケージ(もしあれば)で特定されている評価方法と評価アクティビティも含まれなければならず、それらは評価されているPPモジュールが適合を主張しているものでなければならない。

評価者アクションエレメント:

ACE_CCL.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8.4 PPモジュールセキュリティ課題定義(ACE_SPD)

8.4.1 目的

PPモジュールのこの部分は、TOE及びTOEの運用環境によって対処されるセキュリティ課題を定義する。

セキュリティ課題定義の評価は、TOE及びTOEの運用環境によって対処されることが意図されているセキュリティ課題が明確に定義されていることを実証するために必要である。

8.4.2 ACE_SPD.1 PPモジュールセキュリティ課題定義

依存性：なし

開発者アクションエレメント:

ACE_SPD.1.1D

開発者は、セキュリティ課題定義を提供しなければならない。

内容・提示エレメント:

ACE_SPD.1.1C

セキュリティ課題定義は、*脅威*を記述しなければならない。

ACE_SPD.1.2C

全ての*脅威*は、*脅威*エージェント、資産、及び有害なアクションの観点から記述しなければならない。

ACE_SPD.1.3C

セキュリティ課題定義は、OSPを記述しなければならない。

ACE_SPD.1.4C

セキュリティ課題定義は、TOEの運用環境についての前提条件を記述しなければならない。

評価者アクションエレメント:

ACE_SPD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8.5 PPモジュールセキュリティ対策方針(ACE_OBJ)

8.5.1 目的

セキュリティ対策方針は、セキュリティ課題定義(APE_SPD)ファミリを通して定義されるセキュリティ課題に対して意図される対応の簡潔なステートメントである。

セキュリティ対策方針の評価は、セキュリティ対策方針が適切かつ完全にセキュリティ課題定義を扱うこと、及びTOEとその運用環境の間でのこの課題に対する分担が明確に定義されていることを実証するために必要である。

8.5.2 コンポーネントのレベル付け

このファミリのコンポーネントは、運用環境のセキュリティ対策方針のみを記述しているのか(ACE_OBJ.1参照)、又はTOEのセキュリティ対策方針も含めて記述しているのか(ACE_OBJ.2参照)によって、レベル付けされている。

8.5.3 ACE_OBJ.1 PPモジュール運用環境のセキュリティ対策方針

依存性：なし

開発者アクションエレメント:

ACE_OBJ.1.1D

開発者は、PPモジュールの運用環境のセキュリティ対策方針のステートメントを提供しなければならない。

ACE_OBJ.1.2D

開発者は、PPモジュールの運用環境のセキュリティ対策方針根拠を提供しなければならない。

内容・提示エレメント:

ACE_OBJ.1.1C

セキュリティ対策方針のステートメントは、運用環境のセキュリティ対策方針を記述しなければならない。

ACE_OBJ.1.2C

セキュリティ対策方針根拠は、運用環境の各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威、そのセキュリティ対策方針によって実施されるOSP、及びそのセキュリティ対策方針によって充足される前提条件にまで、さかのぼって追跡しなければならない。

ACE_OBJ.1.3C

セキュリティ対策方針根拠は、運用環境のセキュリティ対策方針が全ての前提条件を充足することを実証しなければならない。

評価者アクションエレメント:

ACE_OBJ.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8.5.4 ACE_OBJ.2 PPモジュールセキュリティ対策方針

依存性：ACE_SPD.1 PPモジュールセキュリティ課題定義

開発者アクションエレメント:

ACE_OBJ.2.1D

開発者は、PPモジュールのセキュリティ対策方針のステートメントを提供しなければならない。

ACE_OBJ.2.2D

開発者は、PPモジュールのセキュリティ対策方針根拠を提供しなければならない。

内容・提示エレメント:

ACE_OBJ.2.1C

セキュリティ対策方針のステートメントは、TOEのセキュリティ対策方針及び運用環境のセキュリティ対策方針を記述しなければならない。

ACE_OBJ.2.2C

セキュリティ対策方針根拠は、TOEの各セキュリティ対策方針をそのセキュリティ対策方針によって対抗される脅威及びそのセキュリティ対策方針によって実施されるOSPまでさかのぼって追跡しなければならない。

ACE_OBJ.2.3C

セキュリティ対策方針根拠は、運用環境の各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威、そのセキュリティ対策方針によって実施されるOSP、及びそのセキュリティ対策方針によって充足される前提条件にまで、さかのぼって追跡しなければならない。

ACE_OBJ.2.4C

セキュリティ対策方針根拠は、セキュリティ対策方針が全ての脅威に対抗することを実証しなければならない。

ACE_OBJ.2.5C

セキュリティ対策方針根拠は、セキュリティ対策方針が全てのOSPを実施することを実証しなければならない。

ACE_OBJ.2.6C

セキュリティ対策方針根拠は、運用環境のセキュリティ対策方針が全ての前提条件を充足することを実証しなければならない。

評価者アクションエレメント:

ACE_OBJ.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8.6 PPモジュール拡張コンポーネント定義(ACE_ECD)

8.6.1 目的

拡張SFRは、CCパート2又はこの文書のコンポーネントではなく、拡張コンポーネント、つまりPPモジュールの作成者によって定義されるコンポーネントに基づく要件である。

拡張機能コンポーネント定義の評価は、拡張機能コンポーネントが明確で曖昧さがないこと、及びそれらが必要であること、つまり既存のCCパート2又はこの文書のコンポーネントを使用して明確には表現できないことを決定するために必要である。

8.6.2 ACE_ECD.1 PPモジュール拡張コンポーネント定義

依存性：なし

開発者アクションエレメント:

ACE_ECD.1.1D

開発者は、PPモジュールのセキュリティ要件のステートメントを提供しなければならない。

ACE_ECD.1.2D

開発者は、PPモジュールの拡張コンポーネント定義を提供しなければならない。

内容・提示エレメント:

ACE_ECD.1.1C

セキュリティ要件のステートメントは、全ての拡張セキュリティ要件を識別しなければならない。

ACE_ECD.1.2C

拡張コンポーネント定義は、各拡張セキュリティ要件に対応する拡張コンポーネントを定義しなければならない。

ACE_ECD.1.3C

拡張コンポーネント定義は、各拡張コンポーネントが既存のCCコンポーネント、ファミリ、及びクラスにどのように関連するかを記述しなければならない。

ACE_ECD.1.4C

拡張コンポーネント定義は、提示モデルとして既存のCCコンポーネント、ファミリ、クラス、及び方法を使用しなければならない。

ACE_ECD.1.5C

拡張コンポーネントは、エレメントに対する適合又は非適合を実証できるように、評価可能で客観的なエレメントで構成されていなければならない。

評価者アクションエレメント:

ACE_ECD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACE_ECD.1.2E

評価者は、拡張コンポーネントが既存のコンポーネントを使用して明確には表現できないことを*確認しなければならない*。

8.7 PPモジュールセキュリティ要件(ACE_REQ)

8.7.1 目的

SFRは、TOEに期待されるセキュリティのふるまいについての、明確で曖昧さがなく十分に定義された記述となる。SARは、TOEで保証を得るために採用される期待されるアクティビティについての、明確で曖昧さがなく十分に定義された記述となる。

セキュリティ要件の評価は、それらの要件が明確で曖昧さがなく十分に定義されていることを保証するために必要である。

8.7.2 コンポーネントのレベル付け

このファミリのコンポーネントは、SFRがSPDから導き出されているのか(ACE_REQ.1参照)、又はSFRがTOEのセキュリティ対策方針から導き出されているのか(ACE_REQ.2参照)によって、レベル付けされている。

8.7.3 ACE_REQ.1 PPモジュールの主張したセキュリティ要件

依存性： APE_ECD.1 拡張コンポーネント定義

ACE_SPD.1 PPモジュールセキュリティ課題定義

開発者アクションエレメント:

ACE_REQ.1.1D

開発者は、PPモジュールのセキュリティ要件のステートメントを提供しなければならない。

ACE_REQ.1.2D

開発者は、PPモジュールのセキュリティ要件根拠を提供しなければならない。

内容・提示エレメント:

ACE_REQ.1.1C

セキュリティ要件のステートメントは、SFR及びSAR(PPモジュールに適用されるSARは、明示されてもよいし、PPモジュール基盤から継承されてもよい)を記述しなければならない。

ACE_REQ.1.2C

SFR及びSARで使用される全てのサブジェクト、オブジェクト、操作、セキュリティ属性、外部のエンティティ及びその他の用語が定義されなければならない。

ACE_REQ.1.3C

セキュリティ要件のステートメントは、セキュリティ要件の全ての操作を識別しなければならない。

ACE_REQ.1.4C

全ての操作は正しく実行しなければならない。

ACE_REQ.1.5C

セキュリティ要件の各依存性が満たされていなければならない。また、満たされない依存性がある場合は、セキュリティ要件根拠によってそのことが正当化されなければならない。

ACE_REQ.1.6C

セキュリティ要件根拠は、各SFRを、そのSFRによって対抗される脅威及びそのSFRによって実施されるOSPにまでさかのぼって追跡しなければならない。

ACE_REQ.1.7C

セキュリティ要件根拠は、SFRが(運用環境のセキュリティ対策方針と合わせて)TOEの全ての脅威に対抗していることを実証しなければならない。

ACE_REQ.1.8C

セキュリティ要件根拠は、SFRが(運用環境のセキュリティ対策方針と合わせて)TOEのOSPの全てを実施することを実証しなければならない。

ACE_REQ.1.9C

セキュリティ要件根拠は、なぜSARが選ばれたかを説明しなければならない。

ACE_REQ.1.10C

セキュリティ要件のステートメントは、内部的に一貫していなければならない。

評価者アクションエレメント:

ACE_REQ.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていること*を確認しなければならない*。

8.7.4 ACE_REQ.2 PPモジュールの導出されたセキュリティ要件

依存性： ACE_ECD.1 PPモジュール拡張コンポーネント定義

ACE_OBJ.2 PPモジュールセキュリティ対策方針

開発者アクションエレメント:

ACE_REQ.2.1D

開発者は、PPモジュールのセキュリティ要件のステートメントを提供しなければならない。

ACE_REQ.2.2D

開発者は、PPモジュールのセキュリティ要件根拠を提供しなければならない。

内容・提示エレメント:

ACE_REQ.2.1C

セキュリティ要件のステートメントは、SFR及びSAR(PPモジュールに適用されるSARは、明示されてもよいし、PPモジュール基盤から継承されてもよい)を記述しなければならない。

ACE_REQ.2.2C

SFR及びSARで使用される全てのサブジェクト、オブジェクト、操作、セキュリティ属性、外部のエンティティ及びその他の用語が定義されなければならない。

ACE_REQ.2.3C

セキュリティ要件のステートメントは、セキュリティ要件の全ての操作を識別しなければならない。

ACE_REQ.2.4C

全ての操作は正しく実行しなければならない。

ACE_REQ.2.5C

セキュリティ要件の各依存性が満たされていなければならない。また、満たされない依存性がある場合は、セキュリティ要件根拠によってそのことが正当化されなければならない。

ACE_REQ.2.6C

セキュリティ要件根拠は、各SFRを、そのSFRによって実施される**TOEのセキュリティ対策方針**にまでさかのぼって追跡しなければならない。

ACE_REQ.2.7C

セキュリティ要件根拠は、SFRが**TOEのセキュリティ対策方針の全てを満たす**ことを実証しなければならない。

ACE_REQ.2.8C

セキュリティ要件根拠は、なぜSARが選ばれたかを説明しなければならない。

ACE_REQ.2.9C

セキュリティ要件のステートメントは、内部的に一貫していなければならない。

評価者アクションエレメント:

ACE_REQ.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

8.8 PPモジュール一貫性(ACE_MCO)

8.8.1 目的

このファミリの目的は、PPモジュールの一貫性を決定し、PPモジュールとPPモジュール基盤との対応関係を述べることである。

8.8.2 ACE_MCO.1 PPモジュール一貫性

依存性：　ACE_INT.1 PPモジュール概説

ACE_SPD.1 PPモジュールセキュリティ課題定義

ACE_OBJ.1 直接根拠PPモジュール運用環境のセキュリティ対策方針又は ACE_OBJ.2 PPモジュールセキュリティ対策方針

ACE_REQ.1 直接根拠PPモジュールの主張されたセキュリティ要件又はACE_REQ.2 PPモジュールの導出されたセキュリティ要件

開発者アクションエレメント:

ACE_MCO.1.1D

開発者は、PPモジュール概説で識別されている各PPモジュール基盤について、PPモジュールの一貫性根拠を提供しなければならない。

ACE_MCO.1.2D

開発者は、PPモジュール概説で識別されている各PPモジュール基盤について、PPモジュールの保証根拠を提供しなければならない。

内容・提示エレメント:

ACE_MCO.1.1C

一貫性根拠は、PPモジュールのTOE種別が、PPモジュール基盤のTOE種別と一貫していることを実証しなければならない。

ACE_MCO.1.2C

一貫性根拠は、PPモジュールのSPDで定義された資産のうちPPモジュール基盤にも属するものを識別し、その中でPPモジュールとPPモジュール基盤が異なるセキュリティ課題を定義している資産を識別しなければならない。

ACE_MCO.1.3C

一貫性根拠は、次のことを実証しなければならない。

— セキュリティ課題定義のステートメントが、**PP**モジュール基盤のセキュリティ課題定義のステートメントと一貫している。

— セキュリティ課題定義のステートメントが、適合が主張されている機能パッケージのセキュリティ課題定義のステートメントと一貫している。

ACE_MCO.1.4C

一貫性根拠は、次のことを実証しなければならない。

— セキュリティ対策方針の定義が、**PP**モジュール基盤のセキュリティ対策方針と一貫していること。

— セキュリティ対策方針の定義が、適合が主張されている機能パッケージのセキュリティ対策方針と一貫していること。

ACE_MCO.1.5C

一貫性根拠は、次のことを実証しなければならない。

— セキュリティ機能要件の定義が、**PP**モジュール基盤のセキュリティ機能要件と一貫していること。

— セキュリティ機能要件の定義が、適合が主張されている機能パッケージのセキュリティ機能要件と一貫していること。

ACE_MCO.1.6C

保証根拠は、セキュリティ課題定義に関して、**PP**モジュールのセキュリティ保証要件のセットの内部的な一貫性を実証しなければならない。

ACE_MCO.1.7C

保証根拠は、**PP**モジュール基盤のセキュリティ保証要件に関して、**PP**モジュールのセキュリティ保証要件のセットの一貫性を実証しなければならない。

評価者アクションエレメント:

ACE_MCO.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。**PP**モジュールが別の**PP**モジュール基盤を特定する場合、評価者は各一貫性根拠に対してこのアクションを*実行しなければならない*。

8.9 PP構成一貫性(ACE_CCO)

8.9.1 目的

このファミリの目的は、PP構成の適格性と一貫性を決定することである。

8.9.2 ACE_CCO.1 PP構成一貫性

依存性： ACE_INT.1 PPモジュール概説

ACE_CCL.1 PPモジュール適合主張

ACE_SPD.1 PPモジュールセキュリティ課題定義

ACE_OBJ.1 PPモジュール運用環境のセキュリティ対策方針又はACE_OBJ.2 PPモジュールセキュリティ対策方針

ACE_ECD.1 PPモジュール拡張コンポーネント定義

ACE_REQ.1 PPモジュールの主張するセキュリティ要件又はACE_REQ.2 PPモジュールの導出されたセキュリティ要件

ACE_MCO.1 PPモジュール一貫性

APE_* (全てのAPEコンポーネント)

開発者アクションエレメント:

ACE_CCO.1.1D

開発者は、**PP構成**の参照を提供しなければならない。

ACE_CCO.1.2D

開発者は、コンポーネントステートメントを提供しなければならない。

ACE_CCO.1.3D

開発者は、**TOE**概要を提供しなければならない。

ACE_CCO.1.4D

開発者は、適合主張を提供しなければならない。

ACE_CCO.1.5D

開発者は、適合主張の中で適合ステートメントを提供しなければならない。

ACE_CCO.1.6D

開発者は、一貫性根拠を提供しなければならない。

ACE_CCO.1.7D

開発者は、**SAR**ステートメントを提供しなければならない。

ACE_CCO.1.8D

開発者は、**PP構成**に適用される評価方法及び/又は評価アクティビティのセットを提供しなければならない。

内容・提示エレメント:

ACE_CCO.1.1C

PP構成参照は、**PP構成**を一意に識別しなければならない。

ACE_CCO.1.2C

PP構成コンポーネントステートメントは、**PP構成**を構成する**PP**と**PP**モジュールを一意に識別しなければならない。

ACE_CCO.1.3C

PP構成コンポーネントステートメントで識別される各**PP**モジュールについて、コンポーネントステートメントは、識別された**PP**モジュールが必要とする**PP**モジュール基盤を含まなければならない。**PP**モジュールが別の**PP**モジュール基盤を特定する場合、これらの**PP**モジュール基盤のうち**1**つのみを、**PP構成**において参照しなければならない。

ACE_CCO.1.4C

マルチ保証PP構成の場合、コンポーネントステートメントは、PP構成に定義されたPP及びPPモジュールに定義されたサブTSFの観点から、TSFの構成を記述しなければならない。

ACE_CCO.1.5C

TOE概要は、TOE種別を識別しなければならない。

ACE_CCO.1.6C

TOE概要は、TOEの使用法及び主要なセキュリティ機能の特徴を記述しなければならない。

ACE_CCO.1.7C

TOE概要は、TOEが利用できるTOE以外のハードウェア/ソフトウェア/ファームウェアを識別しなければならない。

ACE_CCO.1.8C

適合主張は、PP構成コンポーネントが適合を主張するCCの版を識別しなければならない。

ACE_CCO.1.9C

適合主張は、CCパート2に対するPP構成の適合をCCパート2適合又はCCパート2拡張のいずれかとして記述しなければならない。

ACE_CCO.1.10C

適合主張は、CCパート3[vii]に対するPP構成の適合を「CCパート3適合」又は「CCパート3拡張」のいずれかとして記述しなければならない。

ACE_CCO.1.11C

適合主張は、PP構成コンポーネントの適合主張と一貫していなければならない。

ACE_CCO.1.12C

PP構成の適合主張は、PP構成の保証パッケージへの適合をパッケージ適合又はパッケージ追加として記述するステートメントからなる保証パッケージ適合主張を含まなければならない。

ACE_CCO.1.13C

適合ステートメントは、PP構成に要求される適合を、完全適合、正確適合又は論証適合のいずれかとして特定するか、又はPP構成の各コンポーネントに要求される適合種別のリストを提供しなければならない。

ACE_CCO.1.14C

完全適合の場合、PP構成のコンポーネントステートメントに含まれる各PPの適合ステートメントの併用許可ステートメントは、PP構成の全てのコンポーネントを、PP構成のPPと組み合わせて使用することが許可されているものとして、識別しなければならない。

ACE_CCO.1.15C

完全適合の場合、PP構成のコンポーネントステートメントに含まれる各PPモジュールの適合ステートメントの併用許可ステートメントは、その特定のPPモジュールのPPモジュール基盤にない全てのPP構成コンポーネントを、PP構成のPPモジュールと組み合わせて使用することが許可されているものとして、識別しなければならない。

ACE_CCO.1.16C

完全適合でないPP構成(すなわち、正確適合又は論証適合のPP構成)の場合、PP構成の適合ステートメントは、評価されているPP構成に適用できるCEMから派生した評価方法及び評

価アクティビティのセットを識別する、評価方法及び評価アクティビティの参照ステートメントを含むことができる。

ACE_CCO.1.17C

一貫性根拠は、**PP**構成で定義された**TOE**種別が、**PP**構成のコンポーネントステートメントに属する**PP**及び**PP**モジュールで定義された**TOE**種別と一貫していることを実証しなければならない。

ACE_CCO.1.18C

一貫性根拠は、**PP**構成コンポーネントに定義された全ての**SPD**、セキュリティ対策方針及びセキュリティ機能要件を合わせたものが一貫していることを実証しなければならない。

ACE_CCO.1.19C

単一保証**PP**構成では、**SAR**のステートメントは**TOE**全体に適用される単一の**SAR**のセットを定義しなければならない。正確及び論証適合の場合、**SAR**のセットは、**PP**構成コンポーネントのそれぞれで識別されている**SAR**を含まなければならない。完全適合の場合、**SAR**のセットは、**PP**構成コンポーネントのそれぞれで識別されている**SAR**のセットと同一でなければならない。

ACE_CCO.1.20C

マルチ保証**PP**構成の場合、**SAR**のステートメントは**TOE**全体に適用されるグローバルな**SAR**のセットと、各サブ**TSF**に適用される**SAR**を定義しなければならない。正確及び論証適合の場合、グローバルな**SAR**の保証セットは、**PP**構成コンポーネント間の共通**SAR**のセットを含み、サブ**TSF**に適用される**SAR**の各セットは、そのサブ**TSF**に関連する**PP**構成コンポーネントに識別されたものを含まなければならない。完全適合の場合、グローバルな**SAR**の保証セットは、**PP**構成コンポーネント間の共通**SAR**のセットであり、サブ**TSF**に適用される**SAR**の各セットは、そのサブ**TSF**に関連する**PP**構成コンポーネントに識別されたものと同一でなければならない。

ACE_CCO.1.21C

PP構成の**SAR**のステートメントには、評価されている**PP**構成コンポーネントで定義された**SAR**及び関連する評価方法と評価アクティビティにおける、適用される**SAR**のセットの一貫性を実証する保証の根拠を含まなければならない。マルチ保証**PP**構成の場合、保証の根拠は以下のことを実証しなければならない。

— グローバルな**SAR**のセットは、**PP**構成コンポーネントの**SPD**に定義された脅威と一貫したものであること。及び、

— グローバルな**SAR**のセットと各サブ**TSF**の**SAR**のセットが互いに一貫したものであること。

評価者アクションエレメント:

ACE_CCO.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACE_CCO.1.2E

評価者は、コンポーネントステートメントにおいて識別される全ての**PP**と**PP**モジュールで構成される**PP**構成に一貫性があることを*チェックしなければならない*。

9 ASEクラス: セキュリティターゲット(ST)評価

9.1 一般

STの評価は、STが信頼でき内部的に一貫していること、及びSTがPP構成、1つ又は複数のPP又はパッケージに基づいている場合に、それらのPP構成、PP及びパッケージをSTが正しく具体化していることを実証するために必要である。これらの特性は、STがTOE評価の基礎として使用するのに適しているために必要である。

9章は、CCパート1の附属書B、C及びDとともに使用されるべきである。これらの附属書は、ここでの概念を明確にし、多くの例を提供する。

図6は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。

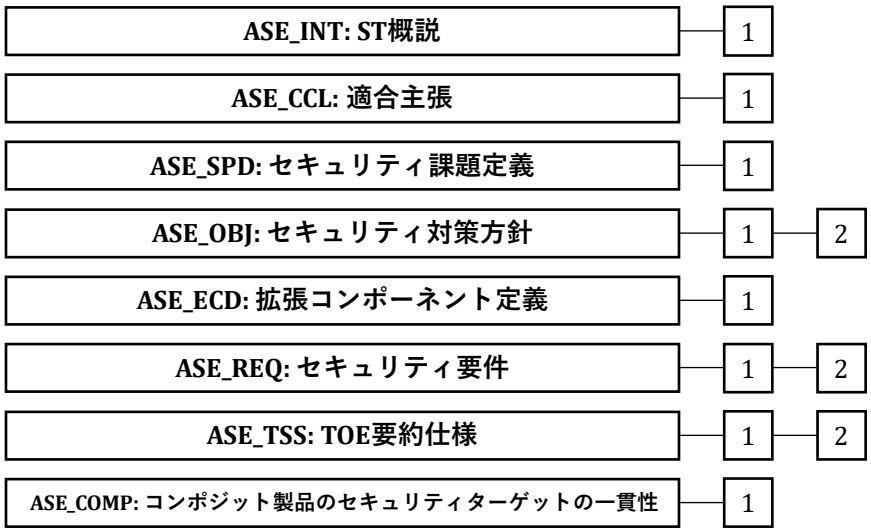


図 6 — ASE: セキュリティターゲット(ST)評価クラスのコンポーネント構成

9.2 ST 概説(ASE_INT)

9.2.1 目的

このファミリの目的は、TOEを、TOE参照、TOE概要、及びTOE記述の3つの抽象レベルで順序立てて記述することである。

ST概説の評価は、STとTOEが正しく識別されていること、TOEが3つの抽象レベルで正しく記述されていること、及びその3つの記述が互いに一貫していることを実証するために必要である。

9.2.2 ASE_INT.1 ST概説

依存性：なし

開発者アクションエレメント:

ASE_INT.1.1D

開発者は、ST概説を提供しなければならない。

内容・提示エレメント:

ASE_INT.1.1C

ST概説は、ST参照、TOE参照、TOE概要、及びTOE記述を含めなければならない。

ASE_INT.1.2C

ST参照は、STを一意に識別しなければならない。

ASE_INT.1.3C

TOE参照は、TOEを一意に識別しなければならない。

ASE_INT.1.4C

TOE概要は、TOEの使用法及び主要なセキュリティ機能の特徴を要約しなければならない。

ASE_INT.1.5C

TOE概要は、TOE種別を識別しなければならない。

ASE_INT.1.6C

TOE概要は、TOEに必要なTOE以外のハードウェア/ソフトウェア/ファームウェアを識別しなければならない。

ASE_INT.1.7C

マルチ保証STの場合、TOE概要は、STが適合を主張するPP構成に定義されるサブTSFの観点から、TSF構成を記述しなければならない。

ASE_INT.1.8C

TOE記述は、TOEの物理的範囲を記述しなければならない。

ASE_INT.1.9C

TOE記述は、TOEの論理的範囲を記述しなければならない。

評価者アクションエレメント:

ASE_INT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ASE_INT.1.2E

評価者は、TOE参照、TOE概要、及びTOE記述が相互に一貫していることを*確認しなければならない*。

9.3 適合主張(ASE_CCL)

9.3.1 目的

このファミリの目的は、適合主張の有効性を決定することである。さらに、このファミリは、STがPP又はPP構成に対する適合を主張する方法を特定する。

9.3.2 ASE_CCL.1 適合主張

依存性： ASE_INT.1 ST概説

ASE_ECD.1 拡張コンポーネント定義

ASE_REQ.1 直接根拠が主張されたセキュリティ要件

開発者アクションエレメント:

ASE_CCL.1.1D

開発者は、適合主張を提供しなければならない。

ASE_CCL.1.2D

開発者は、適合主張根拠を提供しなければならない。

内容・提示エレメント:

ASE_CCL.1.1C

適合主張は、STとTOEが適合を主張するCCの版を識別しなければならない。

ASE_CCL.1.2C

適合主張は、CCパート2に対するSTの適合をCCパート2適合又はCCパート2拡張のいずれかとして記述しなければならない。

ASE_CCL.1.3C

適合主張は、STの適合を「CCパート3適合」又は「CCパート3拡張」のいずれかとして記述しなければならない。

ASE_CCL.1.4C

適合主張は、拡張コンポーネント定義と一貫していなければならない。

ASE_CCL.1.5C

適合主張は、STが適合を主張するPP構成、又は全てのPP及びセキュリティ要件パッケージを識別しなければならない。

ASE_CCL.1.6C

適合主張は、パッケージへのSTの適合をパッケージ適合又はパッケージ追加のいずれかとして記述しなければならない。

ASE_CCL.1.7C

適合主張は、他のPPに対するSTの適合をPP適合として記述しなければならない。

ASE_CCL.1.8C

適合主張根拠は、TOE種別が、適合が主張されているPP構成又はPP内のTOE種別と一貫していることを実証しなければならない。

ASE_CCL.1.9C

適合主張根拠は、セキュリティ課題定義のステートメントが、適合が主張されているPP構成[1]、PP及び機能パッケージ内のセキュリティ課題定義のステートメントと一貫していることを実証しなければならない。

ASE_CCL.1.10C

適合主張根拠は、セキュリティ対策方針のステートメントが、適合が主張されているPP構成[2]、PP及び機能パッケージ内のセキュリティ対策方針のステートメントと一貫していることを実証しなければならない。

ASE_CCL.1.11C

---

[1] 実際には、PP 構成コンポーネントで定義された SPD の和集合を指す。

[2] 実際には、PP 構成コンポーネントで定義されたセキュリティ対策方針の和集合を指す。

適合主張根拠は、セキュリティ要件のステートメントが、適合が主張されているPP構成[3]、PP及び機能パッケージ内のセキュリティ要件のステートメントと一貫していることを実証しなければならない。

ASE_CCL.1.12C

PP又はPP構成の適合主張は、完全適合、正確適合又は論証適合、あるいは適合種別のリストでなければならない。

ASE_CCL.1.13C

適合主張が、TOEを評価するために使用されるCEMワークユニットから派生した評価方法と評価アクティビティのセットを特定する場合、このセットは、STが適合を主張するPP構成のパッケージ、PP又はPPモジュールに含まれるものを全て含み、それ以外を含んではならない。

評価者アクションエレメント:

ASE_CCL.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

9.4 セキュリティ課題定義(ASE_SPD)

9.4.1 目的

STのこの部分は、TOE及びTOEの運用環境によって対処されるセキュリティ課題を定義する。

セキュリティ課題定義の評価は、TOE及びTOEの運用環境によって対処されることが意図されているセキュリティ課題が明確に定義されていることを実証するために必要である。

9.4.2 ASE_SPD.1 セキュリティ課題定義

依存性：なし

開発者アクションエレメント:

ASE_SPD.1.1D

開発者は、セキュリティ課題定義を提供しなければならない。

内容・提示エレメント:

ASE_SPD.1.1C

セキュリティ課題定義は、脅威を記述しなければならない。

ASE_SPD.1.2C

全ての脅威は、脅威エージェント、資産、及び有害なアクションの観点から記述しなければならない。

ASE_SPD.1.3C

セキュリティ課題定義は、OSPを記述しなければならない。

ASE_SPD.1.4C

[3] 実際には、PP 構成コンポーネントで定義された SFR の和集合を指す。

セキュリティ課題定義は、TOEの運用環境についての前提条件を記述しなければならない。

評価者アクションエレメント:

ASE_SPD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

9.5 セキュリティ対策方針(ASE_OBJ)

9.5.1 目的

セキュリティ対策方針は、セキュリティ課題定義(ASE_SPD)ファミリを通して定義されるセキュリティ課題に対して意図される対応の簡潔なステートメントである。

セキュリティ対策方針の評価は、セキュリティ対策方針が適切かつ完全にセキュリティ課題定義を扱うこと、及びTOEとその運用環境の間でのこの課題に対する分担が明確に定義されていることを実証するために必要である。

9.5.2 コンポーネントのレベル付け

このファミリのコンポーネントは、運用環境のセキュリティ対策方針のみを記述しているのか(ASE_OBJ.1)、又はTOEのセキュリティ対策方針も含めて記述しているのか(ASE_OBJ.2)によって、レベル付けされている。

9.5.3 ASE_OBJ.1 運用環境のセキュリティ対策方針

依存性：なし

開発者アクションエレメント:

ASE_OBJ.1.1D

開発者は、運用環境のセキュリティ対策方針のステートメントを提供しなければならない。

ASE_OBJ.1.2D

開発者は、運用環境のセキュリティ対策方針根拠を提供しなければならない。

内容・提示エレメント:

ASE_OBJ.1.1C

セキュリティ対策方針のステートメントは、運用環境のセキュリティ対策方針を記述しなければならない。

ASE_OBJ.1.2C

セキュリティ対策方針根拠は、運用環境の各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威、そのセキュリティ対策方針によって実施されるOSP、及びそのセキュリティ対策方針によって充足される前提条件にまで、さかのぼって追跡しなければならない。

ASE_OBJ.1.3C

セキュリティ対策方針根拠は、運用環境のセキュリティ対策方針が全ての前提条件を充足することを実証しなければならない。

評価者アクションエレメント:

ASE_OBJ.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

9.5.4 ASE_OBJ.2 セキュリティ対策方針

依存性：ASE_SPD.1 セキュリティ課題定義

開発者アクションエレメント:

ASE_OBJ.2.1D

開発者は、セキュリティ対策方針のステートメントを提供しなければならない。

ASE_OBJ.2.2D

開発者は、セキュリティ対策方針根拠を提供しなければならない。

内容・提示エレメント:

ASE_OBJ.2.1C

セキュリティ対策方針のステートメントは、TOEのセキュリティ対策方針及び運用環境のセキュリティ対策方針を記述しなければならない。

ASE_OBJ.2.2C

セキュリティ対策方針根拠は、TOEの各セキュリティ対策方針をそのセキュリティ対策方針によって対抗される脅威及びそのセキュリティ対策方針によって実施されるOSPまでさかのぼって追跡しなければならない。

ASE_OBJ.2.3C

セキュリティ対策方針根拠は、運用環境の各セキュリティ対策方針を、そのセキュリティ対策方針によって対抗される脅威、そのセキュリティ対策方針によって実施されるOSP、及びそのセキュリティ対策方針によって充足される前提条件にまで、さかのぼって追跡しなければならない。

ASE_OBJ.2.4C

セキュリティ対策方針**根拠**は、セキュリティ対策方針が**全ての脅威に対抗することを実証**しなければならない。

ASE_OBJ.2.5C

セキュリティ対策方針根拠は、セキュリティ対策方針が全てのOSPを実施することを実証しなければならない。

ASE_OBJ.2.6C

セキュリティ対策方針根拠は、運用環境のセキュリティ対策方針が全ての前提条件を充足することを実証しなければならない。

評価者アクションエレメント:

ASE_OBJ.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

9.6 拡張コンポーネント定義(ASE_ECD)

9.6.1 目的

拡張セキュリティ要件は、CCパート2又はこの文書のコンポーネントではなく、拡張コンポーネント、つまりSTの作成者によって定義されるコンポーネントに基づく要件である。

拡張コンポーネント定義の評価は、拡張コンポーネントが明確で曖昧さがないこと、及びそれらが必要であること、つまり既存のCCパート2又はこの文書のコンポーネントを使用して明確には表現できないことを決定するために必要である。

9.6.2 ASE_ECD.1 拡張コンポーネント定義

依存性：なし

開発者アクションエレメント:

ASE_ECD.1.1D

開発者は、セキュリティ要件のステートメントを提供しなければならない。

ASE_ECD.1.2D

開発者は、拡張コンポーネント定義を提供しなければならない。

内容・提示エレメント:

ASE_ECD.1.1C

セキュリティ要件のステートメントは、全ての拡張セキュリティ要件を識別しなければならない。

ASE_ECD.1.2C

拡張コンポーネント定義は、各拡張セキュリティ要件に対応する拡張コンポーネントを定義しなければならない。

ASE_ECD.1.3C

拡張コンポーネント定義は、各拡張コンポーネントが既存のCCコンポーネント、ファミリ、及びクラスにどのように関連するかを記述しなければならない。

ASE_ECD.1.4C

拡張コンポーネント定義は、提示モデルとして既存のCCコンポーネント、ファミリ、クラス、及び方法を使用しなければならない。

ASE_ECD.1.5C

拡張コンポーネントは、エレメントに対する適合又は非適合を実証できるように、評価可能で客観的なエレメントで構成されていなければならない。

評価者アクションエレメント:

ASE_ECD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ASE_ECD.1.2E

評価者は、拡張コンポーネントが既存のコンポーネントを使用して明確には表現できないことを*確認しなければならない*。

9.7 セキュリティ要件(ASE_REQ)

9.7.1 目的

SFRは、TOEに期待されるセキュリティのふるまいについての、明確で曖昧さがなく十分に定義された記述となる。SARは、TOEで保証を得るために採用される期待されるアクティビティについての、明確で曖昧さのない標準的な記述となる。

セキュリティ要件の評価は、それらの要件が明確で曖昧さがなく十分に定義されていることを保証するために必要である。

9.7.2 コンポーネントのレベル付け

このファミリのコンポーネントは、そのままで主張されているのか(ASE_REQ.1参照)、又はSFRがTOEのセキュリティ対策方針から導き出されているのか(ASE_REQ.2参照)によって、レベル付けされている。

9.7.3 ASE_REQ.1 直接根拠セキュリティ要件

依存性：ASE_ECD.1 拡張コンポーネント定義

開発者アクションエレメント:

ASE_REQ.1.1D

開発者は、セキュリティ要件のステートメントを提供しなければならない。

ASE_REQ.1.2D

開発者は、セキュリティ要件根拠を提供しなければならない。

内容・提示エレメント:

ASE_REQ.1.1C

セキュリティ要件のステートメントは、SFR及びSARを記述しなければならない。

ASE_REQ.1.2C

単一保証のSTの場合、セキュリティ要件のステートメントは、TOE全体に適用されるグローバルなSARのセットを定義しなければならない。SARのセットは、STが適合を主張するPP又はPP構成と一貫していなければならない。

ASE_REQ.1.3C

マルチ保証STの場合、セキュリティ要件のステートメントは、TOE全体に適用されるグローバルなSARのセットと、各サブTSFに適用されるSARのセットを定義しなければならない。SARのセットは、STが適合を主張するマルチ保証PP構成と一貫していなければならない。

ASE_REQ.1.4C

SFR及びSARで使用される全てのサブジェクト、オブジェクト、操作、セキュリティ属性、外部のエンティティ及びその他の用語が定義されなければならない。

ASE_REQ.1.5C

セキュリティ要件のステートメントは、セキュリティ要件の全ての操作を識別しなければならない。

ASE_REQ.1.6C

全ての操作は正しく実行しなければならない。

ASE_REQ.1.7C

セキュリティ要件の各依存性が満たされていなければならない。また、満たされない依存性がある場合は、セキュリティ要件根拠によってそのことが正当化されなければならない。

ASE_REQ.1.8C

セキュリティ要件根拠は、**SFR**が（運用環境のセキュリティ対策方針と合わせて）**TOE**の全ての脅威に対抗していることを実証しなければならない。

ASE_REQ.1.9C

セキュリティ要件根拠は、**SFR**が（運用環境のセキュリティ対策方針と合わせて）**OSP**の全てを実施することを実証しなければならない。

ASE_REQ.1.10C

セキュリティ要件根拠は、なぜ**SAR**が選ばれたかを説明しなければならない。

ASE_REQ.1.11C

セキュリティ要件のステートメントは、内部的に一貫していなければならない。

ASE_REQ.1.12C

STが**SAR**のセットを定義し、それが**PP**又は適合を主張する**PP**構成の**SAR**のセットを拡張する場合、セキュリティ要件根拠は、拡張の一貫性を正当化する保証根拠を含み、**SAR**のセットの拡張により影響を受ける、適合ステートメントで特定されるあらゆる評価方法/評価アクティビティの処理に関する根拠を提供しなければならない。

評価者アクションエレメント:

ASE_REQ.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

9.7.4 ASE_REQ.2 導出されたセキュリティ要件

依存性： ASE_OBJ.2 セキュリティ対策方針

ASE_ECD.1 拡張コンポーネント定義

開発者アクションエレメント:

ASE_REQ.2.1D

開発者は、セキュリティ要件のステートメントを提供しなければならない。

ASE_REQ.2.2D

開発者は、セキュリティ要件根拠を提供しなければならない。

内容・提示エレメント:

ASE_REQ.2.1C

セキュリティ要件のステートメントは、SFR及びSARを記述しなければならない。

ASE_REQ.2.2C

単一保証のSTの場合、セキュリティ要件のステートメントは、TOE全体に適用されるグローバルなSARのセットを定義しなければならない。SARのセットは、STが適合を主張するPP又はPP構成と一貫していなければならない。

ASE_REQ.2.3C

マルチ保証STの場合、セキュリティ要件のステートメントは、TOE全体に適用されるグローバルなSARのセットと、各サブTSFに適用されるSARのセットを定義しなければならない。SARのセットは、STが適合を主張するマルチ保証PP構成と一貫していなければならない。

ASE_REQ.2.4C

SFR及びSARで使用される全てのサブジェクト、オブジェクト、操作、セキュリティ属性、外部のエンティティ及びその他の用語が定義されなければならない。

ASE_REQ.2.5C

セキュリティ要件のステートメントは、セキュリティ要件の全ての操作を識別しなければならない。

ASE_REQ.2.6C

全ての操作は正しく実行しなければならない。

ASE_REQ.2.7C

セキュリティ要件の各依存性が満たされていなければならない。また、満たされない依存性がある場合は、セキュリティ要件根拠によってそのことが正当化されなければならない。

ASE_REQ.2.8C

セキュリティ要件根拠は、SFRがTOEのセキュリティ対策方針の全てを満たすことを実証しなければならない。

ASE_REQ.2.9C

セキュリティ要件根拠は、なぜSARが選ばれたかを説明しなければならない。

ASE_REQ.2.10C

セキュリティ要件のステートメントは、内部的に一貫していなければならない。

ASE_REQ.2.11C

STが、適合を主張するPP又はPP構成のSARのセットを拡張するSARのセットを定義する場合、セキュリティ要件根拠は、拡張の一貫性を正当化する保証根拠を含み、SARのセットの拡張によって影響を受ける適合ステートメントにおいて特定されるあらゆる評価方法と評価アクティビティの処理に関する根拠を提供しなければならない。

評価者アクションエレメント:

ASE_REQ.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

9.8 TOE要約仕様(ASE_TSS)

9.8.1 目的

TOE要約仕様は、TOEがどのように実装されているかについて、評価者と潜在的消費者が概要を理解できるようにする。

TOE要約仕様の評価は、以下の点について適切に記述されているかどうかを決定するために必要である。

— TOEがSFRをどのように満たすか

— TOEが干渉、論理的な改ざん及びバイパスに対して自身をどのように保護するか

また、TOE要約仕様がTOEの他の叙述的記述と一貫しているかどうかを確認するためにも必要である。

9.8.2 コンポーネントのレベル付け

このファミリのコンポーネントは、TOE要約仕様が、TOEがどのようにSFRを満たすかのみを記述するために必要かどうか、又はTOEが論理的な改ざんやバイパスから自身をどのように保護するか記述するためにも必要かどうかによってレベル付けされる。この追加の記述は、TOEセキュリティアーキテクチャに関する特定の問題が発生する可能性がある特殊な状況で使用することができる。

9.8.3 ASE_TSS.1 TOE要約仕様

依存性： ASE_INT.1 ST概説

ASE_REQ.1 直接根拠で主張されたセキュリティ要件

ADV_FSP.1 基本機能仕様

開発者アクションエレメント:

ASE_TSS.1.1D

開発者は、TOE要約仕様を提供しなければならない。

内容・提示エレメント:

ASE_TSS.1.1C

TOE要約仕様は、TOEがどのように各SFRを満たすかを記述しなければならない。

評価者アクションエレメント:

ASE_TSS.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ASE_TSS.1.2E

評価者は、TOE要約仕様がTOE概要及びTOE記述と一貫していることを*確認しなければならない*。

9.8.4 ASE_TSS.2 アーキテクチャ設計要約を伴うTOE要約仕様

依存性： ASE_INT.1 ST概説

ASE_REQ.1 直接根拠で主張されたセキュリティ要件

ADV_ARC.1 セキュリティアーキテクチャ記述

開発者アクションエレメント:

ASE_TSS.2.1D

開発者は、TOE要約仕様を提供しなければならない。

内容・提示エレメント:

ASE_TSS.2.1C

TOE要約仕様は、TOEがどのように各SFRを満たすかを記述しなければならない。

ASE_TSS.2.2C

TOE要約仕様は、TOEがどのように干渉や論理的な改ざんから自身を保護するかを記述しなければならない。

ASE_TSS.2.3C

TOE要約仕様は、TOEがどのようにバイパスから自身を保護するかを記述しなければならない。

評価者アクションエレメント:

ASE_TSS.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ASE_TSS.2.2E

評価者は、TOE要約仕様がTOE概要及びTOE記述と一貫していることを*確認しなければならない*。

9.9 コンポジット製品のセキュリティターゲットの一貫性(ASE_COMP)

9.9.1 目的

このファミリの目的は、コンポジット製品のST[4]が、関連する基本コンポーネントのST[5,6]と矛盾しないかどうかを決定することである。

9.9.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

9.9.3 適用上の注釈

コンポジット製品のSTを作成し、評価しなければならない。

コンポジット製品評価者は、コンポジット製品のSTが、関連する基本コンポーネントのSTと矛盾していないことを検査しなければならない。具体的には、コンポジット製品評価者は、コンポジット製品のSTと基本コンポーネントのSTについて、前提条件の矛盾、依存コンポーネントが必要とするセキュリティ対策方針、セキュリティ要件及びセキュリティ機能性の互換性を検査しなければならないことを意味する。

コンポジット製品評価スポンサーは、基本コンポーネントのSTが依存コンポーネント開発者、コンポジット製品評価者、コンポジット製品評価監督機関に対して利用可能であることを保証しなければならない。基本コンポーネントのSTの公開版で入手できる情報では、十分でない場合がある。

この適用上の注釈は、開発者がコンポジット製品STを作成し、評価者がコンポジット製品STを分析することを支援し、そのための一般的な方法を記述する。

コンポジット製品STを作成するために、開発者は以下のステップを実行するべきである。

---

[4] 以下では、コンポジット製品セキュリティターゲット又はコンポジット ST と表記する。

[5] 以下では、基本コンポーネントセキュリティターゲット又は基本 ST と表記する。

[6] 一般に、セキュリティターゲットは定義された TOE のセキュリティ方針を表現する。

ステップ1：開発者は、標準実施基準を用いて、コンポジット製品の予備ST（コンポジットST）を策定する。コンポジットSTは、少なくとも正式なPP適合主張がない限り、コンポジット製品の関連基本コンポーネントのST（基本ST）とは無関係に策定することができる。

ステップ2：開発者は、基本STとコンポジットSTの間の重複を、それぞれのTOEセキュリティ機能性（TSF）を分析し比較することにより決定する。[7] [8]

ステップ3：開発者は、どのような条件下で、新たに検査することなく、基本コンポーネントTSFが複合STで使用されていることを信用し、信頼できるかを決定する。

これらのステップを経て、開発者はコンポジット製品の予備STを完成させる。

コンポジット製品とそれに関連する基本コンポーネントが、CCの同じ版に従って評価されることは必須ではない。これは、(i)基本コンポーネントの保証レベルがコンポジット製品の意図する保証レベルをカバーしており、(ii)基本コンポーネントの評価が有効で（すなわち、基本コンポーネント評価監督機関に受け入れられて）かつ最新であれば、コンポジット製品の依存コンポーネントが基本コンポーネントの一部のセキュリティサービスに依存できるという事実によるものである。異なるCCの版に属する単一保証コンポーネント（ひいては保証レベル）の同等性は、コンポジット製品評価監督機関により確立/承認されなければならない。

PPへの適合が主張される場合、例えば、コンポジット製品STがPPへの適合を主張する場合（さらなるPPへの適合を主張する場合もある）、一貫性のチェックは、これらのPPによって既にカバーされていないSTのエレメントのみのチェックにまで軽減することができる。しかし、一般に、PPに適合しているという事実は、矛盾を回避するのに十分ではない。次のような状況を想定する。ここで。→は「適合」を表す。

コンポジットST → PP1 → PP2 ← 基本ST

PP1はどのような適合種別[9]を要求してもよいが、基本STがPP2に加えて導入する可能性のある「追加要素」には影響を与えない。結論として、これらの追加要素は、PP1に加えて選択されたコンポジットSTの追加要素と必ずしも一貫したものではない。それらの一貫性を「構造によって」保証するシナリオは存在しない。

一貫性は直接的なマッチングではない可能性があることに注意。例えば、基本コンポーネントの環境に関する目標が、コンポジットTOEの目標になる可能性がある。

9.9.4 ASE_COMP.1 セキュリティターゲット(ST)の一貫性

依存性：なし

開発者アクションエレメント:

ASE_COMP.1.1D

開発者は、コンポジット製品セキュリティターゲットと基本コンポーネントセキュリティターゲットの間の互換性に関するステートメントを提供しなければならない。このステートメントは、コンポジット製品セキュリティターゲット内で提供されてもよい。

---

[7] なぜなら、TSF は(TOE の運用環境のセキュリティ対策方針を実施する組織的な対策とともに)セキュリティターゲットを実施するからである。

[8] 比較は、SFR の抽象化のレベルで行われなければならない。開発者がセキュリティターゲットの TSS でセキュリティ機能性のグループ(TSF グループ)を定義した場合、評価者は、TOE が提供するセキュリティサービスのコンテキストをよりよく理解するために、それらも考慮しなければならない。

[9] 例えば、CC に従った「正確」、「完全」又は「論証」。

内容・提示エレメント:

ASE_COMP.1.1C

互換性のステートメントは、基本コンポーネントの**TSF**を、コンポジット製品セキュリティターゲットなどで使用されている関連する基本コンポーネントの**TSF**に分離することを記述しなければならない。

ASE_COMP.1.2C

コンポジット製品セキュリティターゲットと基本コンポーネントセキュリティターゲットの間の互換性に関するステートメントは、コンポジット製品のセキュリティターゲットと関連する基本コンポーネントのセキュリティターゲットが一致すること、すなわち、コンポジット製品セキュリティターゲットと基本コンポーネントセキュリティターゲットのセキュリティ環境、セキュリティ対策方針及びセキュリティ要件の間に矛盾がないことを(例えば、マッピングの形で)示さなければならない。これは、コンポジット製品セキュリティターゲットに関連するエレメントを直接示し、必要に応じて説明文を添えることで提供することができる。

評価者アクションエレメント:

ASE_COMP.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

10 ADVクラス: 開発

10.1 一般

開発クラスの要件は、TOEに関する情報を提供する。この情報から得られた知識は、AVAクラス及びATEクラスで記述されているTOEに対する脆弱性分析とテストを実施するための基礎として使用される。

開発クラスは、抽象化の様々なレベル及び形式でTSFを構造化し、表現するための要件で構成される7つのファミリを含んでいる。これらのファミリには、次のものが含まれる:

— SFRの設計及び実装の(様々な抽象レベルでの)記述に関する要件(ADV_FSP、ADV_TDS、ADV_IMP、ADV_COMP)

— ドメイン分離、TSFの自己保護、及びセキュリティ機能性の非バイパス性というアーキテクチャ指向の特徴の記述に関する要件(ADV_ARC)

— セキュリティ方針モデルに関する要件、及びセキュリティ方針モデルと機能仕様の間の対応付けに関する要件(ADV_SPM)

— モジュール化、階層化、複雑さの最小化などの側面に対応するTSFの内部構造に関する要件(ADV_INT)

TOEのセキュリティ機能性について証拠資料を提出する際に、2つの特性を実証する必要がある。第1の特性は、セキュリティ機能性が正しく機能すること、つまり仕様どおりに動作することである。第2の特性は、おそらく簡単には実証できないが、セキュリティ機能性が破壊又はバイパスされかねない方法ではTOEを使用できないようになっていることである。この2つの特性の分析にはやや異なるアプローチが必要になるため、ADVのファミリはそれらのアプローチをサポートするように構成されている。機能仕様(ADV_FSP)、TOE設計(ADV_TDS)、実装表現(ADV_IMP)、セキュリティ方針モデル化(ADV_SPM)の各ファミリは、第1の特性つまりセキュリティ機能性の仕様を扱う。セキュリティアーキテクチャ(ADV_ARC)及びTSF内部構造(ADV_INT)の各ファミリは、第2の特性、つまりセキュリティ機能性が破壊又はバイパスされないことを実証するTOE設計の仕様を扱う。どちらの特性も実現される必要があることに注意するべきである。つまり、特性が満たされているという信頼が高いほど、TOEの信頼も高くなる。コンポジット製品のTSFは、ADVクラスのファミリにおいて、様々な抽象度で表現される。コンポジット設計適合(ADV_COMP)のファミリは、関連する基本コンポーネントから課される依存コンポーネントの要件が、コンポジット製品において満たされているかどうかを決定する。ADVクラスのファミリでは、コンポジット製品のTSFが様々なレベルに分散しているため、このファミリは図7に表れない。ファミリ内のコンポーネントは、コンポーネントの階層が上がるにつれて、より多くの保証が得られるように設計されている。

第1の特性を対象とするファミリのパラダイムは、設計分解の1つである。最上位レベルには、TSFのインタフェースに関するTSFの機能仕様がある(TSFに対するサービスの要求及びその結果の応答に関してTSFが*何を*行うかが記述される)。TSFが(求められる保証とTOEの複雑さに応じて)より小さな単位に分解され、TSFがその機能を*どのように*果たすかが(保証レベルに合った詳細レベルまで)記述され、TSFの実装が示される。セキュリティのふるまいの形式的なモデルが示される場合もある。分解の全てのレベルが、その他全てのレベルの完全性と正確さの決定に使用され、それによってレベルの相互サポートが保証される。種々のTSF表現に関する要件は、複数のファミリに分けられて、PP/STの作成者が必要なTSF表現を

特定できるようになっている。選択されたレベルによって、必要な保証/得られる保証が決まる。

図7は、ADVクラスの各種TSF表現、及びそれらと他のクラスとの関係を示している。この図が示すように、APEクラスとASEクラスは、SFRとTOEセキュリティ対策方針の対応に関する要件を定義する。また、ASEクラスは、セキュリティ対策方針とSFR、及びどのようにTOEがSFRを満たすか説明するTOE要約仕様間の対応に関する要件についても定義する。ALC_CMC.5.2Eのアクティビティには、ATEクラス及びAVAクラスでテストされるTSFが実際に全てのADV分解レベルで記述されているかどうかの検証が含まれる。

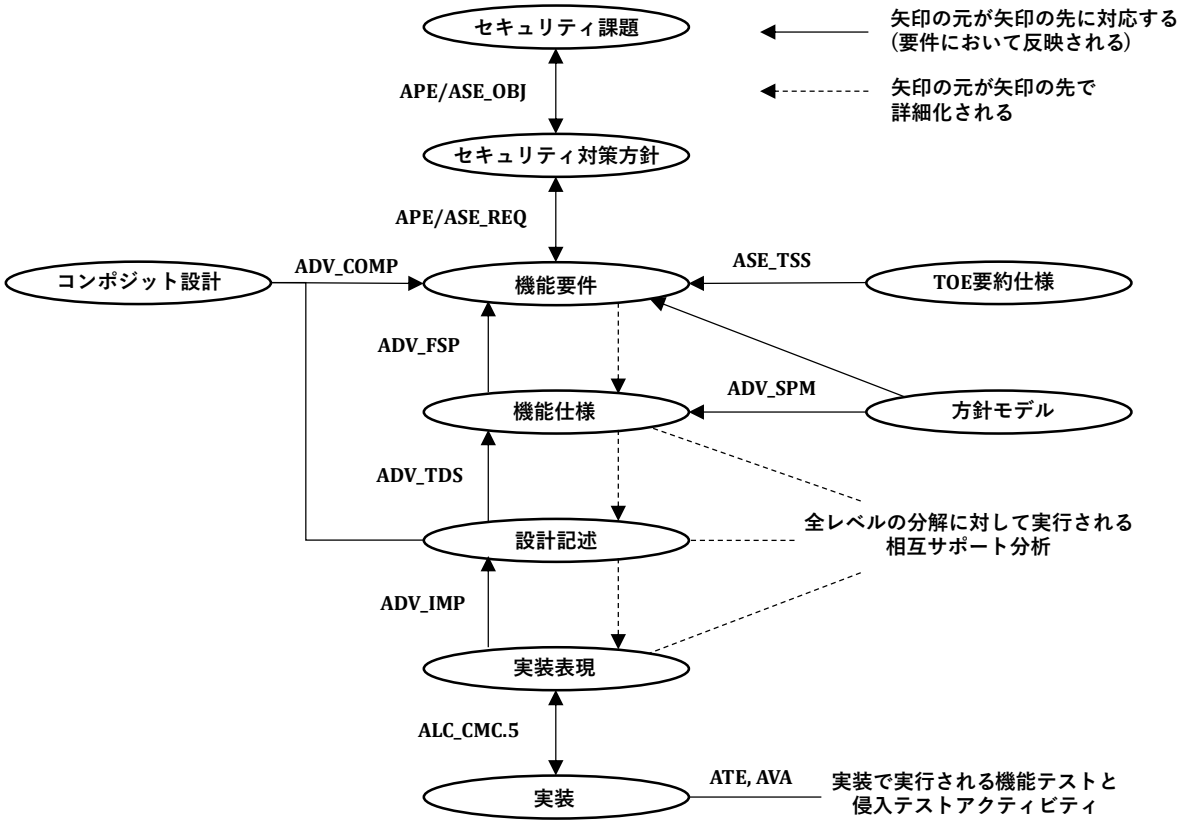


図 7 — ADV 構造間及びそれらと他のファミリとの関係

これ以外の図 7 で示される対応関係は、全て TOE の ADV クラスで定義される。セキュリティ方針モデル化(ADV_SPM)ファミリは、選択された SFR セキュリティ機能を形式的にモデル化するための要件、及び機能仕様と形式的モデルの間の対応を提供するための要件を定義する。TSF 表現に焦点を当てた各保証ファミリ(つまり、機能仕様(ADV_FSP)、TOE 設計(ADV_TDS)、及び実装表現(ADV_IMP))は、その TSF 表現を SFR に関係付ける要件を定義する。全ての分解は他の全ての分解を正確に反映する(つまり相互サポートする)必要がある。開発者はコンポーネントの最後の.C エレメントでの追跡を提供する。この要因に関連する保証は、特定レベルの分解が分析される間に、他のレベルの分解を(再帰的に)参照する方法で各レベルの分解を分析することによって得られる。評価者は 2 つ目の E エレメントの一部として対応を検証する。これらのレベルの分解から得られた理解は、機能テスト及び侵入テスト成果の基礎となる。

ADV_INTファミリは、TSFの内部構造に関連し、TSF表現の詳細化のプロセスには間接的にしか関連していないため、この図には示されていない。同様に、ADV_ARCファミリは、TSFの表現ではなくそのアーキテクチャの健全性に関連しているため、この図に示されていない。

ADV_INTとADV_ARCはどちらも、TOEがそのセキュリティ機能性を回避又は破壊できないようになっているという特性の分析に関連する。

TOEセキュリティ機能(TSF)は、SFRの実施に必要となるTOEの全ての部分で構成される。TSFは、SFRを直接実施する機能性、及び、SFRを侵害する可能性のある機能性も含めた、SFRを直接実施しないが間接的にSFRの実施に寄与する機能性を含んでいる。これには、立ち上げ時に呼び出され、TSFをそのセキュアな初期状態にするTOEの各部分が含まれる。

ADVファミリのコンポーネントの開発には、いくつかの重要な概念が採用された。これらの概念については、この節で簡単に紹介し、ファミリの適用上の注釈で詳しく説明する。

最も重要な概念の1つは、提供される情報が多いほど、セキュリティ機能性がa)正しく実装され、b)損なわれず、c)バイパスされないことの保証が高まることである。これは、証拠資料が正確で他の証拠資料と一貫していることを検証し、テストアクティビティ(機能テストと侵入テストの両方)が包括的であることを保証するための情報を提供することで実現される。これは、ファミリのコンポーネントのレベル付けで反映される。一般に、コンポーネントは、提供される(さらにその後分析される)情報の量に基づいてレベル付けされる。

全てのTOEには当てはまらないが、TSFが複雑であるために、TSFのある部分に他の部分よりも厳しい検査が必要になることは一般的である。このような部分の判別はあいにくやや主観的であるため、保証レベルの増大に伴って、詳細な検査の必要なTSFの部分を判別する責任が開発者から評価者にシフトするように、用語及びコンポーネントが定義されている。この概念を表すのに役立つように、次の用語が導入されている。このクラスのファミリでは、TOEのSFRに関連する部分を表す際に、(つまり、機能仕様(ADV_FSP)、TOE設計(ADV_TDS)、実装表現(ADV_IMP)の各ファミリに統合されているエレメントとワークユニットで)この用語が使用されることに注意するべきである。他のファミリには一般的な概念(TOEの一部分に対する*関心*が他の部分より高いという概念)が適用されるが、その基準は、必要な保証を得るために異なる方法で表される。

TSFの全ての部分は*セキュリティに関連*している。これは、それらの部分が、SFR及びドメイン分離と非バイパス性に関する要件で表されているとおりにTOEのセキュリティを保持しなければならないことを意味する。セキュリティ関連性の側面の1つに、TSFの一部分がセキュリティ要件を実施する程度がある。TOEの各部分が、セキュリティ要件の実施においてそれぞれに異なる役割を果たす(あるいは明らかな役割がない)ため、このことによってSFRの関連性の連続体が作成される。この連続体の一方の終端は、*SFR実施*と呼ばれるTOEの部分である。このような各部分は、あらゆるSFRをTOEに実装するために直接的な役割を果たす。ここで言うSFRとは、STに含まれているSFRのいずれかによって提供される機能性を指す。SFR実施機能性での*役割を果たす*という表現の定義は、定量的に表すことが不可能であることに注意するべきである。例えば、任意アクセス制御(DAC)メカニズムの実装の場合、ごく狭い視野で見た*SFR実施*は、オブジェクトの属性に対してサブジェクトの属性を実際にチェックする数行のコードである。より広い視野で見ると、その数行のコードを含んだソフトウェアエンティティ(例えばC関数)が含まれる。より広い視野には、C関数のコール元も含まれる。これは、属性チェックによって返された決定を実施する責任をそれらのコール元が負うためである。さらに広い視野には、そのC関数のコールツリー(又は使用される実装言語における同等のプログラミング構造)内のコードが含まれる(例えば、ファーストマッチアルゴリズムの実装でアクセス制御リストのエントリをソートするソート機能)。ある点で、コンポーネントはセキュリティ方針の*実施*にそれほど関与しなくなり、*サポート*の役割を果たすようになる。このようなコンポーネントは*SFR支援*と呼ばれる。SFR支援機能性の特性の1つは、誤りなく動作することで正しいSFR実装を保つと信頼されている点である。この機能性にSFR実施機能性が依存している場合もあるが、一般にその依存性は、メモリ管理やバッファ管理などの機能レベルである。セキュリティ関連性の連続体における次の機能性は、

*SFR非干渉*と呼ばれる。この機能性は、SFRの実装では役割がなく、その環境のためにTSFの一部分である場合が多い。例えば、オペレーティングシステム上で特権的なハードウェアモードで実行されるコードが挙げられる。このコードが損なわれると(あるいは悪意のあるコードで置き換えられると)、それが特権的なハードウェアモードで動作することによってSFRの正しい動作が損なわれる可能性があるため、このコードはTSFの一部とみなす必要がある。SFR非干渉機能性の例には、処理速度を考慮してカーネルモードに実装される一連の浮動小数点演算が挙げられる。

アーキテクチャファミリ(セキュリティアーキテクチャ(ADV_ARC))は、ドメイン分離、自己保護、及び非バイパス性の特性に基づいた、TOEの要件と分析を提供する。これらの特性は、存在しない場合、SFRを実装しているメカニズムの障害につながる可能性があるという点で、SFRに関連している。これらの特性に関連する機能性と設計は、その性質と分析要件が根本的に異なっていることから、上で説明した連続体の一部とはみなされず、別個に扱われる。

SFRの実装(SFR実施機能性及びSFR支援機能性)と、初期化、自己保護、非バイパス性に関係する、どちらかというと基本的なTOEのセキュリティ特性の実装との分析における違いは、SFR関連機能性がある程度直接的に見え、比較的テストが容易であるのに対し、上で説明した特性には、はるかに広い範囲の機能性セットに対して様々な程度の分析が必要であるという点である。さらに、このような特性の分析の深さは、TOEの設計によって異なる。ADVファミリは、初期化、自己保護、及び非バイパス性の各要件の分析のみを行う別のファミリ(セキュリティアーキテクチャ(ADV_ARC))でこれに対応し、その他のファミリでSFRを支援する機能性の分析を行うように構成されている。

複数の抽象レベルに別々の記述が必要な場合であっても、個々のTSF表現を別個の文書として記述する必要はまったくない。実際、要求されているのはこれらのTSF表現についての情報であって、結果としての文書構造ではないために、1つの文書が複数のTSF表現に対する証拠資料要件に合致する場合もある。複数のTSF表現が1つの文書中に混在している場合、開発者は、文書のどの部分が、どの要件に合致しているかを示すべきである。

3種の仕様の様式(非形式的、準形式的、及び形式的)がこのクラスによって指定される。機能仕様及びTOE設計証拠資料は、常に非形式的又は準形式的のいずれかの様式で記述される。準形式的な様式は、非形式的表現に比べて、これらの文書での曖昧さが少ない。準形式的表現に*加えて*形式的仕様が必要となる場合がある。その意義は、TSFが複数の方法で記述されることにより、TSFが完全かつ正確に特定されているというさらなる保証が得られるという点にある。

非形式的仕様とは、自然言語によって普通に書かれる。ここで言う自然言語とは、(スペイン語や、ドイツ語、フランス語、英語、オランダ語など)通常の会話で用いられる言葉を示している。非形式的仕様は、その言語で通常用いられている(例: 文法や構文)規則として要求されること以外、表記や特別な制約は課されない。表記に関する制約は課されないものの、非形式的仕様では、文脈上、通常用いられる意味と異なる場合には、定められている用語の意味が定義されていなければならない。

準形式的文書と非形式的文書の違いは、形式/表現の点のみである。準形式的表記には、例えば、明示的な用語集や標準化された表現形式が含まれる。準形式的仕様は、標準の表現テンプレートに書き込まれる。自然言語で記述される場合は、表現で用語が矛盾なく使用されるべきである。表現では、より構造化された言語/図が使用される場合もある(例えば、データフロー図、状態遷移図、E-R図、データ構造図、プロセスやプログラムの構造図)。図又は自然言語のどちらに基づいている場合でも、表現では一連の規則を使用しなければならない。用語集は、正確かつ一定して使用される単語を明示的に識別する。同様に、標準化された形式は、できる限り明確になるように文書を方式的に準備することに、最大限の注意が払われたことを示す。TSFの基本的に異なっている部分は、その準形式的表記規則及び表現様式が

異なっていることがあるので注意するべきである(ただし、異なっている「準形式的表記」の数が少ない場合)。この点はまだ*準形式的表現*の概念に適合している。

形式的仕様とは、数学的概念に基づいた表記によって書かれ、これに(非形式的な)補足説明が加わっているようなものをいう。これらの数学的概念は、表記の構文と意味、及び論理的な推論を助ける証明規則を定義するために用いられる。形式的表記をサポートする構文意味規則は、どのようにして曖昧さなくその構造を認識し、その意味を決定付けるかを定義するべきである。矛盾を引き出すのが不可能であることの証拠が必要であり、表記をサポートする全ての規則を定義又は参照付けする必要がある。

図8は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。

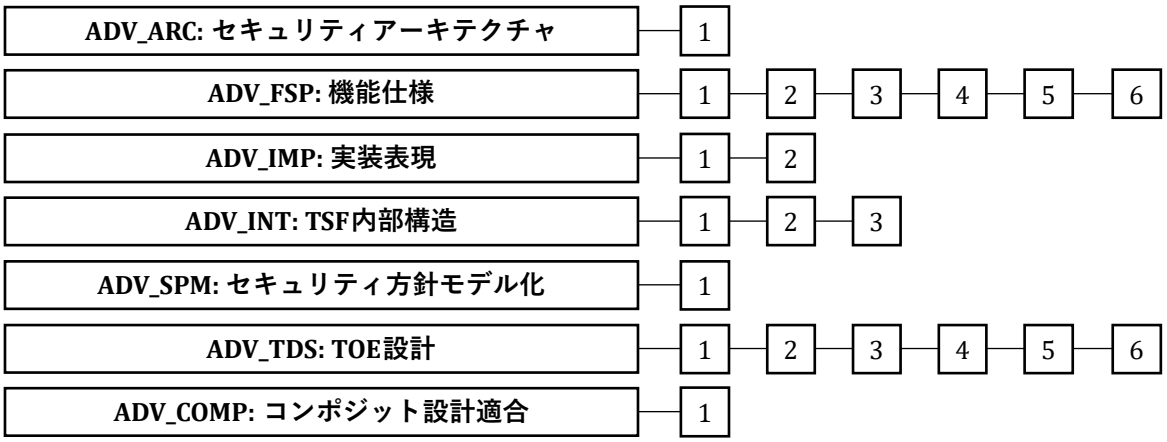


図 8 — ADV: 開発クラスのコンポーネント構成

マルチ保証評価の場合、SFRの設計及び実装の(様々な抽象レベルでの)記述に関する要件(ADV_FSP、ADV_TDS、ADV_IMP、ADV_COMP)はTOEの**サブTSF**に対して提示されることになる。アーキテクチャファミリ(セキュリティアーキテクチャ(ADV_ARC))は、ドメイン分離、自己保護、及び非バイパス性の特性に基づいた、TOEの要件と分析を提供する。その特性は**サブTSF**間の境界にも適用される可能性がある。

10.2 セキュリティアーキテクチャ(ADV_ARC)

10.2.1 目的

このファミリの目的は、TSFのセキュリティアーキテクチャの記述を、開発者が提供することである。この証拠にTSFに対して提示されるその他の証拠が加味されると、TSFが目的の特性を達成したことを確認する情報の分析が可能となる。セキュリティアーキテクチャの記述は、TOEのセキュリティ分析はTSFの検査により達成できるという暗黙的な主張をサポートする。適切なアーキテクチャがない場合は、TOEの機能性全体を調査する必要がある。

10.2.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

10.2.3 適用上の注釈

自己保護、ドメイン分離、及び非バイパス性の特性は、CCパート2のSFRで表現されているセキュリティ機能性とは区別される。これは、一般に自己保護や非バイパス性は、TSFに直接観察可能なインタフェースを持たないからである。これらはむしろTOE及びTSFの設計によって達成されるTSFの特性であり、その設計の正しい実装によって実施される。

このファミリでのアプローチとしては、まず開発者が、上述の特性を示すTSFを設計し提供することと、TSFのこれらの特性を説明する証拠を(証拠資料の形で)提供する。この説明は、TOE設計文書におけるTOEのSFR実施エレメントの記述と同じ詳細レベルで提供される。評価者には、その証拠を調べ、TOE及びTSFのために配付されるその他の証拠と組み合わせて、特性が達成されていることを決定する責任がある。

SFRを実装するセキュリティ機能性の仕様(機能仕様(ADV_FSP)及びTOE設計(ADV_TDS)内)が、自己保護及び非バイパス性を実装する際に採用されるメカニズム(メモリ管理メカニズムなど)を必ずしも記述するとは限らない。このため、これらの要件が達成されていることの保証を提供するために必要な資料には、ADV_FSP及びADV_TDSに組み込まれたTSFの設計コンポーネント構成からは切り離された表現が適している。これは、このコンポーネントで要求されるセキュリティアーキテクチャ記述が、設計コンポーネント構成資料を参照又は利用できないことを意味するものではないが、コンポーネント構成証拠資料内の詳細情報の大半は、セキュリティアーキテクチャ記述文書に提供されている論証とは無関係である。

アーキテクチャの健全性の記述は、TSFが健全であり、そのSFRを全て実施する理由についての正当性を提供するという点で、開発者の脆弱性分析と考えることができる。健全性が特定のセキュリティメカニズムを通して達成される場合、これらは深さ(ATE_DPT)要件の一部としてテストされる。健全性がアーキテクチャのみを通して達成される場合、そのふるまいはAVA: 脆弱性評定要件の一部としてテストされる。

このファミリは、自己保護、ドメイン分離、非バイパス性の各原則を記述するセキュリティアーキテクチャ記述に関する要件で構成される。これには、これらの原則が、TSFの初期化に使用されるTOEの部分によってどのようにサポートされるかについての記述が含まれる。

マルチ保証評価の場合、自己保護、ドメイン分離、及び非バイパス性という特性は、**サブTSF**間の境界についても記述されるかもしれない。

自己保護、ドメイン分離、及び非バイパス性のセキュリティアーキテクチャ特性に関する追加情報が、附属書A.1「ADV_ARC: セキュリティアーキテクチャに関する補足資料」に記載されている。

10.2.4 ADV_ARC.1 セキュリティアーキテクチャ記述

依存性： ADV_FSP.1 基本機能仕様

ADV_TDS.1 基本設計

開発者アクションエレメント:

ADV_ARC.1.1D

開発者は、TSFのセキュリティ特性がバイパスされないようにTOEを設計及び実装しなければならない。

ADV_ARC.1.2D

開発者は、TSFが信頼できない能動的なエンティティによって改ざんされるのを防ぐことができるようにTSFを設計及び実装しなければならない。

ADV_ARC.1.3D

開発者は、TSFのセキュリティアーキテクチャ記述を提供しなければならない。

内容・提示エレメント:

ADV_ARC.1.1C

セキュリティアーキテクチャ記述は、TOE設計文書に記述されているSFR実施抽象概念の記述に見合った詳細レベルでなければならない。

ADV_ARC.1.2C

セキュリティアーキテクチャ記述は、TSFによって維持されるセキュリティドメインを、SFRと一貫する形で記述しなければならない。

ADV_ARC.1.3C

セキュリティアーキテクチャ記述は、TSFの初期化プロセスのセキュリティがどのようにして確保されるのかを記述しなければならない。

ADV_ARC.1.4C

セキュリティアーキテクチャ記述は、TSFが改ざんから自分自身を保護することを実証しなければならない。

ADV_ARC.1.5C

セキュリティアーキテクチャ記述は、TSFがSFR実施機能性のバイパスを防ぐことを実証しなければならない。

評価者アクションエレメント:

ADV_ARC.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

10.3 機能仕様(ADV_FSP)

10.3.1 目的

このファミリは、TSFインタフェース(TSFI)を記述する機能仕様に対して要件を課す。TSFIは、外部エンティティ(又はTOEのサブジェクトでTSFの範囲外)が、TSFにデータを提供し、TSFからデータを受け取り、又はTSFからサービスを呼び出す、全ての手段で構成される。ただし、TSFがそれらのサービス要求を処理する方法や、TSFがその運用環境からサービスを呼び出す際の通信については記述*されない*。この情報は、TOE設計(ADV_TDS)ファミリ及び依存コンポーネントの依存(ACO_REL)ファミリでそれぞれ扱われる。

このファミリは、TSFが主張されたSFRをどのように満たしているかを評価者が把握できるようにすることで、直接的に保証を提供する。また、次の保証ファミリ及びクラスへの入力として、間接的にも保証を提供する。

— ADV_ARC、このファミリでは、TSFが不正行為(自己保護やドメイン分離の破壊)及び/又はバイパスからどのように保護されるかを的確に把握するためにTSFIの記述が使用される。

— ATE、このクラスでは、TSFIの記述が、開発者テストと評価者テストの両方の重要な入力として使用される。

— AVA、このクラスでは、TSFIの記述が脆弱性の探索に使用される。

10.3.2 コンポーネントのレベル付け

このファミリのコンポーネントは、TSFIの記述で要求される詳細の程度、及びTSFI記述で要求される形式化の程度に基づいて、レベル付けされている。

10.3.3 適用上の注釈

10.3.3.1 一般

TSFIが決定されると(TSFIの決定付けのガイダンス及び例については、A.2.2を参照のこと)、それらは記述される。下位レベルのコンポーネントでは、開発者は証拠資料を(評価者は分析を)、よりTOEのセキュリティに関連する側面に焦点を合わせる。TSFIの3つのカテゴリは、それらを通して利用できるサービスと、主張されているSFRとの関連性に基づいて定義される:

— インタフェースを通して実行されるサービスを、TSFに課せられたSFRの1つにまでたどることができる場合、そのインタフェースは*SFR実施*と呼ばれる。場合によっては、インタフェースに各種のサービスと結果があり、その中にSFR実施とそうでないものが含まれる可能性があるので注意する必要がある。

— SFR実施機能性が依存しているが、TOEのセキュリティ方針を保持するために正しく機能することだけが要求されるサービスへのインタフェース(又はそのサービスに関連するインタフェースを通じて利用可能なサービス)は、*SFR支援*と呼ばれる。

— SFR実施機能性が一切依存していないサービスへのインタフェースは、*SFR非干渉*と呼ばれる。

インタフェースをSFR支援又はSFR非干渉とする場合、そのインタフェースにはSFR実施のサービスや結果が含まれてはならないという点に注意するべきである。一方、SFR実施インタフェースは、SFR支援サービスを含むこともできる(例えば、システムの時刻を設定する操作はSFR実施サービスで、それと同じインタフェースを使用するシステムの日付を表示するサービスはSFR支援という場合もある)。純粋なSFR支援インタフェースの例としては、利用者及び利用者の代わりに実行されるTSFの一部の両方が使用するシステムコールインタフェースなどがある。

TSFIに関する情報がより多く提供されるほど、インタフェースが正しく分類/分析される保証も高くなる。評価者がこの判断を効果的に行えるように、要件は、最も低いレベルでSFR非干渉インタフェースに必要な情報が必要最小限となるように構造化される。レベルが上位になるほど、利用可能な情報が増え、評価者がより強い自信を持って指示を行うことができる。

3つのラベル(SFR実施、SFR支援、及びSFR非干渉)を定義し、それぞれに(より下位の保証コンポーネントで)異なる要件を課す目的は、分析及びその分析の実行対象である証拠の最初のおおよその焦点を絞ることである。開発者が提供したTSFインタフェースの証拠資料が、全てのインタフェースを、SFR実施インタフェースの要件で特定された程度まで記述している(つまり証拠資料が要件を上回っている)場合、開発者は要件と一致する新たな証拠を作成する必要はない。同様に、ラベルは要件内でインタフェースのタイプを区別する手段に過ぎないため、開発者はインタフェースをSFR実施、SFR支援、又はSFR非干渉と分類するためだけに証拠を更新する必要はない。このラベル付けの主な目的は、成熟した開発方法(及び詳細なインタフェース及び設計証拠資料などの関連する資料)を確立していない開発者が、過度なコストをかけずに必要な証拠のみを提供できるようにすることである。

このファミリ内の各コンポーネントの最後のCエレメントは、SFRと機能仕様間の直接的な対応を提供する。つまり、主張されている各SFRを呼び出すために使用されるインタフェースを示す。STに、TSFIではその機能性が発現しないCCパート2などの機能要件が含まれている場合は、機能仕様及び/又は追跡が、これらのSFRを識別することが期待される。機能仕

様にそれらのSFRを含めると、下位レベルの分解でそれらが失われず、関連性を持つことを保証するのに役立つ。

10.3.3.2 インタフェースに関する詳細

要件は提供されるTSFIに関する詳細の集合を定義する。要件上、インタフェースはその目的、使用方法、パラメタ、パラメタ記述、及び誤りメッセージの観点から(様々な詳細の程度で)特定される。

インタフェースの*目的*は、インタフェースの全般的な目標を上位レベルで記述することである(例えば、GUIコマンドの処理、ネットワークパケットの受信、プリンタ出力の提供など)。

インタフェースの*使用方法*は、インタフェースがどのように使用されることが期待されているかを記述する。この記述は、そのインタフェースで利用可能な各種の相互作用を中心に作成されるべきである。例えば、インタフェースがUNIXコマンドシェルである場合は、*ls*、*mv*、*cp*が、そのインタフェースの相互作用である。使用方法は、そのインタフェースでのふるまい(例えば、プログラマによるAPIの呼び出しやWindows利用者によるレジストリ設定の変更など)及び他のインタフェースでのふるまい(例えば、監査レコードの生成)の両方に対して相互作用が何を行うかを、相互作用ごとに記述する。

*パラメタ*は、インタフェースへの明示的な入力、及びインタフェースからの明示的な出力であり、そのインタフェースのふるまいを制御する。例えば、APIに渡される引数、特定のネットワークプロトコルのパケットの様々なフィールド、Windowsレジストリの個々のキーの値、チップの一連のピンでやり取りされる信号、*ls*に設定可能なフラグなどがパラメタである。パラメタはそれらの内容の単純なリストで「識別」される。

*パラメタの記述*は、そのパラメタが何であるかを意味のある形で伝える。例えば、インタフェース*foo(i)*に対して受け入れられるパラメタ記述は、「パラメタiは、現在システムにログインしている利用者の数を示す整数である」という記述になる。「パラメタiは整数である」などの記述は受け入れられない。

インタフェースの*アクション*の記述は、インタフェースが何を実行するのかを記述する。これは、「目的」がそれを使用する理由を表すのに対し、「アクション」はそれが実行する全てのものを表すという点で、目的よりも詳細になる。これらのアクションは、SFRと関連する場合と関連しない場合がある。インタフェースのアクションがSFRに関連しない場合、その記述は*概要*、つまり、記述はSFRがまったく関連しないことを単に明らかにするものとなる。

*誤りメッセージ記述*は、そのメッセージが生成された条件、メッセージの内容、及び誤りコードの意味を識別する。誤りメッセージは、問題やある程度の異常が検出されたことを示すためにTSFによって生成される。このファミリの要件は、次のようなさまざまな種類の誤りメッセージを表す:

— 「直接的」誤りメッセージは、特定のTSFI呼び出しを介するセキュリティ関連の応答である。

— 「間接的」誤りは、システム全体の条件(資源の消耗、接続の中断など)が原因で発生するため、特定のTSFI呼び出しに関連付けることはできない。セキュリティに関連しない誤りメッセージも「間接的」とみなされる。

— 「残り」の誤りは、コード内で参照可能な誤りなど、他の全ての誤りである。例えば、論理的には発生しない条件(「case」ステートメントのリストの後に最後の「else」が存在する場合など)をチェックする条件チェックコードの使用が、catch-all誤りメッセージ

を生成するために提供される。運用TOEにおいて、これらの誤りメッセージは表示されるべきではない。

機能仕様の例はA.2.4に示される。

10.3.3.3 このファミリのコンポーネント

このファミリの様々な階層コンポーネントで詳述されているように、インタフェース仕様での完全性と正確さが増すことで高まる保証は、開発者に要求される証拠資料に反映される。

ADV_FSP.1基本機能仕様では、唯一要求される証拠資料として、全てのTSFIの特性及びSFR実施TSFIとSFR支援TSFIの上位レベルの記述がある。TSFの「重要な」側面がTSFIで正しく特徴付けされていることの保証を提供するために、開発者には、SFR実施及びSFR支援TSFIの目的、使用方法、パラメタを提供することが要求される。

ADV_FSP.2セキュリティ実施機能仕様では、開発者に対し、全てのTSFIの目的、使用方法、パラメタ、及びパラメタ記述を提供することが要求される。さらに、SFR実施TSFIについて、開発者はSFR実施アクション及び直接的誤りメッセージを記述しなければならない。

ADV_FSP.3完全な要約を伴う機能仕様では、開発者は、ADV_FSP.2で要求される情報に加えて、SFR支援アクションとSFR非干渉アクションに関し、それらがSFR実施ではないことを示す十分な情報を提供しなければならない。さらに、開発者はSFR実施TSFIの呼び出しに起因する全ての直接的誤りメッセージを、証拠資料として提出しなければならない。

ADV_FSP.4完全な機能仕様では、全てのTSFI(SFR実施、SFR支援、又はSFR非干渉)を、直接的誤りメッセージを全て含め同じ程度に記述する必要がある。

ADV_FSP.5追加の誤り情報を伴う完全な準形式的機能仕様では、TSFI記述にはTSFIの呼び出しによって発生しない誤りメッセージも含まれる。

ADV_FSP.6追加の形式的仕様を伴う完全な準形式的機能仕様では、ADV_FSP.5に必要な情報に加え、全ての残りの誤りメッセージが含まれる。開発者は、TSFIの形式的な記述も提供しなければならない。これにより、TSFIを新たな観点から見て、不一致あるいは不完全な仕様を明らかにすることができる。

10.3.4 ADV_FSP.1 基本機能仕様

依存性：なし

開発者アクションエレメント:

ADV_FSP.1.1D

開発者は、機能仕様を提供しなければならない。

ADV_FSP.1.2D

開発者は、機能仕様からSFRへの追跡を提供しなければならない。

内容・提示エレメント:

ADV_FSP.1.1C

機能仕様は、SFR実施及びSFR支援の各TSFIの目的と使用方法を記述しなければならない。

ADV_FSP.1.2C

機能仕様は、SFR実施及びSFR支援の各TSFIに関連する全てのパラメタを識別しなければならない。

ADV_FSP.1.3C

機能仕様は、暗黙的に**SFR**非干渉として分類されているインタフェースについて、その分類の根拠を示さなければならない。

ADV_FSP.1.4C

追跡は、機能仕様での**TSFI**に対する**SFR**の追跡を実証するものでなければならない。

評価者アクションエレメント:

ADV_FSP.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_FSP.1.2E

評価者は、機能仕様が、**SFR**の正確かつ完全な具体化であることを*決定しなければならない*。

10.3.5 ADV_FSP.2 セキュリティ実施機能仕様

依存性：ADV_TDS.1 基本設計

開発者アクションエレメント:

ADV_FSP.2.1D

開発者は、機能仕様を提供しなければならない。

ADV_FSP.2.2D

開発者は、機能仕様からSFRへの追跡を提供しなければならない。

内容・提示エレメント:

ADV_FSP.2.1C

機能仕様は、完全に**TSF**を表現しなければならない。

ADV_FSP.2.2C

機能仕様は、全ての**TSFI**の目的と使用方法を記述しなければならない。

ADV_FSP.2.3C

機能仕様は、各TSFIに関連する全てのパラメタを識別及び記述しなければならない。

ADV_FSP.2.4C

各**SFR**実施**TSFI**について、機能仕様は、そのTSFIに関連する**SFR**実施アクションを記述しなければならない。

ADV_FSP.2.5C

各**SFR**実施**TSFI**について、機能仕様は、**SFR**実施アクションに関連する処理によって発生する誤りメッセージを記述しなければならない。

ADV_FSP.2.6C

追跡は、機能仕様でのTSFIに対するSFRの追跡を実証するものでなければならない。

評価者アクションエレメント:

ADV_FSP.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_FSP.2.2E

評価者は、機能仕様が、SFRの正確かつ完全な具体化であることを*決定しなければならない*。

10.3.6 ADV_FSP.3 完全な要約を伴う機能仕様

依存性：ADV_TDS.1 基本設計

開発者アクションエレメント:

ADV_FSP.3.1D

開発者は、機能仕様を提供しなければならない。

ADV_FSP.3.2D

開発者は、機能仕様からSFRへの追跡を提供しなければならない。

内容・提示エレメント:

ADV_FSP.3.1C

機能仕様は、完全にTSFを表現しなければならない。

ADV_FSP.3.2C

機能仕様は、全てのTSFIの目的と使用方法を記述しなければならない。

ADV_FSP.3.3C

機能仕様は、各TSFIに関連する全てのパラメタを識別及び記述しなければならない。

ADV_FSP.3.4C

各SFR実施TSFIについて、機能仕様は、そのTSFIに関連するSFR実施アクションを記述しなければならない。

ADV_FSP.3.5C

各SFR実施TSFIについて、機能仕様は、その**TSFIの呼び出し**に関連する**SFR実施アクション及び例外**によって発生する直接的誤りメッセージを記述しなければならない。

ADV_FSP.3.6C

機能仕様は、各TSFIに関連するSFR支援及びSFR非干渉アクションを要約しなければならない。

ADV_FSP.3.7C

追跡は、機能仕様でのTSFIに対するSFRの追跡を実証するものでなければならない。

評価者アクションエレメント:

ADV_FSP.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_FSP.3.2E

評価者は、機能仕様が、SFRの正確かつ完全な具体化であることを*決定しなければならない*。

10.3.7 ADV_FSP.4 完全な機能仕様

依存性：ADV_TDS.1 基本設計

開発者アクションエレメント:

ADV クラス: 開発

ADV_FSP.4.1D

開発者は、機能仕様を提供しなければならない。

ADV_FSP.4.2D

開発者は、機能仕様からSFRへの追跡を提供しなければならない。

内容・提示エレメント:

ADV_FSP.4.1C

機能仕様は、完全にTSFを表現しなければならない。

ADV_FSP.4.2C

機能仕様は、全てのTSFIの目的と使用方法を記述しなければならない。

ADV_FSP.4.3C

機能仕様は、各TSFIに関連する全てのパラメタを識別及び記述しなければならない。

ADV_FSP.4.4C

機能仕様は、各TSFIに関連する**全ての**アクションを**記述**しなければならない。

ADV_FSP.4.5C

機能仕様は、各TSFIの呼び出しによって発生する可能性のある全ての直接的誤りメッセージを記述しなければならない。

ADV_FSP.4.6C

追跡は、機能仕様でのTSFIに対するSFRの追跡を実証するものでなければならない。

評価者アクションエレメント:

ADV_FSP.4.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_FSP.4.2E

評価者は、機能仕様が、SFRの正確かつ完全な具体化であることを*決定しなければならない*。

10.3.8 ADV_FSP.5 追加の誤り情報を伴う完全な準形式的機能仕様

依存性： ADV_TDS.1 基本設計

ADV_IMP.1 TSFの実装表現

開発者アクションエレメント:

ADV_FSP.5.1D

開発者は、機能仕様を提供しなければならない。

ADV_FSP.5.2D

開発者は、機能仕様からSFRへの追跡を提供しなければならない。

内容・提示エレメント:

ADV_FSP.5.1C

機能仕様は、完全にTSFを表現しなければならない。

ADV_FSP.5.2C

機能仕様は、準形式的スタイルを使用してTSFIを記述しなければならない。

ADV_FSP.5.3C

機能仕様は、全てのTSFIの目的と使用方法を記述しなければならない。

ADV_FSP.5.4C

機能仕様は、各TSFIに関連する全てのパラメタを識別及び記述しなければならない。

ADV_FSP.5.5C

機能仕様は、各TSFIに関連する全てのアクションを記述しなければならない。

ADV_FSP.5.6C

機能仕様は、各TSFIの呼び出しによって発生する可能性のある全ての直接的誤りメッセージを記述しなければならない。

ADV_FSP.5.7C

機能仕様は、TSFIの呼び出しによって発生しない全ての誤りメッセージを記述しなければならない。

ADV_FSP.5.8C

機能仕様は、TSFの実装に含まれているがTSFIの呼び出しによって発生しない各誤りメッセージについて、その根拠を示さなければならない。

ADV_FSP.5.9C

追跡は、機能仕様でのTSFIに対するSFRの追跡を実証するものでなければならない。

評価者アクションエレメント:

ADV_FSP.5.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_FSP.5.2E

評価者は、機能仕様が、SFRの正確かつ完全な具体化であることを*決定しなければならない*。

10.3.9 ADV_FSP.6 追加の形式的仕様を伴う完全な準形式的機能仕様

依存性： ADV_TDS.1 基本設計

ADV_IMP.1 TSFの実装表現

開発者アクションエレメント:

ADV_FSP.6.1D

開発者は、機能仕様を提供しなければならない。

ADV_FSP.6.2D

開発者は、TSFの機能仕様の形式的表現を提供しなければならない。

ADV_FSP.6.3D

開発者は、機能仕様からSFRへの追跡を提供しなければならない。

内容・提示エレメント:

ADV_FSP.6.1C

機能仕様は、完全にTSFを表現しなければならない。

ADV_FSP.6.2C

機能仕様は、**形式的**スタイルを使用してTSFIを記述しなければならない。

ADV_FSP.6.3C

機能仕様は、全てのTSFIの目的と使用方法を記述しなければならない。

ADV_FSP.6.4C

機能仕様は、各TSFIに関連する全てのパラメタを識別及び記述しなければならない。

ADV_FSP.6.5C

機能仕様は、各TSFIに関連する全てのアクションを記述しなければならない。

ADV_FSP.6.6C

機能仕様は、各TSFIの呼び出しによって発生する可能性のある全ての直接的誤りメッセージを記述しなければならない。

ADV_FSP.6.7C

機能仕様は、**TSF実装表現に含まれている**全ての誤りメッセージを記述しなければならない。

ADV_FSP.6.8C

機能仕様は、TSF実装に含まれている**が機能仕様には記述されない**、各誤りメッセージについて、TSFIに**関連しない理由を正当化する根拠**を提供しなければならない。

ADV_FSP.6.9C

TSFの機能仕様の形式的表現は、適切な個所に対して非形式的で説明的なテキストで補足される形式的スタイルを使用して、TSFIを記述しなければならない。

ADV_FSP.6.10C

追跡は、機能仕様でのTSFIに対するSFRの追跡を実証するものでなければならない。

評価者アクションエレメント:

ADV_FSP.6.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_FSP.6.2E

評価者は、機能仕様が、SFRの正確かつ完全な具体化であることを*決定しなければならない*。

10.4 実装表現(ADV_IMP)

10.4.1 目的

実装表現(ADV_IMP)ファミリの機能は、TOEの実装表現(及び上位レベルでは実装自体)を、評価者が分析できる形式で開発者が提供できるようにする。実装表現は、他のファミリの分析アクティビティ(TOE設計の分析など)で、TOEがその設計に適合していることを実証し、その他の評価領域(例えば、脆弱性の探索)での分析の基礎を提供するために使用される。実装表現は、TSFの詳細な内部動作を示す形式であることが期待される。これには、ソフトウェアのソースコード、ファームウェアのソースコード、ハードウェア図、及び/又はICハードウェア設計言語コード又はレイアウトデータが含まれる。

10.4.2 コンポーネントのレベル付け

このファミリのコンポーネントは、TOE設計記述にマッピングされる実装の量に基づいてレベル付けされている。

10.4.3 適用上の注釈

ソースコードや、実際のハードウェアの製造に用いられるハードウェア図及び/又はICハードウェア設計言語コードやレイアウトデータは、実装表現の一部の例である。実装表現は、評価者に提供しなければならないが、これは評価者がその表現を所有する必要があることを意味するものではないということが重要である。例えば、開発者の選んだサイトで評価者が実装表現をレビューすることを、開発者が要求する場合がある。

情報不足によって分析アクティビティが制限されることのないように、実装表現全体が提供される。とはいえ、分析アクティビティが行われる際に全ての表現が検査されるわけではない。そのようなことは、ほとんど全ての場合に現実的でないうえ、たいていは、実装表現のターゲットサンプリングに比べてTOEの保証が高くなるわけでもない。実装表現は、他のTOE設計コンポーネント構成(機能仕様やTOE設計など)の分析を可能にするため、及び設計の上位レベルで記述されているセキュリティ機能性が実際にTOEに実装されるという確信を得るために提供される。ある種の実装表現には、コンパイルや実行時の解釈の実際の結果を、実装表現のみから決定するのを困難にしたり不可能にしたりするような規則がある。例えばC言語コンパイラでは、コンパイラディレクティブによって、コードの特定の部分全体が除外されたり含まれたりする。このため、このような「付加的な」情報又は関連ツール(スクリプト、コンパイラなど)を提供して、実装表現が正確に決定されるようにすることが重要である。

実装表現とTOE設計記述間のマッピングの目的は、評価者の分析を支援することである。TOEの内部動作は、TOE設計が実装表現の対応する部分で分析されたときに理解が深まる可能性がある。マッピングは、実装表現への索引として使用される。下位のコンポーネントでは、実装表現のサブセットだけがTOE設計記述にマッピングされる。実装表現の中でそのようなマッピングを必要とする部分が不確定であるため、開発者は実装表現全体を事前にマップするか、実装表現の中で評価者がマッピングを必要とする部分が確認できるまで待つかを選択できる。

実装表現は、開発者によって、実際の実装への変換に適した形式で操作される。例えば開発者は、最終的にコンパイルされてTSFの一部となるソースコードを含むファイルを使用することができる。開発者は、評価者が分析において自動化の技法を使用できるように、開発者が使用する形式で実装表現を提供する。これにより、検査される実装表現が、実際にTSFの作成に使用されるものであるという信頼も高まる(ワードプロセッサ文書などの別の表現形式で提供される場合とは対照的)。ただし、開発者は他の形式の実装表現も使用できるという点に注意するべきである。それらの形式の実装表現も一緒に提供される。全体的な目標は、評価者の分析の効果を最大限に高める情報を提供することである。

ある種の実装表現では、理解や分析に対する重大な障害が持ち込まれるために、追加の情報が必要になることがある。例えば、「隠蔽されている」ソースコードや、理解や分析を妨げるその他の形で分かりにくくされているコードがこれに該当する。一般に、このような形式の実装表現は、TOE開発者によって使用されているバージョンの実装表現に対して、コードを隠蔽したり分かりにくくしたりするプログラムが実行された結果である。隠蔽されている表現はコンパイルの対象であり、元の隠蔽されていない表現より(構造の観点からは)実装に近いと言えるが、そのように分かりにくくされているコードを提供すると、その表現に関連する分析作業にかかる時間が大幅に増加する可能性がある。このような形式の表現が作成される場合は、隠蔽されていない表現を提供できるように、使用されている隠蔽ツール/アル

ゴリズムについての詳細がコンポーネントで必要とされる。この追加の情報は、隠蔽のプロセスによって弱体化しているセキュリティ機能性がないという確信を得るために使用できる。

10.4.4 ADV_IMP.1 TSFの実装表現

依存性： ADV_TDS.3 基本モジュール設計

ALC_TAT.1 明確に定義された開発ツール

開発者アクションエレメント:

ADV_IMP.1.1D

開発者は、TSF全体の実装表現を提供しなければならない。

ADV_IMP.1.2D

開発者は、TOE設計記述と実装表現のサンプルの間のマッピングを提供しなければならない。

内容・提示エレメント:

ADV_IMP.1.1C

実装表現は、それ以上の設計上の決定を必要とせずに、TSFを生成できるような詳細レベルまでTSFを定義しなければならない。

ADV_IMP.1.2C

実装表現の形式は、開発要員が使用する形式でなければならない。

ADV_IMP.1.3C

TOE設計記述と実装表現のサンプルの間のマッピングは、両者の対応を実証しなければならない。

評価者アクションエレメント:

ADV_IMP.1.1E

評価者は、選択された実装表現のサンプルについて、提供された情報が証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

10.4.5 ADV_IMP.2 TSFの実装表現の完全なマッピング

依存性： ADV_TDS.3 基本モジュール設計

ALC_TAT.1 明確に定義された開発ツール

ALC_CMC.5 高度なサポート

開発者アクションエレメント:

ADV_IMP.2.1D

開発者は、TSF全体の実装表現を提供しなければならない。

ADV_IMP.2.2D

開発者は、TOE設計記述と実装表現**全体**の間のマッピングを提供しなければならない。

内容・提示エレメント:

ADV_IMP.2.1C

実装表現は、それ以上の設計上の決定を必要とせずに、TSFを生成できるような詳細レベルまでTSFを定義しなければならない。

ADV_IMP.2.2C

実装表現の形式は、開発要員が使用する形式でなければならない。

ADV_IMP.2.3C

TOE設計記述と実装表現**全体**の間のマッピングは、両者の対応を実証しなければならない。

評価者アクションエレメント:

ADV_IMP.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

10.5 TSF内部構造(ADV_INT)

10.5.1 目的

このファミリは、TSFの内部構造の評定を扱う。内部構造が適切に構成されたTSFは、実装が容易になり、脆弱性の原因となる欠陥を含む可能性が低くなる。また、欠陥をもたらさない保守も容易になる。

10.5.2 コンポーネントのレベル付け

このファミリのコンポーネントは、必要となる構造と複雑さの最小化の量に基づいて、レベル付けされている。ADV_INT.1 TSF内部構造の適切に構成されたサブセットが要件を課す対象は、TSFの選択された部分のみでの適切に構成された内部構造である。このコンポーネントはEALに含まれない。これは、このコンポーネントが特別な状況(例えば、スポンサーが、TSFの他の部分からは孤立している暗号モジュールに特別な関心を持っている場合)で使用されると見られ、広く適用されることがないからである。

次のレベルでは、適切に構成された内部構造に対する要件がTSF全体に課せられる。最後に、複雑さの最小化が最上位のコンポーネントに導入される。

10.5.3 適用上の注釈

一般的にこれらの要件は、TSFの内部構造に適用されることで、開発者と評価者の両方がTSFを理解しやすくなるという改良をもたらし、またテストスイートを設計及び評価するための基礎を提供する。さらに、TSFの理解のしやすさが向上することは、開発者によるTSFの保守性の単純化に役立つ。

このファミリの要件は、かなり抽象的なレベルで表される。TOEの多様性から、「適切に構成された」又は「最小の複雑さ」よりも具体的なものを体系化することは不可能である。構造と複雑さに関する判断は、TOEで使用される特定の技術から導き出されることが期待される。例えば、ソフトウェアエンジニアリングの分野で挙げられる特性を示す場合、ソフトウェアは適切に構成されたものとみなされる可能性が高い。このファミリ内のコンポーネントは、適切に構成され複雑すぎない特性を測定するための標準を識別することを要求する。

10.5.4 ADV_INT.1 適切に構成されたTSF内部構造のサブセット

依存性： ADV_IMP.1 TSFの実装表現

ADV_TDS.3 基本モジュール設計

ALC_TAT.1 明確に定義された開発ツール

目的

このコンポーネントの目的は、TSFの特定の部分を適切に構成することを要求する手段を提供することである。その意図は、TSF全体は適切なエンジニアリングの原則を使用して設計及び実装されるが、分析は特定のサブセットにのみ基づいて実行されるということである。

適用上の注釈

このコンポーネントは、PP又はSTの作成者に対し、TSFのサブセットを割付に記入することを要求する。このサブセットは、抽象化のいずれかの層でTSFの内部構造という観点から識別される場合がある。たとえば:

a) TOE設計で識別されるTSFの構造エレメント(例: 「開発者は、適切に構成された内部構造を持つように*監査サブシステム*を設計及び実行しなければならない」)。

b) 実装(例: 「開発者は*encrypt.c ファイルと decrypt.c ファイル*を、適切に構成された内部構造を持つように設計及び実装しなければならない」又は「開発者は、適切に構成された内部構造を持つように*6227 IC チップ*を設計及び実装しなければならない」)。

分析の焦点となる場所を示さないため、主張されているSFRを参照しても確実に実現できない可能性がある(例: 「開発者は、適切に構成された内部構造を持つように、*FPR_ANO.2 で定義されているとおりに匿名性を提供するTSFの部分*を設計及び実装しなければならない」)。

このコンポーネントは値が制限されており、悪意を持っている可能性がある利用者/サブジェクトによるTSFIへのアクセスが制限又は厳しく制御されている場合や、TSFの選択されたサブセットがTSFの他の部分から悪影響を受けない(例えば暗号化機能性は適切に構成され、TSFの他の部分から分離される)ことを保証する別の保護手段(例えばドメイン分離)が存在する場合に適している。

開発者アクションエレメント:

ADV_INT.1.1D

開発者は、適切に構成された内部構造を持つように[割付: *TSFのサブセット*]を設計及び実装しなければならない。

ADV_INT.1.2D

開発者は、内部構造の記述及び正当化を提供しなければならない。

内容・提示エレメント:

ADV_INT.1.1C

正当化は、「適切に構成された」の意味を判断するために使用される特性を説明しなければならない。

ADV_INT.1.2C

TSF内部構造の記述は、割り付けられたTSFのサブセットが適切に構成されていることを実証しなければならない。

評価者アクションエレメント:

ADV_INT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_INT.1.2E

評価者は、割り付けられたTSFのサブセットに関する内部構造の分析を*実行しなければならない*。

10.5.5 ADV_INT.2 適切に構成された内部構造

依存性： ADV_IMP.1 TSFの実装表現

ADV_TDS.3 基本モジュール設計

ALC_TAT.1 明確に定義された開発ツール

目的

このコンポーネントの目的は、TSFを適切に構成することを要求する手段を提供することである。その意図は、TSF全体が適切なエンジニアリングの原則を使用して設計及び実装されていることである。

適用上の注釈

構造の適切性に関する判断は、TOEで使用される特定の技術から導き出されることが期待される。このコンポーネントは、適切に構成されているという特性を測定するための標準を識別することを要求する。

開発者アクションエレメント:

ADV_INT.2.1D

開発者は、適切に構成された内部構造を持つように**TSF全体**を設計及び実装しなければならない。

ADV_INT.2.2D

開発者は、内部構造の記述及び正当化を提供しなければならない。

内容・提示エレメント:

ADV_INT.2.1C

正当化は、「適切に構成された」の意味を判断するために使用される特性を**記述**しなければならない。

ADV_INT.2.2C

TSF内部構造の記述は、TSF**全体**が適切に構成されていることを実証しなければならない。

評価者アクションエレメント:

ADV_INT.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_INT.2.2E

評価者は、TSFに関する内部構造の分析を*実行しなければならない*。

10.5.6 ADV_INT.3 最小限複雑な内部構造

依存性： ADV_IMP.1 TSFの実装表現

ADV_TDS.3 基本モジュール設計

ALC_TAT.1 明確に定義された開発ツール

目的

このコンポーネントの目的は、TSFが適切に構成され最小の複雑さであることを要求する手段を提供することである。その意図は、TSF全体が適切なエンジニアリングの原則を使用して設計及び実装されていることである。

適用上の注釈

構造及び複雑さの適切性に関する判断は、TOEで使用される特定の技術から導き出されることが期待される。このコンポーネントは、構造及び複雑さを測定するための標準を識別することを要求する。

開発者アクションエレメント:

ADV_INT.3.1D

開発者は、適切に構成された内部構造を持つようにTSF全体を設計及び実装しなければならない。

ADV_INT.3.2D

開発者は、内部構造の記述及び正当化を提供しなければならない。

内容・提示エレメント:

ADV_INT.3.1C

正当化は、「適切に構成された」及び「複雑」の意味を判断するために使用される特性を記述しなければならない。

ADV_INT.3.2C

TSF内部構造の記述は、TSF全体が適切に構成され、**複雑すぎない**ことを実証しなければならない。

評価者アクションエレメント:

ADV_INT.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_INT.3.2E

評価者は、TSF**全体**に関する内部構造の分析を*実行しなければならない*。

10.6 セキュリティ方針モデル化(ADV_SPM)

10.6.1 目的

このファミリの目的は、SFRとSTのセキュリティ対策方針によって定義されるTSFとその特性の形式的表現(それぞれ形式的モデルと形式的特性と呼ぶ)の開発を通じて、追加の保証を提供することである。これらの形式的特性が形式的モデルにおいて成立することを形式的証明によって確立し、TOE機能仕様が形式的モデルに対して証明された形式的特性を保持することを対応の根拠によって確立することが期待される。形式的仕様又は準形式的仕様が存在する場合(それぞれADV_FSP.6又はADV_FSP.5)、それらの仕様における形式的特性保持の形式的証明又は準形式的実証が期待される。

10.6.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

10.6.3 適用上の注釈

TOEの不十分性は、セキュリティ要件の理解不足、又はそれらのセキュリティ要件の実装に欠陥があることに起因している可能性がある。セキュリティ要件を適切に定義してそれらが確実に理解されるようにすることは難しい。これは、TOEの実装中に、望まない結果、つまりわずかでも欠陥が生じないようにするには、定義が十分に正確でなければならないからである。設計、実装、レビューの各プロセス全体で、TSFの形式的表現及び特性が正確な設計及び実装のガイダンスとして使用されることで、STのSFR及びセキュリティ対策方針をTOEが満たしていることの保証が高まる。形式的モデルを定義し、形式言語を用いて形式的特性を特定し、形式的モデルにおいてこれらの形式的特性が成立することを形式的に証明することによって、結果としてのガイダンスと、STのSFR及びセキュリティ対策方針によって定義されるTSF表現とその特性の精度は著しく改善する。

TSFの形式的セキュリティポリシーモデル(SPM)の作成は、STに関して完全でなければならない。このようなモデルは、曖昧な、一貫しない、矛盾した、又は実施できないエレメントを識別及び排除し、スコープに関するあらゆる誤解を回避するのに役立つ。このため、評価者は、形式的モデル及び形式的特性がSTを完全にカバーするかを決定し、スコープが一致するST及びSPMのみを受け入れなければならない。TOEが構築されると、形式的モデルは、実装されているTSFを開発者がどの程度十分に理解しているか、及びSTのセキュリティ対策方針で定義された形式的特性とTOE設計の間に一貫しない点がないかどうかについての評価者の判断に寄与することにより、評価作業に役立つ。モデルの特性を形式的に証明することで得られる確信は、形式的モデルとTOE機能仕様(ADV_FSPで定義されている)との間の対応の根拠を定義することで得られる確信を伴う。対応の根拠は、TOE機能仕様の形式的側面にマッピングする場合は形式的証明、それ以外の場合は準形式的実証で構成される。STの異なる部分(SFRとセキュリティ対策方針)及び対応の根拠のために、異なる形式的システム(モデリング言語、ツール、証明システム)の組合せを使用することができる。

10.6.4 ADV_SPM.1 形式的TOEセキュリティ方針モデル

依存性： ASE_OBJ.2 セキュリティ対策方針

ASE_REQ.2 導出されたセキュリティ要件

ADV_FSP.4 完全な機能仕様

開発者アクションエレメント:

ADV_SPM.1.1D

開発者は、説明文によってサポートされる、TSFの形式的モデルを提供しなければならない。

ADV_SPM.1.2D

開発者は、説明文によってサポートされる、TOEの形式的特性のセットを提供しなければならない。

ADV_SPM.1.3D

開発者は、説明文によってサポートされる、モデルが形式的特性を満たすことの形式的証明を提供しなければならない。

ADV_SPM.1.4D

開発者は、形式的モデルと機能仕様との間の対応の根拠を提供しなければならない。

ADV_SPM.1.5D

開発者は、形式的モデルと任意の準形式的な機能仕様との間の対応の準形式的な実証を提供しなければならない。

ADV_SPM.1.6D

開発者は、形式的モデルと任意の形式的な機能仕様との間の対応の形式的な証明を提供しなければならない。

内容・提示エレメント:

ADV_SPM.1.1C

形式的モデル、特性及び証明は、十分に根拠のある数学的理論を使用して定義されなければならない。

ADV_SPM.1.2C

説明文は、形式的モデル、形式的特性及び証明の全体をカバーし、証明を再現するための指示及び対応の根拠を含み、モデル化及び検証の選択に対する根拠を提供しなければならない。

ADV_SPM.1.3C

形式的モデルは、**TSF**を定義する**SFR**の完全なセットをカバーしなければならない。

ADV_SPM.1.4C

形式的特性は、**TOE**のセキュリティ対策方針の完全なセットをカバーしなければならない。

ADV_SPM.1.5C

形式的証明は、形式的モデルが全ての形式的特性を満足すること、及び基礎となる数学的理論の一貫性が保たれることを示さなければならない。

ADV_SPM.1.6C

対応の根拠は、形式的モデルについて証明された形式的特性が、機能仕様についても成立することを示さなければならない。

ADV_SPM.1.7C

対応の準形式的実証は、形式的モデルについて証明された形式的特性が、任意の準形式的機能仕様について成立することを示さなければならない。

ADV_SPM.1.8C

対応の形式的証明は、形式的モデルについて証明された特性が、任意の形式的機能仕様について成立することを示さなければならない。

ADV_SPM.1.9C

形式的特性又は形式的モデルと機能仕様の関係をモデル化又は証明するために使用されるツールは、明確に定義され、曖昧さなく識別されなければならず、ツールの適合性及び信頼性の証拠資料及び根拠を伴わなければならない。

評価者アクションエレメント:

ADV_SPM.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

10.7 TOE設計(ADV_TDS)

10.7.1 目的

TOEの設計記述は、TSFの記述の枠組み、及びTSFの綿密な記述の両方を提供する。より高い保証が必要になると、記述で提供される詳細のレベルも高くなる。TSFの大きさと複雑さが増大するにつれて、複数レベルの分解が適切となる。設計要件の意図するところは、指定の保証レベルに見合った情報を提供して、SFRが実現されていることを判断できるようにすることである。

10.7.2 コンポーネントのレベル付け

このファミリのコンポーネントは、TSFに関して提示する必要がある情報の量、及び設計記述に要求される形式化の程度に基づいて、レベル付けされている。

10.7.3 適用上の注釈

10.7.3.1 一般

設計証拠資料の目標は、TSFの境界を決定するための十分な情報を提供し、TSFがSFRを*どのように*実装するかを記述することである。設計証拠資料の量と構造は、TOEの複雑さとSFRの数によって決まる。一般に、多数のSFRを実装する非常に複雑なTOEは、わずかな数のSFRのみを実装する非常に単純なTOEに比べて、より多くの設計証拠資料を必要とする。非常に複雑なTOEでは、設計を記述する際に様々なレベルの分解を生成することが(提供される保証の点で)有用であるが、非常に単純なTOEは、その実装の上位レベルの記述と下位レベルの記述の両方が必要というわけではない。

このファミリは、*サブシステム*及び*モジュール*の2つのレベルの分解を使用する。モジュールは、最も具体的な機能性の記述、つまり実装の記述である。開発者は、モジュールで記述されたTOEの部分を、それ以上の設計上の決定を行うことなく実装できるべきである。サブシステムはTOEの設計の記述である。つまり、TOEの部分が何をどのように行うかを上位レベルで記述するのに役立つ。したがって、サブシステムは下位レベルのサブシステム又はモジュールに分割することができる。非常に複雑なTOEで、TOEの動作内容に関する有用な記述を十分に伝えるためには、複数レベルのサブシステムが必要となる。これとは対照的に、非常に単純なTOEはサブシステムレベルの記述を必要としない。つまり、TOEの動作内容はモジュールで明確に記述される。

設計証拠資料に採用される一般的な手法では、保証レベルの上昇に伴って、記述の重点が概略(サブシステムレベル)から詳細(モジュールレベル)にシフトする。モジュールレベルで十分に記述できるほどTOEが単純であるために、モジュールレベルの抽象化が適切である場合は、保証レベルでサブシステムレベルの記述が要求されても、モジュールレベルの記述で間に合う。しかしながら、複雑なTOEの場合はこれが当てはまらない。膨大な量の(モジュールレベルの)詳細は、サブシステムレベルの記述が付随していないと理解できない。

この手法は、TSFの実装に関する追加の詳細を提供することで、SFRが正しく実装されることの保証が高まり、さらにテスト(ATE: テスト)でこれを実証するための情報が提供されるという、一般的なパラダイムに従っている。

このファミリの要件では、*インタフェース*という用語が(2つのサブシステム又はモジュール間の)通信手段として使用される。インタフェースは、通信が起動される方法を記述する点でTSFIの詳細に似ている(機能仕様(ADV_FSP)を参照のこと)。*相互作用*という用語は、通信の目的を識別するために使用される。つまりこの用語は、2つのサブシステム又はモジュールが通信する理由を識別する。

10.7.3.2 サブシステム及びモジュールに関する詳細

要件は提供されるサブシステム及びモジュールに関する詳細の集合を定義する:

a) サブシステムとモジュールは、それらの内容を示す単純なリストで*識別*される。

b) サブシステムとモジュールは、「SFR実施」、「SFR支援」、又は「SFR非干渉」として(暗黙的又は明示的に)*分類*できる。これらの用語は機能仕様(ADV_FSP)で使用されているものと同様に使用される。

c) サブシステムの*ふるまい*は、そのサブシステムが実施する内容である。ふるまいも、SFR実施、SFR支援、又はSFR非干渉として分類できる。サブシステムのふるまいは、サブシステムそのものの分類よりもよりSFRに関連するものとして分類されない。例えば、SFR実施サブシステムは、SFR支援もしくはSFR非干渉のふるまいと同様に、SFR実施のふるまいを持つことができる。

d) サブシステムの*ふるまいの要約*は、そのサブシステムが実行するアクションの概要である(例: 「TCPサブシステムはIPデータグラムを信頼できるバイトストリームに集合させる」)。

e) サブシステムの*ふるまいの記述*は、サブシステムが行う全てのアクションの説明である。この記述は、ふるまいがSFRの実施と何らかの関連性を持つかどうかを確実に決定できる1つの詳細レベルにあるべきである。

f) サブシステムもしくはモジュール間での*相互作用の記述*は、サブシステムもしくはモジュールが通信する理由、及び渡される情報の特性を識別する。インタフェース仕様と同じレベルの詳細まで情報を定義する必要はない。例えば、「サブシステムXはメモリマネージャにメモリのブロックを要求し、メモリマネージャは割り当てられたメモリの場所で応答する」というような記述で十分である。

g) *インタフェースの記述*は、どのように、モジュール間での相互作用が達成されるかの詳細を提供する。インタフェースの記述は、モジュールが通信する理由や、通信の目的を記載(即ち、相互作用の記載)するというよりは、メッセージの構造と内容、セマフォ、内部プロセス通信の観点から、どのように、その通信が成し遂げられるかの詳細を記述する。

h) *目的*は、どのようにモジュールがそれらの機能を提供するかを記載する。目的には、それ以上の設計上の決定が必要ない十分な詳細を提供する。モジュールを実装する実装表現とモジュールの目的との間の対応は確実に明らかにすべきである。

i) そうしない場合、モジュールはエレメントで識別されるもの全ての観点から*記述される*。

サブシステムとモジュール、及び「SFR実施」については、附属書AのA.4「ADV_TDS: サブシステム及びモジュール」で詳しく説明されている。

10.7.4 ADV_TDS.1 基本設計

依存性：ADV_FSP.2 セキュリティ実施機能仕様

開発者アクションエレメント:

ADV_TDS.1.1D

開発者は、TOEの設計を提供しなければならない。

ADV_TDS.1.2D

開発者は、機能仕様のTSFIからTOE設計で利用可能な分解の最下位レベルへのマッピングを提供しなければならない。

内容・提示エレメント:

ADV_TDS.1.1C

設計は、サブシステムの観点からTOEの構造を記述しなければならない。

ADV_TDS.1.2C

設計は、TSFの全てのサブシステムを識別しなければならない。

ADV_TDS.1.3C

設計は、TSFの各SFR支援又はSFR非干渉サブシステムのふるまいの要約を提供しなければならない。

ADV_TDS.1.4C

設計は、SFR実施サブシステムのSFR実施のふるまいを要約しなければならない。

ADV_TDS.1.5C

設計は、TSFのSFR実施サブシステム間、及びTSFのSFR実施サブシステムとTSFのその他のサブシステム間の相互作用の記述を提供しなければならない。

ADV_TDS.1.6C

マッピングは、全てのTSFIが、それらが呼び出すTOE設計で記述されているふるまいを追跡することを実証しなければならない。

評価者アクションエレメント:

ADV_TDS.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_TDS.1.2E

評価者は、設計が、全てのセキュリティ機能要件の正確かつ完全な具体化であることを*決定しなければならない*。

10.7.5 ADV_TDS.2 アーキテクチャ設計

依存性：ADV_FSP.3 完全な要約を伴う機能仕様

開発者アクションエレメント:

ADV_TDS.2.1D

開発者は、TOEの設計を提供しなければならない。

ADV_TDS.2.2D

開発者は、機能仕様のTSFIからTOE設計で利用可能な分解の最下位レベルへのマッピングを提供しなければならない。

内容・提示エレメント:

ADV_TDS.2.1C

設計は、サブシステムの観点からTOEの構造を記述しなければならない。

ADV_TDS.2.2C

設計は、TSFの全てのサブシステムを識別しなければならない。

ADV_TDS.2.3C

設計は、TSFの各SFR非干渉サブシステムのふるまいの要約を提供しなければならない。

ADV_TDS.2.4C

設計は、SFR実施サブシステムのSFR実施のふるまいを**記述**しなければならない。

ADV_TDS.2.5C

設計は、SFR実施サブシステムの**SFR支援及びSFR非干渉**のふるまいを要約しなければならない。

ADV_TDS.2.6C

設計は、**SFR支援**サブシステムのふるまいを要約しなければならない。

ADV_TDS.2.7C

設計は、TSFの全てのサブシステム間の相互作用の記述を提供しなければならない。

ADV_TDS.2.8C

マッピングは、全てのTSFIが、それらが呼び出すTOE設計で記述されているふるまいを追跡することを実証しなければならない。

評価者アクションエレメント:

ADV_TDS.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_TDS.2.2E

評価者は、設計が、全てのセキュリティ機能要件の正確かつ完全な具体化であることを*決定しなければならない*。

10.7.6 ADV_TDS.3 基本モジュール設計

依存性：ADV_FSP.4完全な機能仕様

開発者アクションエレメント:

ADV_TDS.3.1D

開発者は、TOEの設計を提供しなければならない。

ADV_TDS.3.2D

開発者は、機能仕様のTSFIからTOE設計で利用可能な分解の最下位レベルへのマッピングを提供しなければならない。

内容・提示エレメント:

ADV_TDS.3.1C

設計は、サブシステムの観点からTOEの構造を記述しなければならない。

ADV_TDS.3.2C

設計は、モジュールの観点からTSFを記述しなければならない。

ADV_TDS.3.3C

設計は、TSFの全てのサブシステムを識別しなければならない。

ADV_TDS.3.4C

設計は、**TSF**の**各サブシステム**の**記述を提供**しなければならない。

ADV_TDS.3.5C

設計は、TSFの全てのサブシステム間の相互作用の記述を提供しなければならない。

ADV_TDS.3.6C

設計は、TSFのサブシステムからTSFのモジュールへのマッピングを提供しなければならない。

ADV_TDS.3.7C

設計は、**目的と他のモジュールとの関係の観点から**各**SFR実施モジュール**を記述しなければならない。

ADV_TDS.3.8C

設計は、各SFR実施モジュールのSFR関連インタフェース、それらのインタフェースからの戻り値、及びその他のモジュールとの相互作用及び他のSFR実施モジュールに対して呼び出されるSFR関連インタフェースの観点から各SFR実施モジュールを記述しなければならない。

ADV_TDS.3.9C

設計は、**目的及びその他のモジュールとの相互作用の観点から**各**SFR支援モジュール及びSFR非干渉モジュール**を記述しなければならない。

ADV_TDS.3.10C

マッピングは、全てのTSFIが、それらが呼び出すTOE設計で記述されているふるまいを追跡することを実証しなければならない。

評価者アクションエレメント:

ADV_TDS.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_TDS.3.2E

評価者は、設計が、全てのセキュリティ機能要件の正確かつ完全な具体化であることを*決定しなければならない*。

10.7.7 ADV_TDS.4 準形式的モジュール設計

依存性：ADV_FSP.5 追加の誤り情報を伴う完全な準形式的機能仕様

開発者アクションエレメント:

ADV_TDS.4.1D

開発者は、TOEの設計を提供しなければならない。

ADV_TDS.4.2D

開発者は、機能仕様のTSFIからTOE設計で利用可能な分解の最下位レベルへのマッピングを提供しなければならない。

内容・提示エレメント:

ADV_TDS.4.1C

設計は、サブシステムの観点からTOEの構造を記述しなければならない。

ADV_TDS.4.2C

設計は、**各モジュールをSFR実施、SFR支援、又はSFR非干渉として指示し**、モジュールの観点からTSFを記述しなければならない。

ADV_TDS.4.3C

設計は、TSFの全てのサブシステムを識別しなければならない。

ADV_TDS.4.4C

設計は、**適切な箇所に対して非形式的で説明的なテキストで補足される**、TSFの各サブシステムの**準形式的**記述を提供しなければならない。

ADV_TDS.4.5C

設計は、TSFの全てのサブシステム間の相互作用の記述を提供しなければならない。

ADV_TDS.4.6C

設計は、TSFのサブシステムからTSFのモジュールへのマッピングを提供しなければならない。

ADV_TDS.4.7C

設計は、目的とその他のモジュールとの関係の観点から各SFR実施**及びSFR支援**モジュールを記述しなければならない。

ADV_TDS.4.8C

設計は、各SFR実施モジュール及びSFR支援モジュールのSFR関連インタフェース、それらのインタフェースからの戻り値、その他のモジュールとの相互作用、及びその他のSFR実施**又はSFR支援**モジュールに対して呼び出されるSFR関連インタフェースの観点から各SFR実施モジュール**及びSFR支援**モジュールを記述しなければならない。

ADV_TDS.4.9C

設計は、目的及びその他のモジュールとの相互作用の観点から各SFR非干渉モジュールを記述しなければならない。

ADV_TDS.4.10C

マッピングは、全てのTSFIが、それらが呼び出すTOE設計で記述されているふるまいを追跡することを実証しなければならない。

評価者アクションエレメント:

ADV_TDS.4.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_TDS.4.2E

評価者は、設計が、全てのセキュリティ機能要件の正確かつ完全な具体化であることを*決定しなければならない*。

10.7.8 ADV_TDS.5 完全な準形式的モジュール設計

依存性：ADV_FSP.5 追加の誤り情報を伴う完全な準形式的機能仕様

開発者アクションエレメント:

ADV_TDS.5.1D

開発者は、TOEの設計を提供しなければならない。

ADV_TDS.5.2D

開発者は、機能仕様のTSFIからTOE設計で利用可能な分解の最下位レベルへのマッピングを提供しなければならない。

内容・提示エレメント:

ADV_TDS.5.1C

設計は、サブシステムの観点からTOEの構造を記述しなければならない。

ADV_TDS.5.2C

設計は、各モジュールをSFR実施、SFR支援、又はSFR非干渉として指示し、モジュールの観点からTSFを記述しなければならない。

ADV_TDS.5.3C

設計は、TSFの全てのサブシステムを識別しなければならない。

ADV_TDS.5.4C

設計は、適切な箇所に対して非形式的で説明的なテキストで補足される、TSFの各サブシステムの準形式的記述を提供しなければならない。

ADV_TDS.5.5C

設計は、TSFの全てのサブシステム間の相互作用の記述を提供しなければならない。

ADV_TDS.5.6C

設計は、TSFのサブシステムからTSFのモジュールへのマッピングを提供しなければならない。

ADV_TDS.5.7C

設計は、**目的、相互作用**、インタフェース、インタフェースからの戻り値、及び他のモジュールに対して呼び出されるインタフェースの観点から、**適切な箇所に対して、非形式的で説明的なテキストで補足される**、各モジュールの**準形式的記述を提供**しなければならない。

ADV_TDS.5.8C

マッピングは、全てのTSFIが、それらが呼び出すTOE設計で記述されているふるまいを追跡することを実証しなければならない。

評価者アクションエレメント:

ADV_TDS.5.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_TDS.5.2E

評価者は、設計が、全てのセキュリティ機能要件の正確かつ完全な具体化であることを*決定しなければならない*。

10.7.9 ADV_TDS.6 形式的な上位レベルの設計提示を伴う完全な準形式的モジュール設計

依存性：ADV_FSP.6 追加の形式的仕様を伴う完全な準形式的機能仕様

開発者アクションエレメント:

ADV_TDS.6.1D

開発者は、TOEの設計を提供しなければならない。

ADV_TDS.6.2D

開発者は、機能仕様のTSFIからTOE設計で利用可能な分解の最下位レベルへのマッピングを提供しなければならない。

ADV_TDS.6.3D

開発者は、TSFサブシステムの形式的仕様を提供しなければならない。

ADV_TDS.6.4D

開発者は、TSFサブシステムの形式的な仕様と機能仕様の形式的な仕様との間の対応の証明を提供しなければならない。

内容・提示エレメント:

ADV_TDS.6.1C

設計は、サブシステムの観点からTOEの構造を記述しなければならない。

ADV_TDS.6.2C

設計は、各モジュールをSFR実施、SFR支援、又はSFR非干渉として指示し、モジュールの観点からTSFを記述しなければならない。

ADV_TDS.6.3C

設計は、TSFの全てのサブシステムを識別しなければならない。

ADV_TDS.6.4C

設計は、適切な箇所に対して非形式的で説明的なテキストで補足される、TSFの各サブシステムの準形式的記述を提供しなければならない。

ADV_TDS.6.5C

設計は、TSFの全てのサブシステム間の相互作用の記述を提供しなければならない。

ADV_TDS.6.6C

設計は、TSFのサブシステムからTSFのモジュールへのマッピングを提供しなければならない。

ADV_TDS.6.7C

設計は、目的、相互作用、インタフェース、インタフェースからの戻り値、及び他のモジュールに対して呼び出されるインタフェースの観点から、適切な箇所に対して、非形式的で説明的なテキストで補足される、**準形式的なスタイル**で各モジュールを**記述**しなければならない。

ADV_TDS.6.8C

TSFサブシステムの形式的な仕様は、適切な個所に対して非形式的で説明的なテキストで補足される形式的スタイルを使用して、TSFを記述しなければならない。

ADV_TDS.6.9C

マッピングは、全てのTSFIが、それらが呼び出すTOE設計で記述されているふるまいを追跡することを実証しなければならない。

ADV_TDS.6.10C

TSFサブシステムの形式的仕様と機能仕様の形式的仕様間の対応の証明は、TOE設計に記述されている全てのふるまいがそれを呼び出しているTSFIの正確かつ完全な詳細化であることを実証しなければならない。

評価者アクションエレメント:

ADV_TDS.6.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ADV_TDS.6.2E

評価者は、設計が、全てのセキュリティ機能要件の正確かつ完全な具体化であることを*決定しなければならない*。

10.8 コンポジット設計適合(ADV_COMP)

10.8.1 目的

このファミリの目的は、関連する基本コンポーネントによって課される依存コンポーネントの要件が、コンポジット製品において満たされているかどうかを決定することである。

10.8.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

10.8.3 適用上の注釈

関連する基本コンポーネントから課される依存コンポーネントの要件は、関連する基本コンポーネント関連の利用者ガイダンス、*コンポジット評価のためのETR*(所見及び勧告の形式)及び基本コンポーネント評価監督機関の報告書(制約及び勧告の形式など)で記載される可能性がある。依存コンポーネントの開発者は、利用可能であればこれらの情報源のそれぞれを考慮し、適用される要件が満たされるように依存コンポーネントを実装しなければならない。コンポジット製品評価者は、基本コンポーネントによって課され、その評価関連文書に提供されている依存コンポーネントに関する全ての規定が、コンポジット製品によって満たされること、すなわち、依存コンポーネント開発者によって考慮されたことを検証しなければならない。

コンポジット製品評価スポンサーは、以下のものがコンポジット製品評価者のために利用できるように保証しなければならない。

— 基本コンポーネント関連の利用者ガイダンス、

— 基本コンポーネント評価者が準備した、基本コンポーネントに関連する*コンポジット評価用のETR*、

— 基本コンポーネント評価監督機関の報告書、

— 依存コンポーネント開発者が作成した証拠を含む、セキュアなコンポジット製品の実装に関する根拠。

コンポジット製品のTSFは、ADV開発クラスのファミリの中で様々な抽象度で表現される。経験上、基本コンポーネントの要件がコンポジット製品で満たされるかどうかを検査するために適切な設計表現のレベルは、TOE設計(ADV_TDS)、セキュリティアーキテクチャ

(ADV_ARC)、実装表現(ADV_IMP)である。これらの設計表現レベルが利用できない場合(例えば、選択された保証パッケージがEAL1であるため)、このファミリは適用されない(理由については次の段落を参照)。

コンポジット製品の定義上、基本コンポーネントと依存コンポーネント間のインタフェースは内部インタフェースであるため、機能仕様書(ADV_FSP)を表現レベルとして用いることは、設計の適合性を分析する上で適切ではない。

保証ファミリとしてのセキュリティアーキテクチャ(ADV_ARC)は、ドメイン分離、自己保護、非バイパス性などの統合的なセキュリティサービスが適切に機能することを保証するためのものである。関連する基本コンポーネントのアーキテクチャ内部を洞察することは不可能であり、コンポジット評価の意味するところではない(それは基本コンポーネントの評価の内容である)。ADV_ARCのコンテキストにおいて、コンポジット製品評価者は、以下のことをしなければならない。

i. 依存コンポーネントが、ドメイン分離、自己保護、非バイパス性、セキュアな初期化を提供するために、自身のコンポジット製品ST内で関連する基本コンポーネント[viii]のサービスを使用しているかどうかを決定しなければならない。使用していなければ、ADV_ARCに対するそれ以上のコンポジット評価のアクティビティはない。使用している場合は、

ii. 評価者は、依存コンポーネントが基本コンポーネントのこれらのサービスを適切かつ安全な方法で使用しているかどうかを判断しなければならない(基本コンポーネントの利用者ガイダンスを参照されたい)。

コンポジット製品のセキュリティ方針の一貫性は、保証ファミリASE_COMPのSTの文脈で既に考慮されているため、コンポジット製品のセキュリティ方針モデル(ADV_SPM)とその関連基本コンポーネントのセキュリティ方針モデルの無矛盾性を考慮する必要はない。

10.8.4 ADV_COMP.1 基本コンポーネント関連の利用者ガイダンス、コンポジット評価用のETR、及び基本コンポーネント評価監督機関報告書の設計の適合

依存性：なし

開発者アクションエレメント:

ADV_COMP.1.1D

開発者は、設計の適合に関する正当化を提供しなければならない。

内容・提示エレメント:

ADV_COMP.1.1C

設計適合性の正当化は、関連する基本コンポーネントによって課される依存コンポーネントに対する要件が、コンポジット製品においてどのように満たされるかについて、適切な表現レベルで、設計適合性の根拠を提供しなければならない。

評価者アクションエレメント:

ADV_COMP.1.1E

評価者は、設計の適合の根拠が完全であり、理路整然としており、内部的に一貫したものであることを*確認しなければならない*。

11 AGDクラス: ガイダンス文書

11.1 一般

ガイダンス文書クラスは、全ての利用者の役割に対するガイダンス証拠資料に関する要件を提供する。TOEをセキュアに準備して操作するためには、TOEのセキュアな取り扱いに関連する全ての側面を記述する必要がある。このクラスでは、TOEの意図しない間違った構成や、取り扱いについての可能性についても扱う。

多くの場合、ガイダンスはTOEの準備と操作で別々の文書として提供するか、あるいは、ソフトウェアインタフェースやハードウェアインタフェースなどを使用するエンド利用者、管理者、アプリケーションプログラマなどの様々な利用者の役割ごとに別々の文書として提供するのが適切である。

ガイダンス文書クラスは、利用者準備ガイダンス(STで記述されたように、配付されたTOEを運用環境においてその評価構成に移行するために行うべきこと)及び利用者操作ガイダンス(その評価構成においてTOEの操作中に行うべきこと)に関係する2つのファミリに分割される。

図9は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。

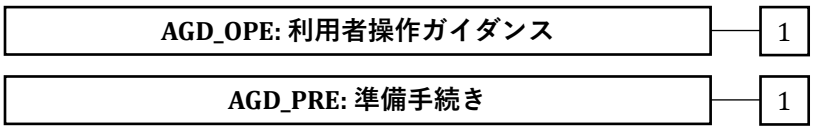


図 9 — AGD: ガイダンス文書クラスのコンポーネント構成

11.2 利用者操作ガイダンス(AGD_OPE)

11.2.1 目的

利用者操作ガイダンスは、TOEの評価構成における全てのタイプのTOE利用者、つまりエンド利用者と、最大のセキュリティを得るための適正な方法でTOEを保守、管理する責任を負う者、及びTOEの外部インタフェースを使用するその他の人々(例えば、プログラマ)に使用されることを意図して書かれた文書である。利用者操作ガイダンスは、TSFにより提供されるセキュリティ機能性を記述し、(警告を含む)指示やガイドラインを提供し、TSFの理解を助け、セキュアな使用のためにセキュリティ上重要な情報と要求されるアクションを含むものである。ガイダンス証拠資料には、誤解を招く不合理なガイダンスが含まれるべきではなく、また、全ての操作モードにおけるセキュアな手続きが示されるべきである。セキュアでない状態は、容易に検出されるべきである。

利用者操作ガイダンスは、悪意のない利用者、管理者、アプリケーションの提供者、その他TOEの外部インタフェースを使用する者が、TOEのセキュアな操作を理解し、意図されたとおりに使用することについて、信頼の指標を提供する。利用者ガイダンスの評価には、セキュアでないにもかかわらず、TOEの利用者が合理的にセキュアであると判断した方法でTOEが使用され得るかどうかの調査も含まれる。目的は、操作中の人的な誤りやそれ以外の誤りが、セキュリティ機能性の非活性化、無効化、又は活性化の失敗を招き、それによって検出されずにセキュアでない状態に陥るリスクを最小化することである。

11.2.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

11.2.3 適用上の注釈

TOEによって認識され、TSFと相互作用を行うことができる各種の利用者の役割とグループが存在することができる。これらの利用者の役割とグループは、利用者操作ガイダンスで考慮されるべきである。これらは管理者と管理者以外の利用者に大きく分類され、さらにTOEの受領、受入れ、設置、保守の担当者、アプリケーションプログラマ、修正者、監査者、日常管理、エンド利用者として責任を負うものへとより明確に分類される。各役割は、広範な一連の能力を含むか、又は単一の能力であることができる。

AGD_OPE.1.1Cの要件には、PP/STに記述されているセキュリティ課題定義と運用環境のセキュリティ対策方針に関して、TOEを運用している間の利用者に対するあらゆる警告が、利用者ガイダンスに適切に扱われているという側面が含まれている。

AGD_OPE.1.3Cで採用されているセキュアな値という概念は、利用者がセキュリティパラメタを管理している場合に関連する。ガイダンスには、このようなパラメタについて、セキュアな及びセキュアでない設定が記述される必要がある。

AGD_OPE.1.4Cは、利用者ガイダンスが、全てのセキュリティ関連事象への適切な対応を記述することを要求する。セキュリティ関連事象の多くは機能を実行した結果であるが、必ずしもそうであるとは限らない(例えば、監査ログが満杯になった場合や侵入が検知された場合)。さらに、セキュリティ関連事象は、機能の特定の連鎖の結果として起こる場合や、逆に、複数のセキュリティ関連事象が1つの機能によって誘発される場合もある。

AGD_OPE.1.7Cでは、利用者ガイダンスが明確で、合理的なものであることが要求される。誤解を招くガイダンスや不合理なガイダンスは、TOEの利用者にセキュアでないTOEをセキュアであると信じさせるおそれがある。

誤解を招くガイダンスの例として、１つのガイダンスの指示が何通りにも解釈でき、そのうちの1つで、セキュアでない状態が生じるおそれのある記述が挙げられる。

不合理なガイダンスの例として、利用者がこのガイダンスに従うことが当然のこととして期待できないほど複雑な手順を要求するものが挙げられる。

11.2.4 AGD_OPE.1 利用者操作ガイダンス

依存性：ADV_FSP.1 基本機能仕様

開発者アクションエレメント:

AGD_OPE.1.1D

開発者は、利用者操作ガイダンスを提供しなければならない。

内容・提示エレメント:

AGD_OPE.1.1C

利用者操作ガイダンスは、適切な警告を含め、セキュアな処理環境で管理するべき、利用者がアクセス可能な機能と権限について、利用者の役割ごとに記述しなければならない。

AGD_OPE.1.2C

利用者操作ガイダンスは、TOEにより提供された利用可能なインタフェースをセキュアな方法でどのように使用するかを利用者の役割ごとに記述しなければならない。

AGD_OPE.1.3C

利用者操作ガイダンスは、利用可能な機能とインタフェース、特に利用者の管理下にある全てのセキュリティパラメタを、必要に応じてセキュアな値を示し、利用者の役割ごとに記述しなければならない。

AGD_OPE.1.4C

利用者操作ガイダンスは、TSFの制御下にあるエンティティのセキュリティ特性の変更を含む、利用者がアクセス可能で実行が必要な機能に関連するセキュリティ関連事象の各タイプについて、利用者の役割ごとに明確に提示しなければならない。

AGD_OPE.1.5C

利用者操作ガイダンスは、TOEの操作の全ての可能なモード(障害や操作誤りの後の操作を含む)、それらの結果、及びセキュアな運用を維持するために必要なことを識別しなければならない。

AGD_OPE.1.6C

利用者操作ガイダンスは、STに記述された運用環境のセキュリティ対策方針を満たすために従うべきセキュリティ管理策を、利用者の役割ごとに記述しなければならない。

AGD_OPE.1.7C

利用者操作ガイダンスは、明確で、合理的なものでなければならない。

評価者アクションエレメント:

AGD_OPE.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

11.3 準備手続き (AGD_PRE)

11.3.1 目的

準備手続きは、開発者の意図したセキュアな方法でTOEが受領され、設置されたことを保証するのに有用である。準備の要件は、配付されたTOEからTOEの最初の運用環境へのセキュアな移行を要求する。これには、セキュアでないにもかかわらずTOEの利用者が合理的にセキュアであると判断した方法で、TOEを構成及び設置できるかどうかの調査も含まれる。

11.3.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

11.3.3 適用上の注釈

これらの要件の適用は、例えば、TOEが運用可能な状態で配付されるか、TOE所有者のサイトで設置しなければならないかなどの側面に応じて変動することが認知されている。

準備手続きで扱われる最初のプロセスは、開発者の配付手続きに従って受領したTOEの消費者のセキュアな受入れである。開発者が配付手続きを定義していない場合、受入れのセキュリティが別の方法で保証されなければならない。

TOEの設置には、STで提供されている運用環境のセキュリティ対策方針に準拠した状態に、TOEの運用環境を移行することが含まれる。

スマートカードなど設置が不要の場合もある。このような場合には、設置手順に対する要求や解析は不適当となりうる。

この保証ファミリの要件は、利用者操作ガイダンス(AGD_OPE)ファミリの要件とは別に提示される。これは、準備手続きがまれにしか(おそらく1回限り)使用されないからである。

11.3.4 AGD_PRE.1 準備手続き

依存性：なし

開発者アクションエレメント:

AGD_PRE.1.1D

開発者は、準備手続きを含めてTOEを提供しなければならない。

内容・提示エレメント:

AGD_PRE.1.1C

準備手続きは、開発者の配付手続きに従って配付されたTOEのセキュアな受入れに必要な全てのステップを記述しなければならない。

AGD_PRE.1.2C

準備手続きには、TOEのセキュアな設置、及びSTに記述された運用環境のセキュリティ対策方針に従った運用環境のセキュアな準備に必要な全てのステップを記述しなければならない。

評価者アクションエレメント:

AGD_PRE.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

AGD_PRE.1.2E

評価者は、TOEが運用に向けてセキュアに準備されることを確認するために、準備手続きを*適用しなければならない*。

12 ALCクラス: ライフサイクルサポート

12.1 一般

ライフサイクルサポートとは、TOEの開発、製造、配付及び保守における適切なセキュリティ管理策を確立する側面である。セキュリティ分析と証拠の作成が、開発、製造、配付及び保守アクティビティの必須部分として標準的に行われるならば、TOEのセキュリティ要件とTOEとの対応の信頼度はより大きくなる。

TOEのライフサイクルでは、TOEが開発環境と利用者環境のどちらに置かれているかということよりも、TOEの開発者と利用者のどちらの責任の下に置かれているかが区別される。移行の時点は、TOEが利用者に受け入れられるときである。ここでいう利用者とは、エンド利用者だけでなく製品インテグレータやシステムインテグレータも指す。

ALCクラスは9つのファミリで構成される。

— 開発ライフサイクル定義(ALC_LCD)は、TOEの開発、製造、配付及び保守のライフサイクルに用いられるライフサイクルモデルの開発者の記述に関する要件を提供する。

— CM能力(ALC_CMC)は、構成要素の管理に関する要件を提供する。

— CM範囲(ALC_CMS)は、構成要素の最小限のセットが、定義された方法で管理されることを要求する。

— 開発環境セキュリティ(ALC_DVS)は、開発者の物理的、論理的、手続き的、人的及びその他のセキュリティ管理策に関するものである。

— ツールと技法(ALC_TAT)は、開発者が使用する開発ツール及び実装標準に関する要件を提供する。

— 欠陥修正(ALC_FLR)は、セキュリティ上の欠陥の処理に関する要件を提供する。

— 配付(ALC_DEL)は、下流利用者へのTOEの配付に使用される手続きに関する要件を提供する。TOEの開発中に発生する配付プロセスは転送と呼ばれ、このクラスの他のファミリでの統合及び受入れ手続きで扱われる。

— ALC_TDAは、開発プロセス中の特定の成果物の生成に関するものである。

— ALC_COMPは、構成部品の統合及び配付手続きの一貫性のチェックに関するものである。

このクラスを通して、開発及び関連用語(開発者、開発する)が、より一般的な意味で開発と製造を含むように意図されている一方で、製造は、実装表現を最終的なTOEに変換するプロセスのみを意味する。

図10は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。[ix]

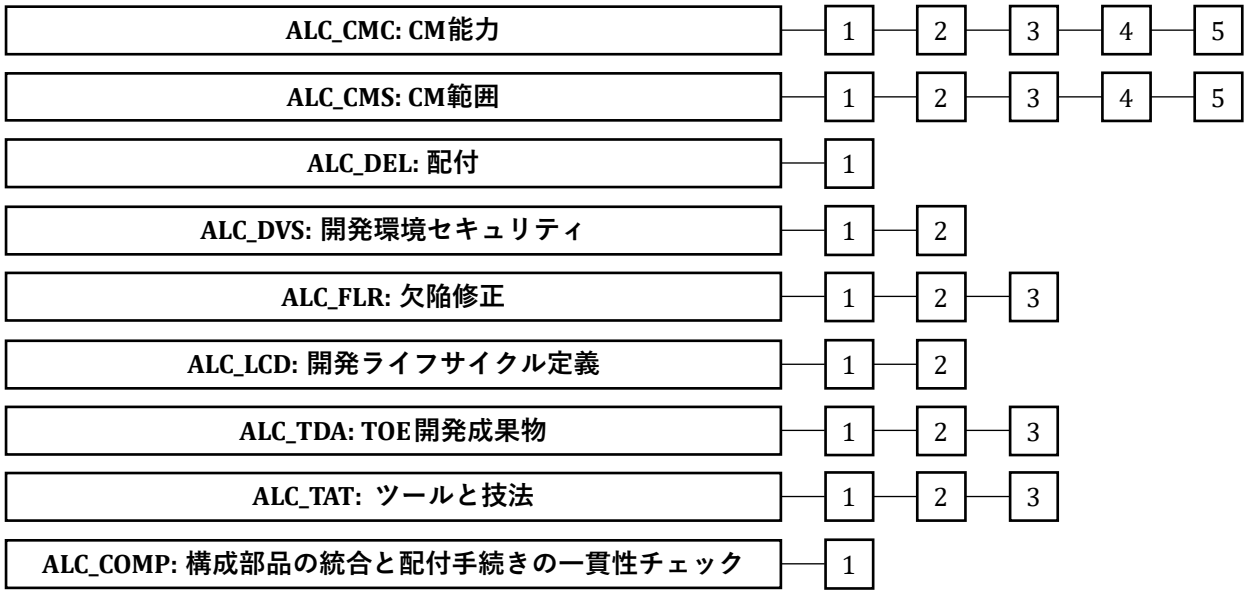


図 10 — ALC: ライフサイクルサポートクラスのコンポーネント構成

12.2 CM 能力(ALC_CMC)

12.2.1 目的

開発ライフサイクルモデルの一部として適切に定義された構成管理(CM)技術は、TOEがSFRを満たすことの保証に寄与する。正しく管理運用されたCMシステムは、TOEへのあらゆる変更を追跡し、TOEへの全ての変更が許可されたものであることの保証を支援する方法を提供することで、制御されるTOEの部分の完全性を保証するのに役立つ。

このファミリの目的は、TOE開発者のCMシステムに一定の能力を要求することである。これらの能力は、構成要素の事故による、又は許可されない修正が起きる可能性を削減することを意図している。CMシステムは、TOEのライフサイクルのうち開発者の管理下にある間について、TOEの完全性を維持することを支援するべきである。

自動化されたCMツール導入の目的は、CMシステムの効率化である。自動化されたCMシステムも手作業のCMシステムもバイパスされたり、無視されたり、許可されていない修正を防止するには不十分であるが、自動化されたシステムの方が、人の誤りや不注意に対し影響を受けにくい。

このファミリは、以下の目的を含んでいる。

a) 下流利用者に送る前に、TOEが識別可能で完全であることを保証する。

b) 評価中に、構成要素の漏れがないことを保証する。

c) TOE構成要素の許可されない修正、追加、削除を防止する。

12.2.2 コンポーネントのレベル付け

このファミリのコンポーネントは、CMシステムの能力、CM証拠資料の適用範囲、及び開発者により提供された証拠に基づいて、レベル付けされている。

12.2.3 適用上の注釈

TOEがある製品のサブセットである場合、このファミリの要件は、製品全体に適用されるのではなく、TOE構成要素のみに適用される。

TOEの設計、開発、製造及び保守の範囲で、複数のCMアプリケーションを指定するか、CMアプリケーションの異なるインスタンスを持つ開発者組織の場合、それら全てについて証拠資料への記載が要求される。評価上の目的から、CMアプリケーションのセットは、基準で扱われるTOEに適用される全体的なCMシステムの部分とみなされるべきである。

全体的なCMシステムは、コンポーネントとなるCMアプリケーション間の統合のあらゆる側面に対処するべきである。

このファミリのエレメントのいくつかは、構成要素を参照する。これらのエレメントは、構成リストで識別される全ての要素に課せられたCM要件を識別するが、リストの内容は開発者の裁量に任せている。CM範囲(ALC_CMS)は、構成リストに含まれCMシステム全体の範囲に含まれなければならない特定の要素を識別することで、この裁量を制限するために使用されることができる。

ALC_CMC.2.3Cは、CMシステムが、全ての構成要素を一意に識別することへの要件である。この要件は、構成要素への修正に対し、構成要素に新たな一意の識別情報を割り当てることを含む。

ALC_CMC.3.8Cは、CMシステムがCM計画に従って機能していることを実証する証拠への要件である。このような証拠の例は、CMシステムが出力する画面のスナップショットや監査証跡のような証拠資料、又は開発者によるCMシステムの詳細な実証である。評価者は、CMシステムがCM計画に従って機能していることを示すのにこの証拠が十分であるかを決定する責任がある。

ALC_CMC.4.5Cは、TOEの生成をサポートするための自動化された手段をCMシステムが提供することへの要件である。これは、正しい構成要素がTOEの生成に使用されているかを決定することを助けるための自動化された手段をCMシステムが提供することを要求する。

ALC_CMC.5.10Cは、TOEとその前のバージョンとの間の変更を明確にするための自動化された手段をCMシステムが提供することへの要件である。TOEの以前のバージョンが存在しない場合でも、TOEとTOEの将来のバージョンとの間の変更を明確にするための自動化された手段を開発者が提供する必要がある。

12.2.4 ALC_CMC.1 TOEのラベル付け

依存性：ALC_CMS.1 TOEのCM範囲

目的

TOEのどの段階のもの(instance)が評価されているかの観点から、曖昧さのないことを保証するために、一意の参照が要求される。TOEをその参照でラベル付けすることは、TOEの利用者がTOEのどの段階のものを使用しているかを知ることができることを保証する。

開発者アクションエレメント:

ALC_CMC.1.1D

開発者は、TOE及びTOEの一意の参照を提供しなければならない。

内容・提示エレメント:

ALC_CMC.1.1C

TOEは、その一意の参照でラベル付けされなければならない。

評価者アクションエレメント:

ALC_CMC.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.2.5 ALC_CMC.2 CMシステムの使用

依存性：ALC_CMS.1 TOEのCM範囲

目的

TOEのどの段階のもの(instance)が評価されているかの観点から、曖昧さのないことを保証するために、一意の参照が要求される。TOEをその参照でラベル付けすることは、TOEの利用者がTOEのどの段階のものを使用しているかを知ることができることを保証する。

構成要素の一意の識別は、TOEの構成をより明快に理解することにつながり、その結果、TOEのための評価要件の対象になる要素を決定することを助ける。

CMシステムの使用は、構成要素が管理された方法で維持されることの保証を高める。

開発者アクションエレメント:

ALC_CMC.2.1D

開発者は、TOE及びTOEの一意の参照を提供しなければならない。

ALC_CMC.2.2D

開発者は、CM証拠資料を提供しなければならない。

ALC_CMC.2.3D

開発者は、CMシステムを使用しなければならない。

内容・提示エレメント:

ALC_CMC.2.1C

TOEは、その一意の参照でラベル付けされなければならない。

ALC_CMC.2.2C

CM証拠資料は、構成要素を一意に識別する方法を記述しなければならない。

ALC_CMC.2.3C

CMシステムは、全ての構成要素を一意に識別しなければならない。

評価者アクションエレメント:

ALC_CMC.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.2.6 ALC_CMC.3 許可の管理

依存性：
- ALC_CMS.1 TOEのCM範囲
- ALC_DVS.1 セキュリティ手段の識別
- ALC_LCD.1 開発者によるライフサイクルプロセスの定義

目的[x]

TOEのどの段階のもの(instance)が評価されているかの観点から、曖昧さのないことを保証するために、一意の参照が要求される。TOEをその参照でラベル付けすることは、TOEの利用者がTOEのどの段階のものを使用しているかを知ることができることを保証する。

構成要素の一意の識別は、TOEの構成をより明快に理解することにつながり、その結果、TOEのための評価要件の対象になる要素を決定することを助ける。

CMシステムの使用は、構成要素が管理された方法で維持されることの保証を高める。

許可されていない修正がTOEに対して行われないことを保証する管理(「CMアクセス制御」)を提供すること、及びCMシステムの適切な機能性と使用を保証することは、TOEの完全性を維持することを助ける。

開発者アクションエレメント:

ALC_CMC.3.1D

開発者は、TOE及びTOEの一意な参照を提供しなければならない。

ALC_CMC.3.2D

開発者は、CM証拠資料を提供しなければならない。

ALC_CMC.3.3D

開発者は、CMシステムを使用しなければならない。

内容・提示エレメント:

ALC_CMC.3.1C

TOEは、その一意の参照でラベル付けされなければならない。

ALC_CMC.3.2C

CM証拠資料は、構成要素を一意に識別する方法を記述しなければならない。

ALC_CMC.3.3C

CMシステムは、全ての構成要素を一意に識別しなければならない。

ALC_CMC.3.4C

CMシステムは、許可された変更のみが構成要素に対して行われる手段を提供しなければならない。

ALC_CMC.3.5C

CM証拠資料は、CM計画を含まなければならない。

ALC_CMC.3.6C

CM計画は、TOEの開発に対してCMシステムがどのように使用されるかを記述しなければならない。

ALC_CMC.3.7C

証拠は、全ての構成要素がCMシステム下で維持されていることを実証しなければならない。

ALC_CMC.3.8C

CMシステムが、CM計画に従って機能していることを証拠により実証しなければならない。

評価者アクションエレメント:

ALC_CMC.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.2.7 ALC_CMC.4製造支援、受入れ手続き、及び自動化

依存性： ALC_CMS.1 TOEのCM範囲

ALC_DVS.1 セキュリティ手段の識別

ALC_LCD.1 開発者によるライフサイクルプロセスの定義

目的[xi]

TOEのどの段階のもの(instance)が評価されているかの観点から、曖昧さのないことを保証するために、一意の参照が要求される。TOEをその参照でラベル付けすることは、TOEの利用者がTOEのどの段階のものを使用しているかを知ることができることを保証する。

構成要素の一意の識別は、TOEの構成をより明快に理解することにつながり、その結果、TOEのための評価要件の対象になる要素を決定することを助ける。

CMシステムの使用は、構成要素が管理された方法で維持されることの保証を高める。

許可されていない修正がTOEに対して行われないことを保証することを助けるアクセス管理(「CMアクセス制御」)を提供すること、及びCMシステムの適切な機能性と使用を保証することは、TOEの完全性を維持することを助ける。

受入れ手続きの目的は、TOEの部分が適切な品質であることを保証すること、及び構成要素のいかなる生成や修正も許可されていることを確認することである。受入れ手続きは、統合プロセス及びTOEのライフサイクル管理における不可欠な要素である。

構成要素が複雑なCMシステムでは、自動化ツールなしでの変更の管理は困難である。特に、このような自動化ツールでは、開発中発生する多数の変更をサポートし、これらの変更が許可されたものであることを保証できることが必要とされる。このコンポーネントの目的は、構成要素が、自動化された手段で管理されることを保証することである。全体的なCMシステムが複数のCMアプリケーションを含む場合、自動化ツールはCMアプリケーション間の統合及びTOEの統合もサポートする。

製造サポート手続きは、特に複数の開発者が関与し、統合プロセスの実行が必要な状況で、管理された構成要素のセットからのTOEの生成が、許可された方法で正しく実行されることを保証する助けとなる。

開発者アクションエレメント:

ALC_CMC.4.1D

開発者は、TOE及びTOEの一意の参照を提供しなければならない。

ALC_CMC.4.2D

開発者は、CM証拠資料を提供しなければならない。

ALC_CMC.4.3D

開発者は、CMシステムを使用しなければならない。

内容・提示エレメント:

ALC_CMC.4.1C

TOEは、その一意の参照でラベル付けされなければならない。

ALC_CMC.4.2C

CM証拠資料は、構成要素を一意に識別する方法を記述しなければならない。

ALC_CMC.4.3C

CMシステムは、全ての構成要素を一意に識別しなければならない。

ALC_CMC.4.4C

CMシステムは、許可された変更のみが構成要素に対して行われる**自動化された**手段を提供しなければならない。

ALC_CMC.4.5C

CMシステムは、自動化された手段によってTOEの製造をサポートしなければならない。

ALC_CMC.4.6C

CM証拠資料は、CM計画を含まなければならない。

ALC_CMC.4.7C

CM計画は、TOEの開発に対してCMシステムがどのように使用されるかを記述しなければならない。

ALC_CMC.4.8C

CM計画は、改変もしくは新規に生成された構成要素をTOEの一部として受け入れるための手続きを記述しなければならない。

ALC_CMC.4.9C

証拠は、全ての構成要素がCMシステム下で維持されていることを実証しなければならない。

ALC_CMC.4.10C

CMシステムが、CM計画に従って機能していることを証拠により実証しなければならない。

評価者アクションエレメント:

ALC_CMC.4.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.2.8 ALC_CMC.5 高度なサポート

依存性： ALC_CMS.1 TOEのCM範囲

ALC_DVS.2 セキュリティ手段の十分性

ALC_LCD.1 開発者によるライフサイクルプロセスの定義

目的[xii]

TOEのどの段階のもの(instance)が評価されているかの観点から、曖昧さのないことを保証するために、一意の参照が要求される。TOEをその参照でラベル付けすることは、TOEの利用者がTOEのどの段階のものを使用しているかを知ることができることを保証する。

構成要素の一意の識別は、TOEの構成をより明快に理解することにつながり、その結果、TOEのための評価要件の対象になる要素を決定することを助ける。

CMシステムの使用は、構成要素が管理された方法で維持されることの保証を高める。

許可されていない修正がTOEに対して行われないことを保証する管理(「CMアクセス制御」)を提供すること、及びCMシステムの適切な機能性と使用を保証することは、TOEの完全性を維持することを助ける。

受入れ手続きの目的は、TOEの部分がTOEの完全性に関して定義された基準を満たしていることを保証することである。受入れ手続きの基準には、構成要素の生成や修正も許可されたものであることを確認するため、コードレビュー、脆弱性のチェック、真正性のチェック及び機能テストが含まれる場合がある。受入れ手続きは、統合プロセス及びTOEのライフサイクル管理における不可欠な要素である。

構成要素が複雑な開発環境では、自動化ツールなしでの変更の管理は困難である。特に、このような自動化ツールでは、開発中発生する多数の変更をサポートし、これらの変更が許可されたものであることを保証できることが必要とされる。このコンポーネントの目的は、構成要素が、自動化された手段で管理されることを保証することである。TOEが複数の開発者によって開発される場合、つまり統合を行う必要がある場合は、自動化ツールの使用が適切である。

製造サポート手続きは、特に複数の開発者が関与し、統合プロセスの実行が必要な状況で、管理された構成要素のセットからのTOEの生成が、許可された方法で正しく実行されることを保証する助けとなる。

CMシステムが、TOEの生成元である実装表現のバージョンを識別できることを要求することは、その資材の完全性が、適切な技術的、物理的、及び手続き的保護手段により保護されることを保証する助けとなる。

TOEのバージョン間の変更を確認する自動化された手段を提供すること、及び他の構成要素の修正によって影響を受ける構成要素を識別することは、TOEの連続するバージョン間の変更による影響を決定する際に役立つ。その結果、このことは、TOEに対する変更の結果が全ての構成要素で相互に矛盾がないかどうかを決定するために、有益な情報を提供できる。

開発者アクションエレメント:

ALC_CMC.5.1D

開発者は、TOE及びTOEの一意の参照を提供しなければならない。

ALC_CMC.5.2D

開発者は、CM証拠資料を提供しなければならない。

ALC_CMC.5.3D

開発者は、CMシステムを使用しなければならない。

内容・提示エレメント:

ALC_CMC.5.1C

TOEは、その一意の参照でラベル付けされなければならない。

ALC_CMC.5.2C

CM証拠資料は、構成要素を一意に識別する方法を記述しなければならない。

ALC_CMC.5.3C

CM証拠資料は、受入れ手続きが、全ての構成要素に対する十分で適切な変更のレビューを提供することを正当化しなければならない。

ALC_CMC.5.4C

CMシステムは、全ての構成要素を一意に識別しなければならない。

ALC_CMC.5.5C

CMシステムは、許可された変更のみが構成要素に対して行われる自動化された手段を提供しなければならない。

ALC_CMC.5.6C

CMシステムは、自動化された手段によってTOEの製造をサポートしなければならない。

ALC_CMC.5.7C

CMシステムは、構成要素をCMに受け入れる責任のある人はその開発者でないことを保証しなければならない。

ALC_CMC.5.8C

CMシステムは、TSFを構成する構成要素を識別しなければならない。

ALC_CMC.5.9C

CMシステムは、監査証跡に発信者、日時を含んでいる自動化された手段により、TOEの全ての変更についての監査をサポートしなければならない。

ALC_CMC.5.10C

CMシステムは、ある構成要素の変更により影響を受ける全ての他の構成要素を特定するための、自動化された手段を提供しなければならない。

ALC_CMC.5.11C

CMシステムは、TOEの生成元である実装表現のバージョンを識別できなければならない。

ALC_CMC.5.12C

CM証拠資料は、CM計画を含まなければならない。

ALC_CMC.5.13C

CM計画は、TOEの開発に対してCMシステムがどのように使用されるかを記述しなければならない。

ALC_CMC.5.14C

CM計画は、改変もしくは新規に生成された構成要素をTOEの一部として受け入れるための手続きを記述しなければならない。

ALC_CMC.5.15C

証拠は、全ての構成要素がCMシステム下で維持されていることを実証しなければならない。

ALC_CMC.5.16C

CMシステムが、CM計画に従って機能していることを証拠により実証しなければならない。

評価者アクションエレメント:

ALC_CMC.5.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_CMC.5.2E

評価者は、製造サポート手続きを適用するとテストアクティビティのために開発者によって提供されたTOEになることを*決定しなければならない*。

12.3 CM範囲(ALC_CMS)

12.3.1 目的

このファミリの目的は、構成要素として含まれる要素を識別することであり、それ故にCM能力(ALC_CMC)のCM要件下に置かれる。これらの追加要素に対して構成管理を適用すれば、TOEの完全性が維持されるという追加の保証を提供する。

12.3.2 コンポーネントのレベル付け

このファミリのコンポーネントは、以下のどれが構成要素として含まれることを要求されるかに基づいてレベル付けされている。TOE及びSARが要求する評価証拠、TOEの部分、実装表現、セキュリティ欠陥、開発ツール及び関連情報。

12.3.3 適用上の注釈

CM範囲(ALC_CMS)は、構成要素のリスト及びこのリスト上の各要素がCM下に置かれることを要求するが、CM能力(ALC_CMC)は、構成リストの内容を開発者の裁量に任せる。CM範囲(ALC_CMS)は、構成リストに組み込まれCM能力(ALC_CMC)のCM要件下に置かれなければならない要素を識別することで、この裁量を制限する。

12.3.4 ALC_CMS.1 TOEのCM範囲

依存性：なし

目的

CMシステムは、CM下に置かれている要素に対してのみ変更を管理することができる(すなわち構成リストで識別されている構成要素)。TOE自体及びSTの他のSARが要求する評価証拠をCM下に置くことは、これらが適切な許可を伴う管理された方法で修正がなされることの保証を提供する。

適用上の注釈

ALC_CMS.1.1Cは、TOE自体及びSTの他のSARが要求する評価証拠が、構成リストに組み込まれ、それによってCM能力(ALC_CMC)のCM要件の影響下に置かれることへの要件である。

開発者アクションエレメント:

ALC_CMS.1.1D

開発者は、TOEの構成リストを提供しなければならない。

内容・提示エレメント:

ALC_CMS.1.1C

構成リストは、TOE自体、及びSARが要求する評価証拠を含まなければならない。

ALC_CMS.1.2C

構成リストは、構成要素を一意に識別しなければならない。

評価者アクションエレメント:

ALC_CMS.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.3.5 ALC_CMS.2 TOEの一部のCM範囲

依存性：なし

目的

CMシステムは、CM下に置かれている要素に対してのみ変更を管理することができる(すなわち構成リストで識別されている構成要素)。TOE自体、TOEを構成する部分、及び他のSARが要求する評価証拠をCM下に置くことは、これらが適切な許可を伴う管理された方法で修正がなされることの保証を提供する。

適用上の注釈

ALC_CMS.2.1Cは、TOEを構成する部分(消費者に配付される全ての部分、例えばハードウェア部品や実行可能ファイル)が構成リストに組み込まれ、それによってCM能力(ALC_CMC)のCM要件の影響下に置かれることへの要件である。

ALC_CMS.2.3Cは、構成リストが各TSF関連構成要素の開発者を示すことへの要件である。

開発者アクションエレメント:

ALC_CMS.2.1D

開発者は、TOEの構成リストを提供しなければならない。

内容・提示エレメント:

ALC_CMS.2.1C

構成リストは、TOE自体、SARが要求する評価証拠、**及びTOEを構成する部分**を含まなければならない。

ALC_CMS.2.2C

構成リストは、構成要素を一意に識別しなければならない。

ALC_CMS.2.3C

各TSF関連の構成要素に対して、構成リストはその要素の開発者を示さなければならない。

評価者アクションエレメント:

ALC_CMS.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.3.6 ALC_CMS.3 実装表現のCM範囲

依存性：なし

目的

CMシステムは、CM下に置かれている要素に対してのみ変更を管理することができる(すなわち構成リストで識別されている構成要素)。TOE自体、TOEを構成する部分、TOE実装表現、及び他のSARが要求する評価証拠をCM下に置くことは、これらが適切な許可を伴う管理された方法で修正がなされることの保証を提供する。

適用上の注釈

ALC_CMS.3.1Cは、TOE実装表現が構成要素のリストに組み込まれ、それによってCM能力(ALC_CMC)のCM要件の影響下に置かれることへの要件である。

開発者アクションエレメント:

ALC_CMS.3.1D

開発者は、TOEの構成リストを提供しなければならない。

内容・提示エレメント:

ALC_CMS.3.1C

構成リストは、TOE自体、SARが要求する評価証拠、TOEを構成する部分、**及び実装表現**を含まなければならない。

ALC_CMS.3.2C

構成リストは、構成要素を一意に識別しなければならない。

ALC_CMS.3.3C

各TSF関連の構成要素に対して、構成リストはその要素の開発者を示さなければならない。

評価者アクションエレメント:

ALC_CMS.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.3.7 ALC_CMS.4 問題追跡のCM範囲

依存性：なし

目的

CMシステムは、CM下に置かれている要素に対してのみ変更を管理することができる(すなわち構成リストで識別されている構成要素)。TOE自体、TOEを構成する部分、TOE実装表現、及び他のSARが要求する評価証拠をCM下に置くことは、これらが適切な許可を伴う管理された方法で修正がなされることの保証を提供する。

CM下にセキュリティ欠陥報告を置くことは、報告の完全性が維持され、それらへのアクセスが管理されることを保証する。さらに、開発者がセキュリティ欠陥をその解決まで追跡することを支援することができる。

適用上の注釈

ALC_CMS.4.1Cは、識別されたセキュリティ欠陥の報告が構成リストに組み込まれ、それによってCM能力(ALC_CMC)のCM要件の影響下に置かれることへの要件である。このため、過去に識別されたセキュリティ上の欠陥の報告及びその解決に関する情報を維持することが要求される。

開発者アクションエレメント:

ALC_CMS.4.1D

開発者は、TOEの構成リストを提供しなければならない。

内容・提示エレメント:

ALC_CMS.4.1C

構成リストは、TOE自体、SARが要求する評価証拠、TOEを構成する部分、実装表現、**及びセキュリティ欠陥報告及び解決ステータス**を含まなければならない。

ALC_CMS.4.2C

構成リストは、構成要素を一意に識別しなければならない。

ALC_CMS.4.3C

各TSF関連の構成要素に対して、構成リストはその要素の開発者を示さなければならない。

評価者アクションエレメント:

ALC_CMS.4.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.3.8 ALC_CMS.5 開発ツールのCM範囲

依存性：なし

目的

CMシステムは、CM下に置かれている要素に対してのみ変更を管理することができる(すなわち構成リストで識別されている構成要素)。TOE自体、TOEを構成する部分、TOE実装表現、及び他のSARが要求する評価証拠をCM下に置くことは、これらが適切な許可を伴う管理された方法で修正がなされることの保証を提供する。

CM下にセキュリティ欠陥報告を置くことは、報告の完全性が維持され、それらへのアクセスが管理されることを保証する。さらに、開発者がセキュリティ欠陥をその解決まで追跡することを支援することができる。

開発ツールは、品質の高いバージョンのTOEの生成を保証するのに重要な役割を持つ。そのため、これらのツールに対する修正を管理することは重要である。

適用上の注釈

ALC_CMS.5.1Cは、開発ツール及びその他の関連情報が構成要素のリストに組み込まれ、それによってCM能力(ALC_CMC)のCM要件の影響下に置かれることへの要件である。開発ツールの例として、プログラミング言語とコンパイラが挙げられる。TOEの生成に付随する情報(コンパイラオプション、生成オプション、及び構築オプションなど)が、開発ツールに関連する情報の例である。

開発者アクションエレメント:

ALC_CMS.5.1D

開発者は、TOEの構成リストを提供しなければならない。

内容・提示エレメント:

ALC_CMS.5.1C

構成リストは、TOE自体、SARが要求する評価証拠、TOEを構成する部分、実装表現、セキュリティ欠陥報告及び解決ステータス、**及び開発ツール及び関連情報**を含まなければならない。

ALC_CMS.5.2C

構成リストは、構成要素を一意に識別しなければならない。

ALC_CMS.5.3C

各TSF関連の構成要素に対して、構成リストはその要素の開発者を示さなければならない。

評価者アクションエレメント:

ALC_CMS.5.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.4 配付(ALC_DEL)

12.4.1 目的

このファミリの関心事は、完成したTOEの開発環境から利用者の責任の下へのセキュアな転送である。

配付要件は、システム管理及び配送設備並びにTOEが利用者に配送される際にTOEのセキュリティが維持されるという保証を提供するのに必要な手段を詳述する手続きを要求する。TOEの確実な配送のため、TOEの配送に用いられる手続きは、配付中のTOEのセキュリティに関係して暗黙の又はPP/STで識別された対策方針を記述する。

12.4.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。TOEの保護レベルの上昇は、STで指定された脆弱性分析(AVA_VAN)ファミリで想定される攻撃能力に相応する配付手続きを要求することにより確立される。

12.4.3 適用上の注釈

下請業者から開発者への転送、又は異なる開発サイト間の転送は、ここでは考慮されず、開発環境セキュリティ(ALC_DVS)ファミリで考慮される。

配付フェーズは、下流利用者の責任下へのTOEの転送を受け入れることで終了する。

注：これは、TOEの下流利用者の場所への到着と必ずしも一致しない。

配付手続きは、適用できる場合、以下のような論点を考慮するべきである:

a) 消費者の受け取ったTOEが評価済みバージョンのTOEと正確に一致することを保証する。

b) 現行のバージョンのTOEに対するあらゆる改ざんを避ける又は検出する。

c) 偽のバージョンのTOEの送付を防止する。

d) 消費者に対し、TOEの配送に関する不要な知識を与えない。潜在的な攻撃者に、配付のタイミングと方法を知られるべきでない場合がある。

e) 配付中にTOEが横取りされるのを避ける又は検出する。及び

f) TOEの配送が遅らされる又は止められるのを避ける。

配付手続きは、これらの論点によって暗示されている受信者のアクションを含むべきである。これらの暗黙のアクションの一貫した記述は、存在する場合は、準備手続き(AGD_PRE)ファミリで検査される。

12.4.4 ALC_DEL.1 配付手続き

依存性：なし

開発者アクションエレメント:

ALC_DEL.1.1D

開発者は、TOE又はその一部を消費者に配付するための手続きの証拠資料を作成し提供しなければならない。

ALC_DEL.1.2D

開発者は、配付手続きを使用しなければならない。

内容・提示エレメント:

ALC_DEL.1.1C

配付証拠資料は、TOEのバージョンを消費者に配送するときにセキュリティを維持するために必要な全ての手続きを記述しなければならない。

評価者アクションエレメント:

ALC_DEL.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.5 開発環境セキュリティ（ALC_DVS）

12.5.1 目的

開発セキュリティは、開発者が提供する環境に関連するセキュリティ管理策の決定と仕様に関係する。

注：このような管理策には、資産管理、人事セキュリティ、物理及び環境セキュリティ、通信及び操作管理、アクセス管理、情報システムの取得、開発及び保守、情報セキュリティインシデント管理、事業継続管理のセキュリティ関連側面が含まれる。

12.5.2 コンポーネントのレベル付け

このファミリのコンポーネントは、セキュリティ管理策が十分であることの正当化が要求されるかどうかに基づいて、レベル付けされている。

12.5.3 適用上の注釈

このファミリは、開発者サイトに存在する脅威[xiii]及びセキュリティリスクを除去する又は減少させるための管理策を扱う。

評価者は、開発セキュリティの証拠を評定するためにサイトを訪問するべきである。これには、TOEの開発及び製造に関わる下請け業者のサイトも含まれる場合がある。訪問を行わないという決定は評価監督機関と合意されなければならない。

開発セキュリティはTOEの保守を扱っており、そのため評価の完了後に関係する内容もあるが、開発環境セキュリティ(ALC_DVS)の要件は、開発セキュリティ管理策が評価時点で適切であることのみを特定する。さらに、開発環境セキュリティ(ALC_DVS)は、評価完了後に、開発セキュリティ管理策を将来的に適用するというスポンサーの意図に関連する要件を含んでいない。

機密性は、開発環境においてTOEを保護するための論点となるとは限らない。用語「必要がある」(necessary)を使用している場合は、適切な保護手段の選択ができる。

12.5.4 ALC_DVS.1 セキュリティ管理策の識別

依存性：なし

開発者アクションエレメント:

ALC_DVS.1.1D

開発者は、開発セキュリティ証拠資料を作成し提供しなければならない。

内容・提示エレメント:

ALC_DVS.1.1C

開発セキュリティ証拠資料は、開発環境でのTOEの設計及び実装の機密性と完全性を保護するために必要となる、物理的、論理的、手続き的、人的、及びその他のセキュリティ管理策を全て記述しなければならない。

評価者アクションエレメント:

ALC_DVS.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_DVS.1.2E

評価者は、セキュリティ管理策が適用されていることを*確認しなければならない*。

12.5.5 ALC_DVS.2 セキュリティ管理策の十分性

依存性：なし

開発者アクションエレメント:

ALC_DVS.2.1D

開発者は、開発セキュリティ証拠資料を作成し提供しなければならない。

内容・提示エレメント:

ALC_DVS.2.1C

開発セキュリティ証拠資料は、開発環境でのTOEの設計及び実装の機密性と完全性を保護するために必要となる、物理的、手続き的、人的、及びその他のセキュリティ管理策を全て記述しなければならない。

ALC_DVS.2.2C

開発セキュリティ証拠資料は、セキュリティ管理策が、TOEの機密性と完全性を維持するうえで、必要な保護レベルを提供することを正当化しなければならない。

評価者アクションエレメント:

ALC_DVS.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_DVS.2.2E

評価者は、セキュリティ管理策が適用されていることを*確認しなければならない*。

12.6 欠陥修正(ALC_FLR)

12.6.1 目的

欠陥修正は、発見されたセキュリティの欠陥が開発者により追跡され訂正されることを要求する。TOE評価時に将来も欠陥修正手続きが遵守されることを決定することはできないが、

開発者が適切に、欠陥を追跡及び訂正し、欠陥情報と訂正を配布するための方針と手続きを評価することは可能である。

12.6.2 コンポーネントのレベル付け

このファミリのコンポーネントは、欠陥修正手続きの対象範囲の拡大と、欠陥修正方針の厳密さに基づいて、レベル付けされている。

12.6.3 適用上の注釈

このファミリは、TOEの開発者にTOEの欠陥を追跡及び訂正することを要求することにより、TOEが将来に渡って維持継続されることの保証を提供する。さらに、欠陥の訂正を配布するための要件も含んでいる。ただし、このファミリは、現在の評価を超えた評価要求を課すものではない。

TOE利用者は、セキュリティ欠陥に対する処置を受け取る及び実装することに責任を負う利用者組織において中心であると考えられる。これは必ずしも個々の利用者ではなく、セキュリティ欠陥の取扱いに責任を負う、組織的な代表者であってもよい。用語「TOE利用者」の使用は、異なる組織が個々の利用者でもよいしあるいは中央管理機関によって行われてもよい欠陥報告を扱うための異なる手続きを持っていることを認識する。

欠陥修正手続きは、可能性のある全てのタイプの欠陥についての対処方法を記述すべきである。これらの欠陥は、開発者、TOEの利用者、あるいはTOEについて熟知している他の機関によって報告されるかもしれない。欠陥によっては、直ちに修正できない場合がある。欠陥が修正できず、他の(例えば、手続き的な)手段が取られなければならない場合もありうる。提供された証拠資料は、運用サイトに修正を提供したり、修正が遅れている(その間何をすればよいか)又は修正ができない欠陥に関する情報を提供したりする手続きを含まなければならない。

TOEのリリース後に適用される変更は、評価されずに示されるが、元の評価の一部の情報が適用される場合もある。したがって、このファミリの中で使用される語句「TOEのリリース」は、変更が適用され認証が済んだTOEのリリースである製品のバージョンのことをいう。

12.6.4 ALC_FLR.1 基本的な欠陥修正

依存性：なし

開発者アクションエレメント:

ALC_FLR.1.1D

開発者は、TOE開発者に対する欠陥修正手続きの証拠資料を作成し提供しなければならない。

内容・提示エレメント:

ALC_FLR.1.1C

欠陥修正手続き証拠資料は、TOEのリリースごとに報告された全てのセキュリティ欠陥を追跡するのに使用する手続きを記述しなければならない。

ALC_FLR.1.2C

欠陥修正手続きは、欠陥の訂正の調査状況の記述とともに、各々のセキュリティ欠陥の性質と影響の記述が提供されることを要求しなければならない。

ALC_FLR.1.3C

欠陥修正手続きは、各々のセキュリティ欠陥の訂正アクションが識別されることを要求しなければならない。

ALC_FLR.1.4C

欠陥修正手続き証拠資料は、TOE利用者に、欠陥情報、訂正、及び訂正アクションについてのガイダンスを提供するために使用する方法を記述しなければならない。

評価者アクションエレメント:

ALC_FLR.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.6.5 ALC_FLR.2 欠陥報告手続き

依存性：なし

目的

開発者がTOE利用者からのセキュリティ欠陥報告に基づいて適切に行動することができ、かつ誰に訂正処置を送るかを知るために、TOE利用者は、開発者へセキュリティ欠陥報告を提出する方法を理解する必要がある。開発者からTOE利用者への欠陥修正ガイダンスは、TOE利用者がこの重要な情報に気づくことを保証する。

開発者アクションエレメント:

ALC_FLR.2.1D

開発者は、TOE開発者に対する欠陥修正手続きの証拠資料を作成し提供しなければならない。

ALC_FLR.2.2D

開発者は、全てのセキュリティ欠陥の報告とそれらの欠陥の訂正要求を受け付け、処理する手続きを確立しなければならない。

ALC_FLR.2.3D

開発者は、TOE利用者に対する欠陥修正ガイダンスを提供しなければならない。

内容・提示エレメント:

ALC_FLR.2.1C

欠陥修正手続き証拠資料は、TOEのリリースごとに報告された全てのセキュリティ欠陥を追跡するのに使用する手続きを記述しなければならない。

ALC_FLR.2.2C

欠陥修正手続きは、欠陥の訂正の調査状況の記述とともに、各々のセキュリティ欠陥の性質と影響の記述が提供されることを要求しなければならない。

ALC_FLR.2.3C

欠陥修正手続きは、各々のセキュリティ欠陥の訂正アクションが識別されることを要求しなければならない。

ALC_FLR.2.4C

欠陥修正手続き証拠資料は、TOE利用者に、欠陥情報、訂正、及び訂正アクションについてのガイダンスを提供するために使用する方法を記述しなければならない。

ALC_FLR.2.5C

欠陥修正手続きは、開発者がTOE利用者からの報告及びTOEの疑わしいセキュリティ欠陥に関する問合せを受け取る手段を記述しなければならない。

ALC_FLR.2.6C

報告されたセキュリティ欠陥を処理する手続きは、報告された全ての欠陥が修正され、TOE利用者に修正手続きが発行されることを保証しなければならない。

ALC_FLR.2.7C

報告されたセキュリティ欠陥を処理する手続きは、これらのセキュリティ欠陥のいかなる訂正も、新規の欠陥を引き起こすことのないよう、保護手段を提供しなければならない。

ALC_FLR.2.8C

欠陥修正ガイダンスは、TOE利用者が開発者へTOEの疑わしいセキュリティ欠陥を報告する手段を記述しなければならない。

評価者アクションエレメント:

ALC_FLR.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.6.6 ALC_FLR.3 系統的な欠陥修正

依存性：なし

目的

開発者がTOE利用者からのセキュリティ欠陥報告に基づいて適切に行動することができ、かつ誰に訂正処置を送るかを知るために、TOE利用者は、開発者にセキュリティ欠陥報告を提出する方法及び開発者がこれらの訂正処置を受け取ることができるように、開発者に対してTOE利用者自身を登録する方法を理解する必要がある。開発者からTOE利用者への欠陥修正ガイダンスは、TOE利用者がこの重要な情報に気づくことを保証する。

開発者アクションエレメント:

ALC_FLR.3.1D

開発者は、TOE開発者に対する欠陥修正手続きの証拠資料を作成し提供しなければならない。

ALC_FLR.3.2D

開発者は、全てのセキュリティ欠陥の報告とそれらの欠陥の訂正要求を受け付け、処理する手続きを確立しなければならない。

ALC_FLR.3.3D

開発者は、TOE利用者に対する欠陥修正ガイダンスを提供しなければならない。

内容・提示エレメント:

ALC_FLR.3.1C

欠陥修正手続き証拠資料は、TOEのリリースごとに報告された全てのセキュリティ欠陥を追跡するのに使用する手続きを記述しなければならない。

ALC_FLR.3.2C

欠陥修正手続きは、欠陥の訂正の調査状況の記述とともに、各々のセキュリティ欠陥の性質と影響の記述が提供されることを要求しなければならない。

ALC_FLR.3.3C

欠陥修正手続きは、各々のセキュリティ欠陥の訂正アクションが識別されることを要求しなければならない。

ALC_FLR.3.4C

欠陥修正手続き証拠資料は、TOE利用者に、欠陥情報、訂正、及び訂正アクションについてのガイダンスを提供するために使用する方法を記述しなければならない。

ALC_FLR.3.5C

欠陥修正手続きは、開発者がTOE利用者からの報告及びTOEの疑わしいセキュリティ欠陥に関する問合せを受け取る手段を記述しなければならない。

ALC_FLR.3.6C

欠陥修正手続きは、セキュリティ欠陥により影響を受ける可能性がある登録された利用者に対する、タイムリな応答、セキュリティ欠陥報告及び関連する訂正の自動配布を要求する手続きを含まなければならない。

ALC_FLR.3.7C

報告されたセキュリティ欠陥を処理する手続きは、報告された全ての欠陥が修正され、TOE利用者に修正手続きが発行されることを保証しなければならない。

ALC_FLR.3.8C

報告されたセキュリティ欠陥を処理する手続きは、これらのセキュリティ欠陥のいかなる訂正も、新規の欠陥を引き起こすことのないよう、保護手段を提供しなければならない。

ALC_FLR.3.9C

欠陥修正ガイダンスは、TOE利用者が開発者へTOEの疑わしいセキュリティ欠陥を報告する手段を記述しなければならない。

ALC_FLR.3.10C

欠陥修正ガイダンスは、TOE利用者がセキュリティ欠陥報告及び訂正を受け取る資格を得るために開発者へ登録する手段を記述しなければならない。

ALC_FLR.3.11C

欠陥修正ガイダンスは、TOEに関係するセキュリティ問題に関する全ての報告及び問合せのための特定の連絡先を識別しなければならない。

評価者アクションエレメント:

ALC_FLR.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.7 開発ライフサイクル定義(ALC_LCD)

12.7.1 目的

TOEの開発、製造及び保守に適用されるプロセスの定義が貧弱、又はプロセスが管理されていない場合、その全てのセキュリティ対策方針を満たさないTOEになってしまう可能性がある。したがってTOEのライフサイクルにおいて、できるだけ早い時期に、十分に定義され管理されたプロセスを確立することが重要である。

このようなプロセスを定義して実施することは、TOEがその全てのSFRを満たすことを保証するものではない。プロセスが不十分又は適当でない可能性もある。

開発者組織のニーズに合ったライフサイクルモデルを採用することで、TOEに適用される開発、製造及び保守プロセスが、指定されたSFRを満たすTOEの正しい設計や実装を支援する可能性が向上する。

プロセスの改善を支援するための適切なプロセス管理を決定することは、長年にわたり確立されてきたベストプラクティスである。

12.7.2 コンポーネントのレベル付け

このファミリのコンポーネントは、ライフサイクルモデルの計測可能性、及びそのモデルに準拠するための要件の増加に基づいて、レベル付けされている。

12.7.3 適用上の注釈

ライフサイクルモデルは、TOEを開発及び保守するために使用される手順、ツール、及び技法を含んでいる。このようなモデルは、設計方法、レビュー手順、プロジェクト管理の統制手段、変更管理手続き、テスト方法、及び受入れ手続きなどのプロセスの側面をカバーしている。効果的なライフサイクルモデルは、このような開発及び保守のプロセスの側面に、責任を割り当て、進捗を監視する全体的な管理機構の中で取り組んでいる。

受け入れの状況には種類があり、以下のように本基準の異なる箇所で扱われている。

— 下請け業者から納入された部品の受け入れ(「統合」)は、このファミリ、[xiv]開発ライフサイクル定義(ALC_LCD)、

— 内部の輸送後の受け入れは開発環境セキュリティ(ALC_DVS)[xv]、

— CMシステムへの部品の受け入れはCM能力(ALC_CMC)、及び

— 配付されたTOEの消費者による受け入れは配付(ALC_DEL)。

最初の3タイプは重複する可能性がある。

ライフサイクルの定義は、TOEの保守も扱っており、そのため評価完了後に関係する内容もあるが、評価時に提供されたTOEのライフサイクル情報の分析を通じて、それらも保証される。

ライフサイクルモデルが、TOEがそのセキュリティ要件を満たさないという危険を十分に最小化できるならば、このモデルは、TOEの開発及び保守で必要な管理方法を提供する。

測定可能なライフサイクルモデルとは、製品の開発特性を測定するために、管理された製品の何らかの定量的評価(数値パラメタ及び/又は数値的尺度)を使用するライフサイクルモデルである。代表的な尺度には、ソースコードの複雑性尺度、欠陥密度(コードのサイズあたりのエラー数)、又は平均故障間隔がある。セキュリティ評価の場合、それらの全ての尺度が関係を持ち、失敗の可能性を減らし、それに伴ってTOEのセキュリティの保証を増加向上させることによって、品質を上げるために使用される。

一方には標準化されたライフサイクルモデル(ウォータフォールモデルなど)、もう一方には標準化された尺度(誤り密度など)が存在し、それらが組み合わせられる可能性があることを考慮するべきである。CCは、両方の側面を定義している一つの標準に正確に従うためのライフサイクルを要求しない。

12.7.4 ALC_LCD.1 開発者によるライフサイクルプロセスの定義

依存性：なし

開発者アクションエレメント:

ALC_LCD.1.1D

開発者は、TOEの開発及び保守で使用されるライフサイクルモデルを確立しなければならない。

ALC_LCD.1.2D

開発者は、ライフサイクル定義証拠資料を提供しなければならない。

内容・提示エレメント:

ALC_LCD.1.1C

ライフサイクル定義証拠資料は、TOEの開発及び保守で使用されるプロセスを記述しなければならない。

ALC_LCD.1.2C

ライフサイクルモデルは、TOEの開発及び保守で必要な管理方法を提供しなければならない。

評価者アクションエレメント:

ALC_LCD.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.7.5 ALC_LCD.2 測定可能なライフサイクルモデル

依存性：なし

開発者アクションエレメント:

ALC_LCD.2.1D

開発者は、TOEの開発及び保守で使用される、測定可能なライフサイクルモデルに基づいたライフサイクルモデルを確立しなければならない。

ALC_LCD.2.2D

開発者は、ライフサイクル定義証拠資料を提供しなければならない。

ALC_LCD.2.3D

開発者は、測定可能なライフサイクルモデルを使用してTOEの開発を測定しなければならない。

ALC_LCD.2.4D

開発者は、ライフサイクル出力証拠資料を提供しなければならない。

内容・提示エレメント:

ALC_LCD.2.1C

ライフサイクル定義証拠資料は、TOE及び/又はTOEの開発の品質を測定するために使用された数値パラメタ及び/又は数値的尺度の詳細を含む、TOEの開発及び保守で使用されるモデルを記述しなければならない。

ALC_LCD.2.2C

ライフサイクルモデルは、TOEの開発及び保守で必要な管理方法を提供しなければならない。

ALC_LCD.2.3C

ライフサイクル出力証拠資料は、測定可能なライフサイクルモデルを使用してTOEの開発の測定結果を提供しなければならない。

評価者アクションエレメント:

ALC_LCD.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_LCD.2.2E

評価者は、TOEの開発プロセス及びTOEのセキュリティ関連特性の測定が、開発プロセス及び／又はTOE自体の改善を支援するものであることを*確認しなければならない*。

12.8 TOE開発成果物(ALC_TDA)

12.8.1 目的

このファミリは、開発プロセス又は開発に信頼を与えることを目的としている。開発プロセスにおいて、特定の成果物を生成することに焦点を当てる。これらの成果物は、後の時点で、開発プロセスがどの程度信頼できるかを評定するために使用される。この信頼は、生成された成果物を検証し、信頼できる開発の十分な証拠であることを確認することで実現される。

このファミリは、信頼できる開発を実現するために必要な成果物を生成するために、開発プロセス内に開発者のプラクティスを導入している。このファミリの要件で、必要な成果物を生成するために自動化を使用することが明示的に規定されていない場合、開発者は対応するプラクティスを手動で行うか、開発プロセスに統合された自動化を使用するか、又はその両方のハイブリッド方式を使用するかを自由に選択することができる。開発プロセスに対する信頼の度合いは、開発プロセスにおいて対応するプラクティスを実施するための自動化の導入の度合いに比例すると予想される。

また、このファミリは、ALC_TATファミリとも関係がある。ALC_TATは、TOEを開発、分析、実装するためのツールや技術の側面に焦点を当てているため、開発プロセスに導入されるこのファミリのプラクティスを説明する際に必要なコンテキストを提供する。

12.8.2 コンポーネントのレベル付け

このファミリのコンポーネントは、他のセキュリティ保証クラスの他のファミリのコンポーネントの関連する証拠との一貫性のクロスチェックを増やすことに基づいて、レベル分けされている。

12.8.3 適用上の注釈

ALC_TDA.1の要件は、TOEの生成時に実際に使用された実装表現のセットを特定する開発者の能力に対して、ある程度の信頼を与えるものである。この信頼度は、「このソフトウェアのソースコードは本当にこれなのか」、「この集積回路ハードウェアのレジスタ転送レベル(RTL)設計又は記述は本当にこれなのか」、又は「このTOEの実装表現のセットは本当にこれなのか」という、評価に潜在的に関係する質問に肯定的に答えるために役立つ。このような信頼度は、以下に基づいて構築される。

a) 実装表現識別子のセットが記録された、又はログに記録されたタイミング、

b) 実装表現識別子の記録の完全性と真正性、及び

c) 実装表現識別子のTOEからのトレーサビリティ

ALC_CMS.3により、いくつかの実装表現要素が構成リストにも含まれる場合、ALC_TDA.2の要件は、これらの実装表現要素がALC_TDA.1の実装表現識別子の使用により実際に識別可能であることを確認するものである。

ALC_TATの適用範囲にある開発ツールで使用されている実際の実装表現を正確に記録又はログすることにより、TOEの再生成が元のTOEと機能的に同等であることを第三者に納得させるための追加の証拠となる。

ALC_TDA.3の要件は、同一の実装表現から独立に生成された、目に見える差異がある可能性のある2つのTOEの間に、機能的な差異がないことを証明する機会を開発者に提供するものである。

12.8.4 ALC_TDA.1 一意に識別される実装表現

依存性：なし

開発者アクションエレメント:

ALC_TDA.1.1D

開発者は、開発ツールがTOEを生成する際に、一意のTOE実装表現識別子のリストを記録するために、TOE実装表現の個々の要素を識別しなければならない。

ALC_TDA.1.2D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストに、その時点の日時を使用したタイムスタンプを付与しなければならない。

ALC_TDA.1.3D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの完全性を維持しなければならない。

ALC_TDA.1.4D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの真正性を、(作成者の)発信元情報の維持とともに保証しなければならない。

ALC_TDA.1.5D

開発者は、TOEから、TOE生成時に記録された一意のTOE実装表現識別子のリストまで追跡することができなければならない。

ALC_TDA.1.6D

開発者は、以下の内容が記述された証拠資料を作成し提供しなければならない。

a) 開発者が、TOE生成時に記録された一意のTOE実装表現識別子のリストを作成したこと

b) TOE生成時に記録された一意のTOE実装表現識別子のリストに、開発者のタイムスタンプが適用されていること

c) TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者の)発信元情報を維持していること

d) TOE生成時に記録された一意の**TOE**実装表現識別子のリストとそれに関連するタイムスタンプ及び(作成者の)発信元情報の完全性を維持していること

e) TOEから**TOE**生成時に記録された一意の**TOE**実装表現識別子のリストを追跡するための開発者のメカニズム

内容・提示エレメント**:**

ALC_TDA.1.1C

TOE生成時に記録された一意の**TOE**実装表現識別子のリストは、**TOE**実装表現要素識別子と**TOE**実装表現要素名の対応関係を実証しなければならない。

ALC_TDA.1.2C

TOE実装表現要素名は、開発ツールが**TOE**を生成するときに使用した、又は参照したものと同じ形式でなければならない。

ALC_TDA.1.3C

TOE生成時に記録された一意の**TOE**実装表現識別子のリストのタイムスタンプは、**TOE**の生成時刻と一貫していなければならない。

ALC_TDA.1.4C

TOE生成時に記録された一意の**TOE**実装表現識別子のリストの(作成者)発信元情報は、**TOE**の(作成者)発信元情報と一貫していなければならない。作成者発信元情報は、組織の関連会社名であってもよい。

評価者アクションエレメント**:**

ALC_TDA.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_TDA.1.2E

評価者は、**TOE**を生成するための開発ツールが、実装表現要素名を使用又は参照できることを*確認しなければならない*。

ALC_TDA.1.3E

評価者は、**TOE**生成時に記録された一意の**TOE**実装表現識別子のリストが、**TOE**の生成時刻と一致することを*確認しなければならない*。

ALC_TDA.1.4E

評価者は、**TOE**生成時に記録された一意の**TOE**実装表現識別子のリストの(作成者)発信元情報が、**TOE**の(作成者)発信元情報と一致することを*確認しなければならない*。

ALC_TDA.1.5E

評価者は、**TOE**生成時に記録された一意の**TOE**実装表現識別子のリストと、それに関連するタイムスタンプ及び(作成者)発信元情報の完全性を*チェックしなければならない*。

ALC_TDA.1.6E

評価者は、**TOE**から、**TOE**生成時に記録された一意の**TOE**実装表現識別子のリストまで開発者が追跡できることを*確認しなければならない*。

12.8.5 ALC_TDA.2 実装表現のCMS範囲との一致

依存性：ALC_CMS.3 実装表現のCM範囲

開発者アクションエレメント:

ALC_TDA.2.1D

開発者は、開発ツールがTOEを生成する際に、一意のTOE実装表現識別子のリストを記録するために、TOE実装表現の個々の要素を識別しなければならない。

ALC_TDA.2.2D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストに、その時点の日時を使用したタイムスタンプを付与しなければならない。

ALC_TDA.2.3D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの完全性を維持しなければならない。

ALC_TDA.2.4D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの真正性を、(作成者の)生成情報の維持とともに保証しなければならない。

ALC_TDA.2.5D

開発者は、TOEから、TOE生成時に記録された一意のTOE実装表現識別子のリストまで追跡することができなければならない。

ALC_TDA.2.6D

開発者は、以下の内容が記述された証拠資料を作成し提供しなければならない。

a) 開発者が、TOE生成時に記録された一意のTOE実装表現識別子のリストを作成したこと

b) TOE生成時に記録された一意のTOE実装表現識別子のリストに、開発者のタイムスタンプが適用されていること

c) TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者の)発信元情報を維持していること

d) TOE生成時に記録された一意のTOE実装表現識別子のリストとそれに関連するタイムスタンプ及び(作成者の)発信元情報の完全性を維持していること

e) TOEからTOE生成時に記録された一意のTOE実装表現識別子のリストを追跡するための開発者のメカニズム

ALC_TDA.2.7D

開発者は、ALC_CMS.3の構成範囲における実装表現の要素が、TOE生成時に記録された一意のTOE実装表現識別子のリストによって識別されることを示す証拠を提供しなければならない。

内容・提示エレメント:

ALC_TDA.2.1C

TOE生成時に記録された一意のTOE実装表現識別子のリストは、TOE実装表現要素識別子とTOE実装表現要素名の対応関係を実証しなければならない。

ALC_TDA.2.2C

TOE実装表現要素名は、開発ツールがTOEを生成するときに使用した、又は参照したものと同じ形式でなければならない。

ALC_TDA.2.3C

TOE生成時に記録された一意のTOE実装表現識別子のリストのタイムスタンプは、TOEの生成時刻と一貫していなければならない。

ALC_TDA.2.4C

TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者)発信元情報は、TOEの(作成者)発信元情報と一貫していなければならない。作成者発信元情報は、組織の関連会社名であってもよい。

ALC_TDA.2.5C

ALC_CMS.3の構成範囲にある実装表現要素の識別子のリストは、TOE生成時に記録された一意のTOE実装表現の識別子のリストと一貫していなければならない。

評価者アクションエレメント:

ALC_TDA.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_TDA.2.2E

評価者は、TOEを生成するための開発ツールが、実装表現要素名を使用又は参照できることを*確認しなければならない*。

ALC_TDA.2.3E

評価者は、TOE生成時に記録された一意のTOE実装表現識別子のリストが、TOEの生成時刻と一致することを*確認しなければならない*。

ALC_TDA.2.4E

評価者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者)発信元情報が、TOEの(作成者)発信元情報と一致することを*確認しなければならない*。

ALC_TDA.2.5E

評価者は、TOE生成時に記録された一意のTOE実装表現識別子のリストと、それに関連するタイムスタンプ及び(作成者)発信元情報の完全性を*チェックしなければならない*。

ALC_TDA.2.6E

評価者は、TOEから、TOE生成時に記録された一意のTOE実装表現識別子のリストまで開発者が追跡できることを*確認しなければならない*。

ALC_TDA.2.7E

評価者は、ALC_CMS.3の構成範囲における実装表現の要素の識別子のリストが、TOE生成時に記録された一意のTOE実装表現識別子のリストと一致することを*確認しなければならない*。

12.8.6 ALC_TDA.3 適切に定義された開発ツールを用いたTOEの再生成

依存性： ALC_CMS.3 実装表現のCM範囲

ALC_TAT.1 明確に定義された開発ツール

ADV_IMP.1 TSFの実装表現

開発者アクションエレメント:

ALC_TDA.3.1D

開発者は、開発ツールがTOEを生成する際に、一意のTOE実装表現識別子のリストを記録するために、TOE実装表現の個々の要素を識別しなければならない。

ALC_TDA.3.2D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストに、その時点の日時を使用したタイムスタンプを付与しなければならない。

ALC_TDA.3.3D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの完全性を維持しなければならない。

ALC_TDA.3.4D

開発者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの真正性を、(作成者の)生成情報の維持とともに保証しなければならない。

ALC_TDA.3.5D

開発者は、TOEから、TOE生成時に記録された一意のTOE実装表現識別子のリストまで追跡することができなければならない。

ALC_TDA.3.6D

開発者は、以下の内容が記述された証拠資料を作成し提供しなければならない。

a) 開発者が、TOE生成時に記録された一意のTOE実装表現識別子のリストを作成したこと

b) TOE生成時に記録された一意のTOE実装表現識別子のリストに、開発者のタイムスタンプが適用されていること

c) TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者の)発信元情報を維持していること

d) TOE生成時に記録された一意のTOE実装表現識別子のリストとそれに関連するタイムスタンプ及び(作成者の)発信元情報の完全性を維持していること

e) TOEからTOE生成時に記録された一意のTOE実装表現識別子のリストを追跡するための開発者のメカニズム

ALC_TDA.3.7D

開発者は、ALC_CMS.3の構成範囲における実装表現の要素が、TOE生成時に記録された一意のTOE実装表現識別子のリストによって識別されることを示す証拠を提供しなければならない。

ALC_TDA.3.8D

一意のTOE実装表現識別子のリストに従って、TOE実装表現の別のコピーに開発ツールを適用してTOEのコピーを再生成した後、TOEのコピーと元のTOEとの間に機能的な差異がある場合、開発者は、その差異を説明しなければならない。

ALC_TDA.3.9D

開発者は、再生成されたTOEコピーとオリジナルのTOEとの間に機能的な差異がある場合、それを説明する証拠資料を作成し、提供しなければならない。

内容・提示エレメント:

ALC_TDA.3.1C

TOE生成時に記録された一意のTOE実装表現識別子のリストは、TOE実装表現要素識別子とTOE実装表現要素名の対応関係を実証しなければならない。

ALC_TDA.3.2C

TOE実装表現要素名は、開発ツールがTOEを生成するときに使用した、又は参照したものと同じ形式でなければならない。

ALC_TDA.3.3C

TOE生成時に記録された一意のTOE実装表現識別子のリストのタイムスタンプは、TOEの生成時刻と一貫していなければならない。

ALC_TDA.3.4C

TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者)発信元情報は、TOEの(作成者)発信元情報と一貫していなければならない。作成者発信元情報は、組織の関連会社名であってもよい。

ALC_TDA.3.5C

ALC_CMS.3の構成範囲にある実装表現要素の識別子のリストは、TOE生成時に記録された一意のTOE実装表現の識別子のリストと一貫していなければならない。

ALC_TDA.3.6C

再生成されたTOEのコピーと元のTOEとの間に機能的差異がある場合、開発者による説明は、再生成されたTOEのコピーと元のTOEとの間に目に見える差異がある場合、その差異は全て考慮されなければならない。

評価者アクションエレメント:

ALC_TDA.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_TDA.3.2E

評価者は、TOEを生成するための開発ツールが、実装表現要素名を使用又は参照できることを*確認しなければならない*。

ALC_TDA.3.3E

評価者は、TOE生成時に記録された一意のTOE実装表現識別子のリストが、TOEの生成時刻と一致することを*確認しなければならない*。

ALC_TDA.3.4E

評価者は、TOE生成時に記録された一意のTOE実装表現識別子のリストの(作成者)発信元情報が、TOEの(作成者)発信元情報と一致することを*確認しなければならない*。

ALC_TDA.3.5E

評価者は、TOE生成時に記録された一意のTOE実装表現識別子のリストと、それに関連するタイムスタンプ及び(作成者)発信元情報の完全性を*チェックしなければならない*。

ALC_TDA.3.6E

評価者は、TOEから、TOE生成時に記録された一意のTOE実装表現識別子のリストまで開発者が追跡できることを*確認しなければならない*。

ALC_TDA.3.7E

評価者は、ALC_CMS.3の構成範囲における実装表現の要素の識別子のリストが、TOE生成時に記録された一意のTOE実装表現識別子のリストと一致することを*確認しなければならない*。

ALC_TDA.3.8E

評価者は、再生成されたTOEのコピーと元のTOEの間に機能的な差異がある場合、開発者の説明が、再生成されたTOEのコピーと元のTOEの間の目に見える差異を全て考慮していることを*チェックしなければならない*。

12.9 ツールと技法(ALC_TAT)

12.9.1 目的

ツールと技法は、TOEの開発、分析、及び実装に使用されるツールの選択に関連する。これは、TOEの開発時に不明確な、一貫性がない、もしくは不正確な開発ツールが使用されるのを防止する要件を含む。これは、プログラミング言語、証拠資料、実装標準、及びサポートするランタイムライブラリのようなTOEの部分も含むが、これらに限定されない。

12.9.2 コンポーネントのレベル付け

このファミリのコンポーネントは、記述と実装標準、及び実装に依存するオプションの証拠資料の範囲に関する要件の増加に基づいて、レベル付けされている。

12.9.3 適用上の注釈

明確に定義された開発ツールが要求される。これらは、明確かつ完全に記述されたツールである。例えば、標準化組織などにより発行された標準に基づいているプログラム言語やCADシステムは、明確に定義されていると考えられる。自己製らのツールは、それらが明確に定義されているかどうかを明確化するために、さらに調査が必要である。

ALC_TAT.1.2.Cの要件は、ソースコードの全てのステートメントが、曖昧でない意味を持つことを保証するために、特にプログラミング言語に適用可能である。

ALC_TAT.2及びALC_TAT.3では、実装ガイドラインが専門家のグループ(例えば、学術専門家や標準化組織)で認められている場合に、それらが実装標準として受け入れられる。実装標準は、通常は特定の業界で公然と十分に受け入れられている一般的な実践であるが、開発者固有の実装ガイドラインも標準として受け入れられる場合があり、専門性に重点が置かれている。

ツールと技法は、開発者が適用する実装標準(ALC_TAT.2.3D)と、サードパーティのソフトウェア、ハードウェア、又はファームウェアを含む「TOEの全ての部分」についての実装標準(ALC_TAT.3.3D)とを区別している。CM範囲(ALC_CMS)で導入されている構成リストでは、各TSF関連の構成要素の生成元がTOE開発者なのかサードパーティの開発者なのかを示す必要がある。

12.9.4 ALC_TAT.1 明確に定義された開発ツール

依存性：ADV_IMP.1 TSFの実装表現

開発者アクションエレメント:

ALC_TAT.1.1D

開発者は、TOEに対して使用される各開発ツールを識別する証拠資料を提供しなければならない。

ALC_TAT.1.2D

開発者は、各開発ツールのオプションの中で実装に依存するものについて証拠資料を作成し提供しなければならない。

内容・提示エレメント:

ALC_TAT.1.1C

実装に使用される各開発ツールは、明確に定義されていなければならない。

ALC_TAT.1.2C

各開発ツールの証拠資料は、実装に使用される全てのステートメントの意味、及び規約と指示文を曖昧さなく定義しなければならない。

ALC_TAT.1.3C

各開発ツールの証拠資料は、実装に依存する全てのオプションの意味を、曖昧さなく定義しなければならない。

評価者アクションエレメント:

ALC_TAT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

12.9.5 ALC_TAT.2 実装標準への準拠

依存性：ADV_IMP.1 TSFの実装表現

開発者アクションエレメント:

ALC_TAT.2.1D

開発者は、TOEに対して使用される各開発ツールを識別する証拠資料を提供しなければならない。

ALC_TAT.2.2D

開発者は、各開発ツールのオプションの中で実装に依存するものについて証拠資料を作成し提供しなければならない。

ALC_TAT.2.3D

開発者は、開発者が適用している実装標準を記述し提供しなければならない。

内容・提示エレメント:

ALC_TAT.2.1C

実装に使用される各開発ツールは、明確に定義されていなければならない。

ALC_TAT.2.2C

各開発ツールの証拠資料は、実装に使用される全てのステートメントの意味、及び規約と指示文を曖昧さなく定義しなければならない。

ALC_TAT.2.3C

各開発ツールの証拠資料は、実装に依存する全てのオプションの意味を、曖昧さなく定義しなければならない。

評価者アクションエレメント:

ALC_TAT.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_TAT.2.2E

評価者は、実装標準が適用されていることを*確認しなければならない*。

12.9.6 ALC_TAT.3 実装標準への準拠 - 全ての部分

依存性：ADV_IMP.1 TSFの実装表現

開発者アクションエレメント:

ALC_TAT.3.1D

開発者は、TOEに対して使用される各開発ツールを識別する証拠資料を提供しなければならない。

ALC_TAT.3.2D

開発者は、各開発ツールのオプションの中で実装に依存するものについて証拠資料を作成し提供しなければならない。

ALC_TAT.3.3D

開発者は、**TOEの全ての部分に対して**開発者**及びサードパーティプロバイダ**が適用している実装標準を記述し提供しなければならない。

内容・提示エレメント:

ALC_TAT.3.1C

実装に使用される各開発ツールは、明確に定義されていなければならない。

ALC_TAT.3.2C

各開発ツールの証拠資料は、実装に使用される全てのステートメントの意味、及び規約と指示文を曖昧さなく定義しなければならない。

ALC_TAT.3.3C

各開発ツールの証拠資料は、実装に依存する全てのオプションの意味を、曖昧さなく定義しなければならない。

評価者アクションエレメント:

ALC_TAT.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_TAT.3.2E

評価者は、実装標準が適用されていることを*確認しなければならない*。

12.10 構成部品の統合と配付手続きの一貫性チェック（ALC_COMP)

12.10.1 目的

このファミリの目的は、次のことを決定することである。

— 依存コンポーネントの正しいバージョンが、関連する基本コンポーネントの正しいバージョンに設置/組み込みされているかどうか、及び

— 基本コンポーネント開発者及び依存コンポーネント開発者の準備ガイダンスの手続きが、コンポジット製品インテグレータの受入れ手続きと互換性があるかどうか。

12.10.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

12.10.3 適用上の注釈

コンポジット製品評価者は、評価されている依存コンポーネントの正しいバージョンが、コンポジット製品の関連基本コンポーネントの評価バージョンに設置/組み込みされていることを検証しなければならない。

コンポジット製品評価スポンサーは、コンポジット製品インテグレータが生成しなければならない適切な証拠が、コンポジット製品評価者に利用可能であることを保証しなければならない。この証拠には、特に基本コンポーネント開発者の構成リスト(例えば、開発者の承認ステートメントの中で提供されたもの)が含まれる場合がある。

コンポジット製品評価者は、基本コンポーネント開発者及び依存コンポーネント開発者の配付手続きが、コンポジット製品インテグレータが使用する受入れ手続きと互換性があることを検証しなければならない。

コンポジット製品評価者は、基本コンポーネント開発者及び依存コンポーネント開発者が規定する全ての構成パラメタ(例えば、プリパーソナライゼーションデータ、プリパーソナライゼーションスクリプト)が、コンポジット製品インテグレータによって使用されていることを検証しなければならない。

コンポジット製品評価スポンサーは、コンポジット製品インテグレータが生成しなければならない適切な証拠が、コンポジット製品評価者に利用可能であることを保証しなければならない。この証拠には、特に、基本コンポーネント開発者による依存コンポーネントの受け取り、受入れ、パラメタリゼーションの証拠(例えば、開発者の承認ステートメントの形式)を含めることができる。

12.10.4 ALC_COMP.1 関連基本コンポーネントへの依存コンポーネントの統合及び配付及び受入れ手続きの一貫性チェック

依存性：なし

開発者アクションエレメント:

ALC_COMP.1.1D

開発者は、コンポーネント構成の証拠資料を提供しなければならない。

内容・提示エレメント:

ALC_COMP.1.1C

コンポーネント構成の証拠資料は、依存コンポーネントの評価版が、関連する基本コンポーネントの評価版に設置/組み込みされたことを示されなければならない。

ALC_COMP.1.2C

コンポーネント構成の証拠資料は、以下のことを示さなければならない。

a) 基本コンポーネント開発者及び依存コンポーネント開発者の配付手続きが、コンポジット製品インテグレータの受入れ手続きと互換性があることを、配付及び受入れの互換性の証拠資料によって示さなければならない。

b) 証拠資料は、基本コンポーネント開発者及び依存コンポーネント開発者が規定する準備ガイダンスの手続きが、コンポジット製品インテグレータによって実際に使用されている、又は、コンポジット製品インテグレータのガイダンスと互換性があり、互いに矛盾していないことを示さなければならない。

評価者アクションエレメント:

ALC_COMP.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ALC_COMP.1.2E

評価者は、配付の互換性に関する証拠が完全であり、理路整然としており、内部的に一貫したものであることを*確認しなければならない*。

13 ATEクラス: テスト

13.1 一般

「テスト」クラスは、5つのファミリを含む。つまり、カバレージ(ATE_COV)、深さ(ATE_DPT)、独立テスト(即ち、評価者によって実行される機能テスト)(ATE_IND)、機能テスト(ATE_FUN)及びコンポジット機能テスト(ATE_COMP)である。テストは、TSFが記述(機能仕様、TOE設計、及び実装表現)に従ってふるまうことの保証を提供し、テストシナリオにおけるSFRの直接的な追跡可能性を実現する。

このクラスは、TSFがその設計記述に従って動作することの確認に重点を置いている。このクラスでは、TSFの設計及び実装での脆弱性の識別を特に求めるTSFの分析に基づく侵入テストを扱わない。侵入テストは、AVA: 脆弱性評定クラスで、脆弱性評定の1つの側面として別に述べられている。

ATE: テストクラスでは、テストが開発者テストと評価者テストに分けられる。カバレージ(ATE_COV)ファミリと深さ(ATE_DPT)ファミリは、開発者テストの完全性を扱う。カバレージ(ATE_COV)は機能仕様がテストされるときの厳格性を扱い、深さ(ATE_DPT)は他の設計記述(セキュリティアーキテクチャ、TOE設計及び実装表現)に対するテストが必要かどうかを扱う。

機能テスト(ATE_FUN)は、開発者によるこれらのテストの実行、及びこのテストについてどのように証拠資料を提出するべきかを扱う。最後に、独立テスト(ATE_IND)は、評価者テスト、つまり評価者が開発者テストの一部又は全部を繰り返すべきであるか、及び評価者がどの程度の量の独立テストを実行するべきであるかを扱う。

コンポジット機能テスト(ATE_COMP)は、コンポジット製品が全体としてSTの機能要件を満たすために必要な特性を示しているかを決定する。

図11は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。

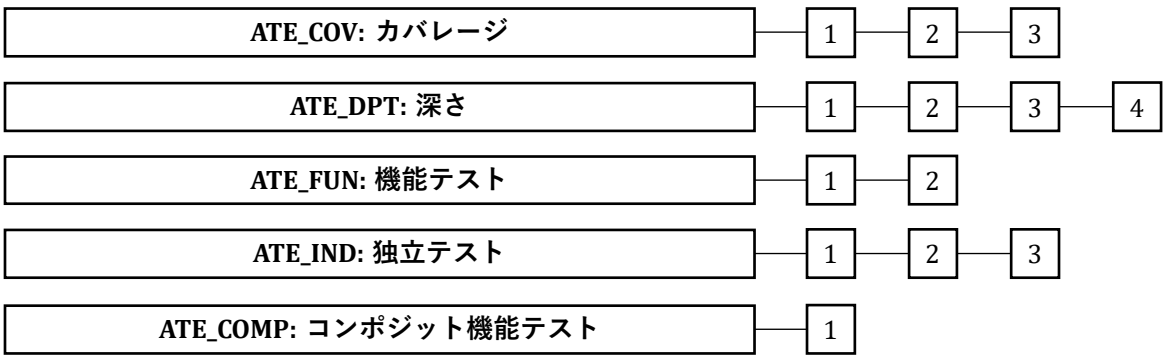


図 11 — ATE: テストクラスのコンポーネント構成

13.2 カバレージ(ATE_COV)

13.2.1 目的

このファミリはTSFがその機能仕様に対してテストされていることを確認する。これは、開発者の対応証拠の検査を通して達成される。

13.2.2 コンポーネントのレベル付け

このファミリのコンポーネントは、仕様に基づいてレベル付けされている。

13.2.3 適用上の注釈

13.2.4 ATE_COV.1 カバレージの証拠

依存性： ADV_FSP.2 セキュリティ実施機能仕様

ATE_FUN.1 機能テスト

目的

このコンポーネントの目的は、TSFIの一部がテストされていることを確認することである。

適用上の注釈

このコンポーネントでは、開発者は、テスト証拠資料内のテストが機能仕様内のTSFIとどのように対応するかを示す。これは、対応のステートメント(多分、表を使うこと)により達成可能である。

開発者アクションエレメント:

ATE_COV.1.1D

開発者は、テストカバレージの証拠を提供しなければならない。

内容・提示エレメント:

ATE_COV.1.1C

テストカバレージの証拠は、テスト証拠資料におけるテストと機能仕様における TSFI との間の対応を提示しなければならない。

評価者アクションエレメント:

ATE_COV.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.2.5 ATE_COV.2 カバレージの分析

依存性： ADV_FSP.2 セキュリティ実施機能仕様

ATE_FUN.1 機能テスト

目的

このコンポーネントの目的は、TSFIの全てがテストされていることを確認することである。

適用上の注釈

このコンポーネントでは、開発者は、テスト証拠資料内のテストが機能仕様内の全てのTSFIと対応していることを確認する。これは、対応のステートメント(多分表を使うこと)によって達成できるが、開発者はテストカバレージの分析も提供する。

開発者アクションエレメント:

ATE_COV.2.1D

開発者は、テストカバレージの**分析**を提供しなければならない。

内容・提示エレメント:

ATE_COV.2.1C

テストカバレージの**分析**は、テスト証拠資料におけるテストと機能仕様における**TSFI**との間の対応を**実証**しなければならない。

ATE_COV.2.2C

テストカバレージの分析は、機能仕様における全ての*TSFI*がテストされていることを実証しなければならない。

評価者アクションエレメント:

ATE_COV.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.2.6 ATE_COV.3 カバレージの厳格な分析

依存性： ADV_FSP.2 セキュリティ実施機能仕様

ATE_FUN.1 機能テスト

目的

このコンポーネントの目的は、機能仕様内の全てのインタフェースについて開発者が徹底的なテストを実行したことを確認することである。

このコンポーネントの目的は、全ての**TSFI**の全てのパラメタがテストされていることを確認することである。

適用上の注釈

このコンポーネントでは、開発者に対し、テスト証拠資料内のテストが機能仕様内の全ての**TSFI**とどのように対応するかを示すことが要求される。これは、対応のステートメント(多分表を使うこと)によって達成できるが、さらに開発者には、全ての**TSFI**の全てのパラメタに対してテストが実行されることを実証することが要求される。この追加の要件には、境界テスト(つまり、指定された限界を超えたときに誤りが生成されることの検証)、及び否定テスト(例えば、利用者**A**にアクセス権が与えられるときに、利用者**A**がアクセスできるようになったことだけでなく、利用者**B**が突然アクセスできるようにはならないことを検証する)が含まれる。この種のテストは厳密には徹底的ではない。なぜなら、パラメタの全ての可能な値がチェックされるようにはなっていないからである。

開発者アクションエレメント:

ATE_COV.3.1D

開発者は、テストカバレージの分析を提供しなければならない。

内容・提示エレメント:

ATE_COV.3.1C

テストカバレージの分析は、テスト証拠資料におけるテストと機能仕様における**TSFI**との間の対応を実証しなければならない。

ATE_COV.3.2C

テストカバレージの分析は、機能仕様における全ての**TSFI**が**完全に**テストされていることを実証しなければならない。

評価者アクションエレメント:

ATE_COV.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.3 深さ (ATE_DPT)

13.3.1 目的

このファミリのコンポーネントは、開発者によるTSFテストの詳細レベルを扱う。TSFのテストは、追加の設計表現及び記述(TOE設計、実装表現、及びセキュリティアーキテクチャ記述)から引き出される情報の深さに基づいている。

目的は、TOEの開発中の誤りを見逃すリスクに対抗することである。特定の内部インタフェースを使用するテストは、TSFが望ましい外部的なセキュリティのふるまいを示すことの保証だけではなく、このふるまいが内部機能性の正常な動作に起因していることの保証も提供することができる。

13.3.2 コンポーネントのレベル付け

このファミリのコンポーネントは、TOE設計から実装表現までのTSF表現で提供される詳細の量に基づいて、レベル付けされている。このレベルは、ADVクラスで提供されたTSF表現を反映する。

13.3.3 適用上の注釈

TOE設計は、内部コンポーネント(例えばサブシステム)、及び場合によってはTSFのモジュールを、それらのコンポーネント及びモジュール間のインタフェースとともに記述する。このTOE設計のテストの証拠は、内部インタフェースが使用され、記述どおりに動作することが確認されたことを示さなければならない。これは、TSFの外部インタフェースを介したテストを通じて、又はテストハーネスを使用するなどしてTOEサブシステムもしくはモジュールインタフェースを単独でテストすることによって達成される。内部インタフェースのある側面が外部インタフェースを介してテストできない場合は、それらの側面のテストが不要であること、又は内部インタフェースを直接テストする必要があることが正当化されるべきである。後者の場合は、直接テストを容易にするために、TOE設計が十分に詳細化されている必要がある。

TSFのアーキテクチャの健全性の記述(セキュリティアーキテクチャ(ADV_ARC)での)で特定のメカニズムが挙げられている場合、開発者が実行するテストは、メカニズムが実行され、記述どおりに動作していることを示さなければならない。

このファミリの最上位のコンポーネントでは、TOE設計に対してだけではなく、実装表現に対してもテストが実行される。

13.3.4 ATE_DPT.1テスト: 基本設計

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_TDS.2 アーキテクチャ設計

ATE_FUN.1 機能テスト

目的

TSFのサブシステム記述は、TSFの内部動作に関する上位レベルの記述を提供する。TOEサブシステムのレベルでのテストは、TSFサブシステムが、TOE設計及びセキュリティアーキテクチャ記述で記述されたとおり動作及び相互作用することへの保証を提供する。

開発者アクションエレメント:

ATE_DPT.1.1D

開発者は、テストの深さの分析を提供しなければならない。

内容・提示エレメント:

ATE_DPT.1.1C

テストの深さの分析は、テスト証拠資料におけるテストとTOE設計におけるTSFサブシステムの間の対応を実証しなければならない。

ATE_DPT.1.2C

テストの深さの分析は、TOE設計内の全てのTSFサブシステムがテストされていることを実証しなければならない。

評価者アクションエレメント:

ATE_DPT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.3.5 ATE_DPT.2 テスト: セキュリティ実施モジュール

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_TDS.3 基本モジュール設計

ATE_FUN.1 機能テスト

目的

TSFのサブシステム及びモジュール記述は、TSFの内部動作に関する上位レベルの記述、及びSFR実施モジュールのインタフェースの記述を提供する。TOE記述のこのレベルでのテストは、TSFサブシステム及びSFR実施モジュールが、TOE設計及びセキュリティアーキテクチャ記述で記述されたとおり動作及び相互作用することへの保証を提供する。

開発者アクションエレメント:

ATE_DPT.2.1D

開発者は、テストの深さの分析を提供しなければならない。

内容・提示エレメント:

ATE_DPT.2.1C

テストの深さの分析は、テスト証拠資料におけるテストとTOE設計におけるTSFサブシステム及びSFR実施モジュールの間の対応を実証しなければならない。

ATE_DPT.2.2C

テストの深さの分析は、TOE設計内の全てのTSFサブシステムがテストされていることを実証しなければならない。

ATE_DPT.2.3C

テストの深さの分析は、TOE設計内のSFR実施モジュールがテストされていることを実証しなければならない。

評価者アクションエレメント:

ATE_DPT.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.3.6 ATE_DPT.3 テスト: モジュール設計

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_TDS.4 準形式的モジュール設計

ATE_FUN.1 機能テスト

目的

TSFのサブシステム及びモジュール記述は、TSFの内部動作に関する上位レベルの記述、及びモジュールのインタフェースの記述を提供する。TOE記述のこのレベルでのテストは、TSFサブシステム及びモジュールが、TOE設計及びセキュリティアーキテクチャ記述で記述されたとおり動作及び相互作用することへの保証を提供する。

開発者アクションエレメント:

ATE_DPT.3.1D

開発者は、テストの深さの分析を提供しなければならない。

内容・提示エレメント:

ATE_DPT.3.1C

テストの深さの分析は、テスト証拠資料におけるテストとTOE設計におけるTSFサブシステム及びモジュールの間の対応を実証しなければならない。

ATE_DPT.3.2C

テストの深さの分析は、TOE設計内の全てのTSFサブシステムがテストされていることを実証しなければならない。

ATE_DPT.3.3C

テストの深さの分析は、TOE設計内の**全てのTSF**モジュールがテストされていることを実証しなければならない。

評価者アクションエレメント:

ATE_DPT.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.3.7 ATE_DPT.4 テスト: 実装表現

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_TDS.4 準形式的モジュール設計

ADV_IMP.1 TSFの実装表現

ATE_FUN.1 機能テスト

目的

TSFのサブシステム及びモジュール記述は、TSFの内部動作に関する上位レベルの記述、及びモジュールのインタフェースの記述を提供する。TOE記述のこのレベルでのテストは、

TSFサブシステム及びモジュールが、TOE設計及びセキュリティアーキテクチャ記述で記述されたとおり、及び実装表現に従って動作及び相互作用することへの保証を提供する。

開発者アクションエレメント:

ATE_DPT.4.1D

開発者は、テストの深さの分析を提供しなければならない。

内容・提示エレメント:

ATE_DPT.4.1C

テストの深さの分析は、テスト証拠資料におけるテストとTOE設計におけるTSFサブシステム及びモジュールの間の対応を実証しなければならない。

ATE_DPT.4.2C

テストの深さの分析は、TOE設計内の全てのTSFサブシステムがテストされていることを実証しなければならない。

ATE_DPT.4.3C

テストの深さの分析は、TOE設計内の全てのTSFモジュールがテストされていることを実証しなければならない。

ATE_DPT.4.4C

テストの深さの分析は、TSFがその実装表現に従って動作することを実証しなければならない。

評価者アクションエレメント:

ATE_DPT.4.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.4 機能テスト(ATE_FUN)

13.4.1 目的

開発者によって実行される機能テストは、テスト証拠資料におけるテストが正しく実行されて証拠資料として提出されることの保証を提供する。TSFの設計記述に対するこれらのテストの対応は、カバレージ(ATE_COV)ファミリ及び深さ(ATE_DPT)ファミリを通じて達成される。

このファミリは、未発見の欠点の公算が比較的少ないという保証を提供するのに寄与する。

カバレージ(ATE_COV)、深さ(ATE_DPT)、機能テスト(ATE_FUN)のファミリは、開発者により提供されるべきテストの証拠を定義するのに組み合わせて使用される。評価者による独立機能テストは、独立テスト(ATE_IND)で特定される。

13.4.2 コンポーネントのレベル付け

このファミリは、2つのコンポーネントを含み、上位は順序依存性を分析することを要求する。

13.4.3 適用上の注釈

テスト遂行の手順は、テスト環境、テスト条件、テストデータのパラメタと値を含むテストプログラムとテストスイートを使うための指示を提供することを期待されている。テスト手順は、またテスト入力からテスト結果がどのように引き出されるかを示すべきである。

順序依存性は、特定のテストの実行がうまくいくかどうかが、特定の状態の存在に依存する場合に関係する。例えば、テストAの実行の成功から生じる状態がテストBの実行の成功に必須であるため、順序依存性は、テストAがテストBの直前に実行されることを要求する。このようにして、テストBの失敗が順序依存性の問題に関係しているかも知れない。前述の例で、テストBは、テストAではなくてテストCがそれの直前に実行されたために失敗するかも知れない、又はテストBの失敗はテストAの失敗に関係しているかも知れない。

13.4.4 ATE_FUN.1 機能テスト

依存性：ATE_COV.1 カバレージの証拠

目的

目的は、テスト証拠資料におけるテストが正しく実行されて証拠資料として提出されることを開発者が実証することである。

開発者アクションエレメント:

ATE_FUN.1.1D

開発者は、TSFをテストし、結果を証拠資料で提出しなければならない。

ATE_FUN.1.2D

開発者は、テスト証拠資料を提供しなければならない。

内容・提示エレメント:

ATE_FUN.1.1C

テスト証拠資料は、テスト計画、期待されるテスト結果、及び実際のテスト結果から構成されなければならない。

ATE_FUN.1.2C

テスト計画は、実行されるべきテストを識別し、各テストを実行するシナリオを記述しなければならない。これらのシナリオは、他のテストの結果への全ての順序依存性を含んでいなければならない。

ATE_FUN.1.3C

期待されるテスト結果は、テストの実行が成功したときの予期される出力を示さなければならない。

ATE_FUN.1.4C

実際のテスト結果は、期待されたテスト結果と一貫していなければならない。

評価者アクションエレメント:

ATE_FUN.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.4.5 ATE_FUN.2 順序付けられた機能テスト

依存性：ATE_COV.1 カバレージの証拠

目的

目的は、開発者が、テスト証拠資料におけるテストが正しく実行され、証拠資料として提出されることを実証すること、及びテスト対象のインタフェースの正しさに関する論証の堂々巡りを回避するようにテストが構成されることを保証することである。

適用上の注釈

テスト手順は、テストの順序に関して必須の初期テスト条件を記述できるかもしれないが、順序の根拠が提供されていない場合がある。テスト順序の分析は、テスト順序に障害が隠れている可能性があることから、テストの妥当性を決定する重要な要因である。

開発者アクションエレメント:

ATE_FUN.2.1D

開発者は、TSFをテストし、結果を証拠資料で提出しなければならない。

ATE_FUN.2.2D

開発者は、テスト証拠資料を提供しなければならない。

内容・提示エレメント:

ATE_FUN.2.1C

テスト証拠資料は、テスト計画、期待されるテスト結果、及び実際のテスト結果から構成されなければならない。

ATE_FUN.2.2C

テスト計画は、実行されるべきテストを識別し、各テストを実行するシナリオを記述しなければならない。これらのシナリオは、他のテストの結果への全ての順序依存性を含んでいなければならない。

ATE_FUN.2.3C

期待されるテスト結果は、テストの実行が成功したときの予期される出力を示さなければならない。

ATE_FUN.2.4C

実際のテスト結果は、期待されたテスト結果と一貫していなければならない。

ATE_FUN.2.5C

テスト証拠資料は、テスト手順の順序依存性の分析を含まなければならない。

評価者アクションエレメント:

ATE_FUN.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

13.5 独立テスト(ATE_IND)

13.5.1 目的

このファミリの目的は、評価者が開発者テストを検証し追加のテストを実行することで、ATE_FUN、ATE_COV、及びATE_DPTファミリで達成される保証に基づいている。

13.5.2 コンポーネントのレベル付け

レベル付けは、開発者テスト証拠資料の量、テスト支援、及び評価者テストの量に基づいている。

13.5.3 適用上の注釈

このファミリは、TSF独立機能テストの程度を扱う。独立機能テストは、全体、又は一部において、開発者機能テストを繰り返す形式、又は開発者のテストの範囲又は深さを拡張する形式でも良い。これらのアクティビティは補完的であり、テスト結果の可用性とカバレージ、及びTSFの機能の複雑性を考慮して、TOEごとに適切な組み合わせが計画されなければならない。

開発者テストのサンプリングは、開発者が計画したTSFに対するテスト計画を実行し、結果を正しく記録していることの確証を提供することを意図している。選択されるべきサンプルの量は、開発者による機能テスト結果の詳細さと品質によって影響される。評価者は、また追加テストを考え出す範囲と、これらの2つの領域での労力から得られる相対的利益を考察する必要がある。全ての開発者テストを繰り返すことは、可能であり、望ましい場合もあるが、それ以外の場合では非常に困難で生産性が低いことが認識されている。したがって、このファミリの最上位のコンポーネントは注意して使用すべきである。サンプリングは、カバレージ(ATE_COV)と深さ(ATE_DPT)の両方の要件を満たすために提供されるテスト結果を含む、利用可能なテスト結果全体から行われる。

評価に含まれるTOEの異なる構成を考慮することもまた必要である。評価者は、提供された結果の有効性を評価し、それに応じて自らのテストを計画する必要がある。

テストに対するTOEの適合は、TOEへのアクセス、テストの実行に必要な支援の証拠資料、及び情報(あらゆるテストソフトウェア又はツールを含む)に基づく。そのような支援の必要性は別の保証ファミリへの依存によって述べられている。

加えて、テストに対するTOEの適合は、別の考えに基づいている。例えば、開発者から提供されたTOEのバージョンが最終バージョンではない可能性がある。

*インタフェース*という用語は、機能仕様及びTOE設計で記述されているインタフェースと、実装表現で識別される呼び出しを通じて渡されるパラメタを指している。使用される一連のインタフェースは、カバレージ(ATE_COV)及び深さ(ATE_DPT)コンポーネントを通じて選択される。

インタフェースのサブセットに対する参照は、実施する評価の目的に一致する適切なテストのセットを、評価者が設計できるようにすることを意図する。

13.5.4 ATE_IND.1 独立テスト - 適合

依存性： ADV_FSP.1 基本機能仕様

AGD_OPE.1 利用者操作ガイダンス

AGD_PRE.1 準備手続き

目的

このコンポーネントの目的は、TOEがその設計表現とガイダンス文書に従って動作することを実証することである。

適用上の注釈

このコンポーネントは、開発者テスト結果の使用について述べていない。そのような結果が利用できない場合や開発者テストが確認なく承認されている場合に有効である。評価者は、機能仕様を含んでいるがそれには限定されないTOEの設計表現に従ってTOEが動作することを確認する目的で、テストを考案し、実施することを要求される。アプローチは、全ての可能なテストを実施するよりも、代表的なテストを通じて正常動作の確信を得ることである。この目的のために計画されるテスト範囲は手法の問題であり、特定のTOEの背景と他の評価アクティビティとのバランスを考慮する必要がある。

開発者アクションエレメント:

ATE_IND.1.1D

開発者は、テストのためのTOEを提供しなければならない。

内容・提示エレメント:

ATE_IND.1.1C

TOEは、テストに適していなければならない。

評価者アクションエレメント:

ATE_IND.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ATE_IND.1.2E

評価者は、TSFが仕様どおりに動作することを確認するために、TSFのサブセットを*テストしなければならない*。

13.5.5 ATE_IND.2 独立テスト - サンプル

依存性：
- ADV_FSP.2 セキュリティ実施機能仕様
- AGD_OPE.1 利用者操作ガイダンス
- AGD_PRE.1 準備手続き
- ATE_COV.1 カバレージの証拠
- ATE_FUN.1 機能テスト

目的

このコンポーネントの目的は、TOEがその設計表現とガイダンス文書に従って動作することを実証することである。評価者テストは、開発者が、機能仕様内の一部のインタフェースについて何らかのテストを実行したことを確認する。

適用上の注釈

意図するところは、開発者テストの効果的な再現に必要な資料を、開発者が評価者に提供すべきことである。これには、マシンが読み取ることのできるテスト証拠資料やテストプログラムなどが含まれる。

このコンポーネントは、テストの計画を補うために、評価者が開発者からの利用可能なテスト結果を入手する要件を含んでいる。評価者は、得られた結果に対する確信を得るために、開発者テストのサンプルを繰り返す。確信が得られたら、評価者は開発者テストに基づいて、別の方法でTOEを使用する追加テストを実施する。正当性が確認された開発者テスト結果の基盤を使うことにより、評価者は単に開発者自身の労力や与えられた固定レベルの資源を使うことで可能となる以上に、より広い範囲の条件で、TOEが正常に動作することの確信が得られる。開発者がTOEのテストを完了しているとの確信を得ることで、評価者は、また証拠資料の調査や専門家の知識で特別に関心がある領域のテストに適切に集中するより多くの自由が得られる。

開発者アクションエレメント:

ATE_IND.2.1D

開発者は、テストのためのTOEを提供しなければならない。

内容・提示エレメント:

ATE_IND.2.1C

TOEは、テストに適していなければならない。

ATE_IND.2.2C

開発者は、TSFの開発者機能テストで使用されたものと同等の一連の資源を提供しなければならない。

評価者アクションエレメント:

ATE_IND.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ATE_IND.2.2E

評価者は、開発者テスト結果を検証するために、テスト証拠資料内のテストのサンプルを*実行しなければならない*。

ATE_IND.2.3E

評価者は、TSFが仕様どおりに動作することを確認するために、TSFのサブセットを*テストしなければならない*。

13.5.6 ATE_IND.3 独立テスト - 完全

依存性：
- ADV_FSP.4 完全な機能仕様
- AGD_OPE.1 利用者操作ガイダンス
- AGD_PRE.1 準備手続き
- ATE_COV.1 カバレージの証拠
- ATE_FUN.1 機能テスト

目的

このコンポーネントの目的は、TOEがその設計表現とガイダンス文書に従って動作することを実証することである。評価者テストは、開発者テストを全て繰り返すことを含む。

適用上の注釈

意図するところは、開発者テストの効果的な再現に必要な資料を、開発者が評価者に提供すべきことである。これには、マシンが読み取ることのできるテスト証拠資料やテストプログラムなどが含まれる。

このコンポーネントでは、評価者はテスト計画の一部として、開発者テストの全てを繰り返さなければならない。前のコンポーネントと同様に、評価者はまた、開発者が行ったのとは異なる方法で、TSFを実行させることを目的とするテストを実施する。開発者テストが徹底的に行われている場合には、これを行う余地は殆ど残っていないであろう。

開発者アクションエレメント:

ATE_IND.3.1D

開発者は、テストのためのTOEを提供しなければならない。

内容・提示エレメント:

ATE_IND.3.1C

TOEは、テストに適していなければならない。

ATE_IND.3.2C

開発者は、TSFの開発者機能テストで使用されたものと同等の一連の資源を提供しなければならない。

評価者アクションエレメント:

ATE_IND.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ATE_IND.3.2E

評価者は、開発者テスト結果を検証するために、テスト証拠資料内の**全ての**テストを*実行しなければならない*。

ATE_IND.3.3E

評価者は、TSF**全体**が仕様どおりに動作することを確認するために、TSFを*テストしなければならない*。

13.6 コンポジット機能テスト（ATE_COMP）

13.6.1 目的

このファミリの目的は、コンポジット製品が全体として、そのコンポジット製品STの機能要件を満たすのに必要な特性を示しているかどうかを決定することである。

13.6.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

13.6.3 適用上の注釈

コンポジット製品のテストは、コンポーネントを個別にテストする方法と、統合された製品をテストする方法で行うことができる。個別テストとは、基本コンポーネントと依存コンポーネントが互いに独立してテストされることを意味する。基本コンポーネントの多くのテストは、その達成された評価の範囲内で実施されている可能性がある。依存コンポーネントは、仮想マシンを表すシミュレータやエミュレータ上でテストされることがある。

統合テストとは、コンポジット製品をそのままテストすることであり、依存コンポーネントは関連する基本コンポーネントと一緒に実行される。

依存コンポーネントの機能テストの中には、このテストの有効性がコンポジット製品のインタフェースを使用して確認できない場合があるため、基本コンポーネントへの組み込み/統合の前に、エミュレータ上でのみ実施できるものがある。それでも、コンポジット製品の機能テストは、コンポジット製品のセキュリティ機能の記述に従い、関連するATE保証クラスが要求する標準的な手法を使用して、コンポジット製品のサンプルでも実行しなければならない。ここでは、追加の開発者のアクションは必要ない。

基本コンポーネントの機能テストの量、カバレッジ、深さは基本コンポーネント評価で既に検証されているため、コンポジット評価でこれらのタスクを再実施する必要はない。なお、基本コンポーネントの機能テストについては、*コンポジット評価用ETR*には記載されないので注意。

一部のSFRの実装のふるまいは、依存コンポーネントだけでなく基本コンポーネントの特性にも依存しうる(例えば、サイドチャネル攻撃に耐えるコンポジット製品の対策の正しさ、物理的攻撃に対する耐タンパ性の実装の正しさなど)。この場合、SFRの実装は、シミュレータやエミュレータではなく、最終的なコンポジット製品でテストしなければならない。

このファミリは、コンポジット製品全体のテストにのみ焦点を当て、ATE保証クラスでカバーされている一般的なテスト手法内の部分的な取り組みを示すにすぎない。これらの統合テストは、ATEクラスの標準保証ファミリの手法を適用することにより、特定し実行しなければならない。

コンポジット製品評価スポンサーは、コンポジット製品評価者が以下のものを入手できるように保証しなければならない。

— テストに適したコンポジット製品のサンプル

13.6.4 ATE_COMP.1 コンポジット製品の機能テスト

依存性：なし

開発者アクションエレメント:

ATE_COMP.1.1D

開発者は、選択された保証パッケージが要求するテストのセットを提供しなければならない。

ATE_COMP.1.2D

開発者は、テストのためのコンポジット製品を提供しなければならない。

内容・提示エレメント:

ATE_COMP.1.1C

***統合*テストの仕様と証拠資料の内容及び提示は、保証ファミリATE_FUN及びATE_COVの標準[10]要件に対応するものでなければならない。**

ATE_COMP.1.2C

提供されるコンポジット製品は、テストに適していなければならない。

---

[10] つまり、CEM で定義されたとおりの。

評価者アクションエレメント:

ATE_COMP.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

14 AVAクラス: 脆弱性評定

14.1 一般

AVA: 脆弱性評定クラスは、TOEの開発又は運用で生じる悪用可能な脆弱性の可能性を扱う。

図12は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。[xvi]

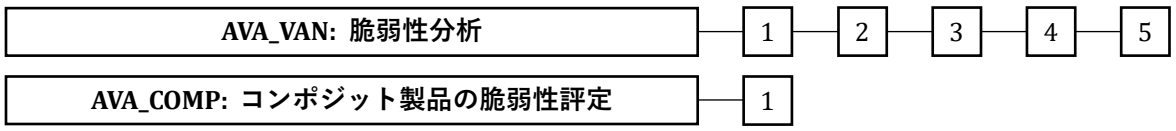


図 12 — AVA: 脆弱性評定クラスのコンポーネント構成

14.2 適用上の注釈

一般的に、脆弱性評定アクティビティは、TOEの開発及び運用における様々な脆弱性をカバーする。開発上の脆弱性は、TOE又はTOEが存在する製品の開発中に入り込むいくつか特性(TSFの改ざん、直接攻撃又は監視によりTSF自己保護が破られる、TSFの監視又は直接攻撃によりTSFドメイン分離が破られる、あるいはTSFの回避(バイパス)により非バイパス性が破られるなど)を悪用する。TOEの運用環境のITシステムへの明示的な依存関係も考慮しなければならない。運用上の脆弱性は、TOE SFRを侵害する非技術的な対抗策(誤使用や不正な構成など)における弱点を悪用する。誤使用は、セキュアでないにもかかわらずTOEの管理者又は利用者が合理的にセキュアであると判断した方法で、TOEが構成又は使用され得るかどうかを調査する。

開発上の脆弱性の評定は、保証ファミリAVA_VANによってカバーされる。基本的に、全ての開発上の脆弱性はAVA_VANの範囲で考慮することができる。これは、このファミリがある種の攻撃シナリオに特定されない幅広い評定方法を適用できるという事実に基づいている。これらの不特定の評定方法には、隠れチャネルが考慮されるTSF(チャネル容量の見積もりは、実際のテスト測定と同様、非形式的な工学的な測定によってなされる)、あるいは直接攻撃の形式で十分な資源を利用することで打開することができるTSF(これらのTSFの基になっている技術的な概念は、確率的又は順列的メカニズムに基づいている。つまり、それらを破るために必要なセキュリティ上のふるまいや労力の評価付けは、定量的又は統計的分析結果を用いて行うことができる)に対応するような評定方法などが含まれる。

TOEの1人の利用者がTOEの別の利用者に関連付けられているアクティビティを観察するのを防止する、又は情報フローを使用して不正なデータ信号を得ることができないことを保証するというセキュリティ対策方針がSTで特定されている場合、脆弱性分析の実施中に隠れチャネル分析を考慮するべきである。これは、観察不能性(FPR_UNO)及びアクセス制御方針(FDP_ACC)及び/又は情報フロー方針(アクセス制御方針(FDP_IFC)によって特定されるマルチレベルアクセス制御方針をSTに含めることで反映されることが多い。

14.3 脆弱性分析(AVA_VAN)

14.3.1 目的

脆弱性分析とは、TOEの開発及び予期される運用の評価を通して、又は他の方法(例えば、欠陥仮説法や、基礎となるセキュリティメカニズムのセキュリティのふるまいを定量的又は統計的に分析する方法)によって識別された潜在的脆弱性が、攻撃者によるSFRの侵害を許すかどうかを決定するための評定のことである。

脆弱性分析は、データと機能性に許可されないアクセスを許す、TSFを妨害又は変更できることを許す、又は他の利用者の許可された能力を妨害することができる、といった欠陥を攻撃者が見つけられるような脅威を扱う。

マルチ保証評価の場合、脆弱性分析はTOE全体だけでなく、定義された**サブTSF**も評価しなければならない。

14.3.2 コンポーネントのレベル付け

レベル付けは、評価者による脆弱性分析の厳格さ、及び攻撃者が潜在的脆弱性を見つけて悪用するために必要とする攻撃能力のレベルに基づいている。

14.3.3 AVA_VAN.1 脆弱性調査

依存性： ADV_FSP.1 基本機能仕様

AGD_OPE.1 利用者操作ガイダンス

AGD_PRE.1 準備手続き

目的

公知の情報の脆弱性調査は、攻撃者が容易に発見する可能性がある潜在的脆弱性を確認するために評価者が実行する。

評価者は、侵入テストを実行して、TOEの運用環境で潜在的脆弱性を悪用できないことを確認する。侵入テストは、評価者が、基本的な攻撃能力を想定して実行する。

開発者アクションエレメント:

AVA_VAN.1.1D

開発者は、テストのための*TOE*を提供しなければならない。

内容・提示エレメント:

AVA_VAN.1.1C

TOEは、テストに適していなければならない。

評価者アクションエレメント:

AVA_VAN.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

AVA_VAN.1.2E

評価者は、TOEの潜在的脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

AVA_VAN.1.3E

評価者は、基本的な攻撃能力を持つ攻撃者からの攻撃にTOEが耐えられることを決定するために、識別された潜在的脆弱性に基づいて侵入テストを*実施しなければならない*。

14.3.4 AVA_VAN.2 脆弱性分析

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_FSP.2 セキュリティ実施機能仕様

ADV_TDS.1 基本設計

AGD_OPE.1 利用者操作ガイダンス

AGD_PRE.1 準備手続き

目的

脆弱性分析は、評価者が潜在的脆弱性の存在を確認するために実行する。

評価者は、侵入テストを実行して、TOEの運用環境で潜在的脆弱性を悪用できないことを確認する。侵入テストは、評価者が、基本的な攻撃能力を想定して実行する。

開発者アクションエレメント:

AVA_VAN.2.1D

開発者は、テストのためのTOEを提供しなければならない。

AVA_VAN.2.2D

開発者は、TOE及びTOEの配付物に含まれるサードパーティコンポーネントのリストを提供しなければならない。

内容・提示エレメント:

AVA_VAN.2.1C

TOEは、テストに適していなければならない。

AVA_VAN.2.2C

サードパーティコンポーネントのリストには、サードパーティから提供されたコンポーネントで、TOEの一部又はTOEの配付物の一部であるものを含まなければならない。

評価者アクションエレメント:

AVA_VAN.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

AVA_VAN.2.2E

評価者は、TOE、**サードパーティコンポーネントリストのコンポーネント、及び環境におけるTOEが依存する特定のIT製品**の潜在的脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

AVA_VAN.2.3E

評価者は、TOEの潜在的脆弱性を識別するために、ガイダンス証拠資料、機能仕様、TOE設計、及びセキュリティアーキテクチャ記述を使用して、TOEの独立脆弱性分析を*実行しなければならない*。

AVA_VAN.2.4E

評価者は、基本的な攻撃能力を持つ攻撃者からの攻撃にTOEが耐えられることを決定するために、識別された潜在的脆弱性に基づいて侵入テストを*実施しなければならない*。

14.3.5 AVA_VAN.3 焦点を置いた脆弱性分析

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_FSP.4 完全な機能仕様

ADV_TDS.3 基本モジュール設計

ADV_IMP.1 TSFの実装表現

AGD_OPE.1 利用者操作ガイダンス

AGD_PRE.1 準備手続き

ATE_DPT.1テスト: 基本設計

目的

脆弱性分析は、評価者が潜在的脆弱性の存在を確認するために実行する。

評価者は、侵入テストを実行して、TOEの運用環境で潜在的脆弱性を悪用できないことを確認する。侵入テストは、評価者が、強化基本的な攻撃能力を想定して実行する。

開発者アクションエレメント:

AVA_VAN.3.1D

開発者は、テストのためのTOEを提供しなければならない。

AVA_VAN.3.2D

開発者は、TOE及びTOEの配付物に含まれるサードパーティコンポーネントのリストを提供しなければならない。

内容・提示エレメント:

AVA_VAN.3.1C

TOEは、テストに適していなければならない。

AVA_VAN.3.2C

サードパーティコンポーネントのリストには、サードパーティから提供されたコンポーネントで、TOEの一部又はTOEの配付物の一部であるものを含まなければならない。

評価者アクションエレメント:

AVA_VAN.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

AVA_VAN.3.2E

評価者は、TOE、サードパーティコンポーネントリストのコンポーネント、及び環境におけるTOEが依存する特定のIT製品の潜在的な脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

AVA_VAN.3.3E

評価者は、TOEの潜在的脆弱性を識別するために、ガイダンス証拠資料、機能仕様、TOE設計、セキュリティアーキテクチャ記述、**及び実装表現**を使用して、TOEの**焦点を置いた**独立脆弱性分析を*実行しなければならない*。

AVA_VAN.3.4E

評価者は、**強化基本的な**攻撃能力を持つ攻撃者からの攻撃にTOEが耐えられることを決定するために、識別された潜在的脆弱性に基づいて侵入テストを*実施しなければならない*。

14.3.6 AVA_VAN.4 系統的脆弱性分析

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_FSP.4 完全な機能仕様

ADV_TDS.3 基本モジュール設計

ADV_IMP.1 TSFの実装表現

AGD_OPE.1 利用者操作ガイダンス

AGD_PRE.1 準備手続き

ATE_DPT.1テスト: 基本設計

目的

系統的脆弱性分析は、評価者が潜在的脆弱性の存在を確認するために実行する。

評価者は、侵入テストを実行して、TOEの運用環境で潜在的脆弱性を悪用できないことを確認する。侵入テストは、評価者が、中程度の攻撃能力を想定して実行する。

開発者アクションエレメント:

AVA_VAN.4.1D

開発者は、テストのためのTOEを提供しなければならない。

AVA_VAN.4.2D

開発者は、TOE及びTOEの配付物に含まれるサードパーティコンポーネントのリストを提供しなければならない。

内容・提示エレメント:

AVA_VAN.4.1C

TOEは、テストに適していなければならない。

AVA_VAN.4.2C

サードパーティコンポーネントのリストには、サードパーティから提供されたコンポーネントで、TOEの一部又はTOEの配付物の一部であるものを含まなければならない。

評価者アクションエレメント:

AVA_VAN.4.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

AVA_VAN.4.2E

評価者は、TOE、サードパーティコンポーネントリストのコンポーネント、及び環境におけるTOEが依存する特定のIT製品の潜在的な脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

AVA_VAN.4.3E

評価者は、TOEの潜在的脆弱性を識別するために、ガイダンス証拠資料、機能仕様、TOE設計、セキュリティアーキテクチャ記述、及び実装表現を使用して、TOEの独立した、**系統的**脆弱性分析を*実行しなければならない*。

AVA_VAN.4.4E

評価者は、**中程度の**攻撃能力を持つ攻撃者からの攻撃にTOEが耐えられることを決定するために、識別された潜在的脆弱性に基づいて侵入テストを*実施しなければならない*。

14.3.7 AVA_VAN.5 高度な系統的脆弱性分析

依存性： ADV_ARC.1 セキュリティアーキテクチャ記述

ADV_FSP.4 完全な機能仕様

ADV_TDS.3 基本モジュール設計

ADV_IMP.1 TSFの実装表現

AGD_OPE.1 利用者操作ガイダンス

AGD_PRE.1 準備手続き

ATE_DPT.1テスト: 基本設計

目的

系統的脆弱性分析は、評価者が潜在的脆弱性の存在を確認するために実行する。

評価者は、侵入テストを実行して、TOEの運用環境で潜在的脆弱性を悪用できないことを確認する。侵入テストは、評価者が、高い攻撃能力を想定して実行する。

開発者アクションエレメント:

AVA_VAN.5.1D

開発者は、テストのためのTOEを提供しなければならない。

AVA_VAN.5.2D

開発者は、TOE及びTOEの配付物に含まれるサードパーティコンポーネントのリストを提供しなければならない。

内容・提示エレメント:

AVA_VAN.5.1C

TOEは、テストに適していなければならない。

AVA_VAN.5.2C

サードパーティコンポーネントのリストには、サードパーティから提供されたコンポーネントで、TOEの一部又はTOEの配付物の一部であるものを含まなければならない。

評価者アクションエレメント:

AVA_VAN.5.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

AVA_VAN.5.2E

評価者は、TOE、サードパーティコンポーネントリストのコンポーネント、及び環境におけるTOEが依存する特定のIT製品の潜在的な脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

AVA_VAN.5.3E

評価者は、TOEの潜在的脆弱性を識別するために、ガイダンス証拠資料、機能仕様、TOE設計、セキュリティアーキテクチャ記述、及び実装表現を使用して、TOEの独立した、系統的脆弱性分析を*実行しなければならない*。

AVA_VAN.5.4E

評価者は、**高い**攻撃能力を持つ攻撃者からの攻撃にTOEが耐えられることを決定するために、識別された潜在的脆弱性に基づいて侵入テストを*実施しなければならない*。

14.4 コンポジット脆弱性評定(AVA_COMP)

14.4.1 目的

このファミリの目的は、意図した環境において、コンポジット製品全体の欠陥や弱点の悪用可能性を決定することである。

14.4.2 コンポーネントのレベル付け

このファミリは、ただ1つのコンポーネントからなる。

14.4.3 適用上の注釈

このファミリは、コンポジット製品*全体*の脆弱性評定にのみ焦点を当て、AVAクラスの標準の[11]保証ファミリあるAVA_VANでカバーされている一般的な手法内の*部分的な取り組み*を示すにすぎない。

コンポジット製品評価者は、特に基本コンポーネントの評価結果を用いて、コンポジット製品の脆弱性分析を実行しなければならない。この脆弱性分析は、侵入テストにより確認しなければならない。

コンポジット製品評価者は、基本コンポーネントに組み込まれた/インストールされた依存コンポーネントの機密保護が、依存コンポーネント開発者がALC_DVSに対して主張する機密性レベルと一貫したものであることをチェックしなければならない。

特殊なケースでは、証拠資料のみが利用可能な場合、脆弱性分析と攻撃の定義が困難で、かなりの時間を要し、広範囲の事前テストを必要とすることがある。また、基本コンポーネントは、基本コンポーネント開発者や基本コンポーネント評価者が予見していなかった方法で使用されたり、依存コンポーネント開発者が基本コンポーネントで提供される規定に従っていなかったりする可能性がある。以下のような場合、コンポジット製品の脆弱性分析を短縮できる、様々な可能性が存在する。例えば、コンポジット製品の評価者は基本コンポーネントの評価者に相談し、基本コンポーネントの評価で得た経験を活用することができる。あるいは、コンポジット製品評価者が、自らの裁量で、テスト用の依存コンポーネントをロードした基本コンポーネントの特定のテストサンプルを使用することで、依存コンポーネントと基本コンポーネントの脆弱性を分離するアプローチもある。ここでは、基本コンポーネント固有の対抗策を解除することなく、対抗策を施さないテスト用の依存コンポーネントを使用することを意図している。

*コンポジット評価用ETR*に記載されたコンポジット製品の基本コンポーネントの脆弱性評定結果は、次の条件下で再利用することができる。最新のものであり、コンポジット製品の正確性に関するアクティビティ(ASE_COMP.1、ALC_COMP.1、ADV_COMP.1及びATE_COMP.1)が全て合格(PASS)という評決で確定していること。

[11] つまり、CEM で定義されたとおりの。

基本コンポーネントと依存コンポーネントの統合により、新たな品質が発生し、*コンポジット評価用ETR*に記載されていない、新たな基本コンポーネントの脆弱性が生じる場合がある。このような場合、コンポジット製品評価監督機関は、基本コンポーネントの再評定又は再評価を、新たな脆弱性の問題に焦点を当てて要求することができる。

コンポジット製品評価スポンサーは、以下のものがコンポジット製品評価者のために利用できるように保証しなければならない。

— 基本コンポーネント関連の利用者ガイダンス、

— 基本コンポーネント評価者が準備した、基本コンポーネントに関連する*コンポジット評価用のETR*、

— 基本コンポーネント評価監督機関の報告書。

14.4.4 AVA_COMP.1 コンポジット製品の脆弱性評定

依存性：なし

開発者アクションエレメント:

AVA_COMP.1.1D

開発者は、侵入テストのためのコンポジット製品を提供しなければならない。

内容・提示エレメント:

AVA_COMP.1.1C

提供されたコンポジット製品は、全体としてテストに適していなければならない。

評価者アクションエレメント:

AVA_COMP.1.1E

評価者は、コンポジット製品のセキュリティターゲットに関連する脆弱性が悪用可能でないことを保証するために、評価者自身の脆弱性分析に基づき、コンポジット製品*全体*に対する侵入テストを*実施しなければならない*。

15 ACOクラス: 統合

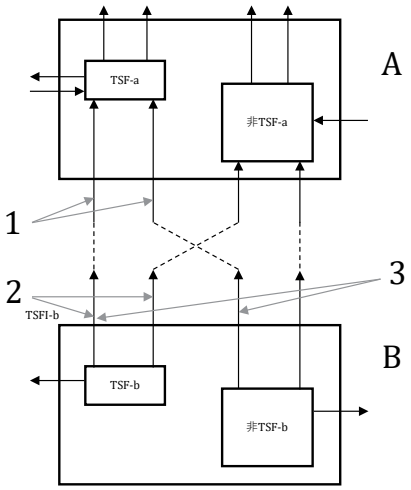
15.1 一般

ACO: 統合クラスは、5つのファミリを含む。これらのファミリでは、統合TOEが、すでに評価されたソフトウェア、ファームウェア、又はハードウェアコンポーネントが提供するセキュリティ機能性に依存する場合にセキュアに動作するという信頼を提供するために策定された保証要件を特定する。

統合では、CCセキュリティ保証要件パッケージに対して正常に評価された複数のITエンティティ(基本コンポーネント及び依存コンポーネント、附属書Bを参照)を、そのいずれのエンティティもそれ以上開発することなく、結合して使用できるようにする。追加のITエンティティ(以前コンポーネント評価の対象になっていなかったエンティティ)の開発は含まれない。統合TOEは、環境に関する対策方針を満たす特定の環境事例全てに設置及び統合できる新しい製品を形成する。

このアプローチは、コンポーネントを評価するための代替アプローチとはならない。ACO下での統合は、統合TOEのインテグレータに1つの方法を提供する。その方法は、統合TSFを再評価することなく、評価が完了した複数のコンポーネントの組み合わせであるTOEで信頼を得るために、CCで特定された他の保証レベルの代替として使用できる。統合TOEのインテグレータは、そのように分類された基本コンポーネント又は依存コンポーネントの開発者に関連して、ACOクラス全体で「開発者」と呼ばれる。

CCパート5で定義されているCAPは、統合TOEの保証尺度を提供する。他の保証パッケージに対して評価されたコンポーネントを組み合わせ、結果として得られる統合TOEにおいて同等の保証を得るためには、全てのSARを統合TOEに適用する必要があるため、例えばEALのような他の保証パッケージに加えてこの保証尺度が必要である。再使用はコンポーネントTOEの評価結果で構成できるが、B.3に記述されているように、統合TOEではコンポーネントの追加側面を考慮しなければならないことがよくある。統合TOEの評価アクティビティには様々な当事者が関わるため、通常は適切なEALを適用するために、コンポーネントのこれらの追加側面に関して必要な全ての証拠を得ることは不可能である。このため、評価されたコンポーネントを組み合わせ、有意な結果を得るための問題に対処するためにCAPが定義されている。これについては、附属書Bで詳しく説明する。



キー

A	依存コンポーネント-a
B	基本コンポーネント-b
1	ACO_REL (コンポーネント-a)
2	ADV_FSP (コンポーネント-b)
3	ACO_DEV (コンポーネント-b)

図 13 — ACO ファミリ間の関係とコンポーネント間の相互作用

統合TOEでは、通常1つのコンポーネントは別のコンポーネントが提供したサービスに依存する。サービスを要求するコンポーネントは依存コンポーネントと呼ばれ、サービスを提供するコンポーネントは基本コンポーネントと呼ばれる。この相互作用と区別については、附属書Bで詳しく説明する。これは、依存コンポーネントの開発者が統合TOE評価を何らかの方法(開発者、スポンサーとして、又は単に協調して依存コンポーネント評価から必要な評価証拠を提供する)でサポートしている場合を想定している。CAP保証パッケージに含まれるACOコンポーネントは、コンポーネントTOE評価の要件追加として使用すべきではない。要件追加として使用した場合、コンポーネントに対する意味のある保証は提供されない。

ACOクラス内のファミリは、コンポーネントTOE評価内のADVクラス、ATEクラス及びAVAクラスと同様の方法で相互作用するため、これらのクラスの要件の仕様を適宜利用する。ただし、統合TOE評価に固有な要素がいくつか存在する。コンポーネント同士がどのように相互作用するかを判別し、コンポーネントの評価からの全ての不足を識別するために、依存コンポーネントの下層の基本コンポーネントに対する依存性が識別される(ACO_REL)。この基本コンポーネントへの依存は、依存コンポーネントSFRの支援をするサービスに対して依存コンポーネントがコールを行う際に使用されるインタフェースの観点から特定される。それらのサービス要求に応答して基本コンポーネントが提供するインタフェース、及び上位レベルにおける支援のふるまいは、ACO_DEVで分析される。ACO_DEVファミリはADV_TDSファミリに基づいており、各コンポーネントの最も単純なレベルにおけるTSFは統合TOEのサブシステムとみなすことができ、各コンポーネントの追加の部分は追加のサブシステムとみなされる。したがって、コンポーネント間のインタフェースは、統合TOE評価内のサブシステム間の相互作用とみなされる。

ACO_DEVに対して提供されるインタフェース、及び支援のふるまいの記述は不完全である可能性が存在する。これは、ACO_CORの実施中に決定される。ACO_CORファミリは、ACO_REL及びACO_DEVの出力を取得して、コンポーネントが評価構成で使用されているかど

うかを判断し、仕様が不完全な個所を識別する。これは、統合TOEのテスト(ACO_CTT)及び脆弱性分析(ACO_VUL)アクティビティへの入力として識別される。

統合TOEのテストは、統合TOEが、統合TOE SFRによって決定された期待されるふるまいを示していることを判断するために実行され、より上位レベルでは、統合TOEのコンポーネント間のインタフェースの互換性を実証する。

統合TOEの脆弱性分析は、コンポーネント評価の脆弱性分析の出力を利用する。統合TOEの脆弱性分析は、コンポーネント評価の全ての残存脆弱性を考慮して、残存脆弱性が統合TOEに適用可能でないことを決定する。コンポーネントに関連する公知の情報の探索もまた、それぞれの評価の完了以降にコンポーネントで報告された全ての問題を識別するために実行される。

ACOファミリ間の相互作用を下記の図14に示す。この図では、1つのファミリ内で得られる証拠と理解が次のアクティビティに提供される場所は実線の矢印で示され、破線の矢印は、上記のとおり、アクティビティが明示的に統合TOE SFRにまでさかのぼるところを識別している。

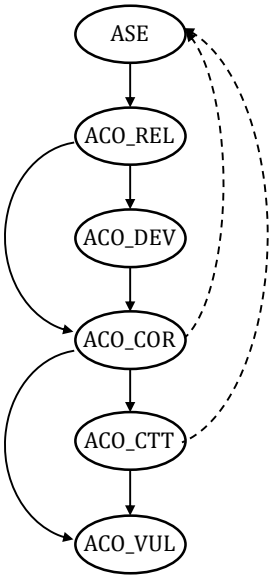


図 14 — ACO ファミリ間の関係

統合TOEにおける定義と相互作用の詳細は、附属書Bに記載されている。

図15は、このクラスのファミリと、各ファミリのコンポーネントの階層を示す。

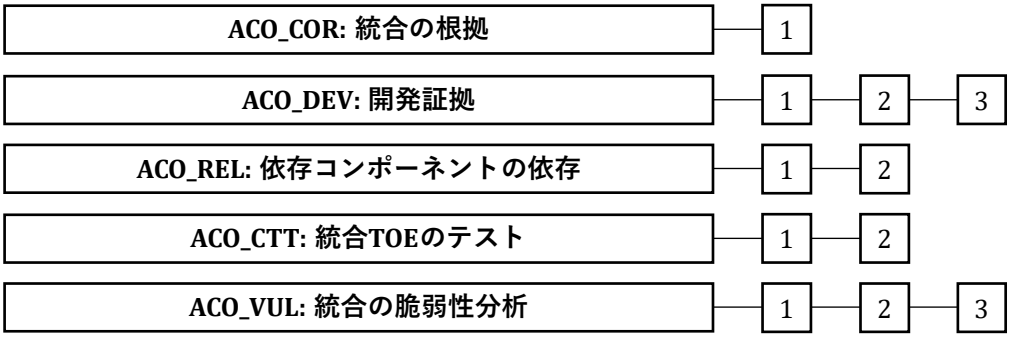


図 15 — ACO: 統合クラスのコンポーネント構成

15.2 統合の根拠(ACO_COR)

15.2.1 目的

このファミリは、統合で使用する適切なレベルの保証を基本コンポーネントが提供できることを実証するための要件を扱う。

15.2.2 コンポーネントのレベル付け

このファミリに含まれるコンポーネントは1つのみである。

15.2.3 ACO_COR.1 統合の根拠

依存性： ACO_DEV.1 機能記述

ALC_CMC.1 TOEのラベル付け

ACO_REL.1 基本依存情報

開発者アクションエレメント:

ACO_COR.1.1D

開発者は、基本コンポーネントの統合の根拠を提供しなければならない。

内容・提示エレメント:

ACO_COR.1.1C

統合の根拠は、基本コンポーネントが、依存コンポーネントのTSFを支援する要求に従い構成された場合、依存コンポーネントのものと少なくとも同じ保証のレベルが、基本コンポーネントの支援機能性に対して得られることを実証しなければならない。

評価者アクションエレメント:

ACO_COR.1.1E

評価者は、情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

15.3 開発証拠(ACO_DEV)

15.3.1 目的

このファミリは、基本コンポーネントの仕様に対する要件を、詳細レベルを上げながら設定する。このような情報は、(依存情報で識別されている)依存コンポーネントの要件を支援するために適切なセキュリティ機能性が提供されているという確信を得るために必要である。

15.3.2 コンポーネントのレベル付け

コンポーネントは、提供されているインタフェースに関する詳細の量、及びそれらの実装状況に基づいて、レベル付けされている。

15.3.3 適用上の注釈

基本コンポーネントのTSFは、その統合での可能な適用における依存性に関する知識なしに定義されることが少なくない。この基本コンポーネントのTSFは、基本コンポーネントSFRを実施するために依存しなければならない基本コンポーネントの全ての部分を含むように定義される。これには、基本コンポーネントのSFRの実装に必要な基本コンポーネントの全ての部分が含まれる。

この基本コンポーネントの機能仕様は、TSFの操作の呼び出しを外部エンティティに許可するために基本コンポーネントが提供するインタフェースの観点からTSFIを記述する。これには、SFRを呼び出すTSFの操作との相互作用を可能にする人間の利用者とのインタフェース、及び外部ITエンティティにTSFへのコールを許可するインタフェースが含まれる。

機能仕様は、TSFがそのインタフェースで何を提供するか、及びTSFの機能性が呼び出される手段についての記述のみを提供する。したがって、機能仕様は、必ずしも外部エンティティと基本コンポーネントの間で利用可能な全てのインタフェースの完全なインタフェース仕様を提供するわけではない。TSFが運用環境に何を期待/要求するかは含まれない。依存コンポーネントTSFが基本コンポーネントに依存する内容の記述は、依存コンポーネントの依存(ACO_REL)で考慮され、開発情報の証拠は、特定されているインタフェースへの応答を提供する。

開発情報の証拠には、基本コンポーネントの仕様が含まれる。これは、基本コンポーネントの評価中に、ADV要件を満たすために使用される証拠の場合もあれば、基本コンポーネントの開発者又は統合TOEの開発者によって生成される別の形式の証拠の場合もある。基本コンポーネントのこの仕様は、開発証拠(ACO_DEV)の中で、依存コンポーネントの要件を支援するために適切なセキュリティ機能性が提供されているという確信を得るために使用される。この証拠に要求される詳細レベルは、統合TOEで要求される保証のレベルを反映して上昇する。これは、コンポーネントに対する保証パッケージの適用によって得られる確信の増加を広く反映することが期待される。評価者は、基本コンポーネントのこの記述が、依存コンポーネントに提供される依存情報と一致していることを決定する。

15.3.4 ACO_DEV.1 機能記述

依存性：ACO_REL.1 基本依存情報

目的

依存コンポーネントが依存する基本コンポーネントのインタフェースの記述が要求される。この記述は、依存情報での依存コンポーネントが依存するインタフェースの記述と一貫しているかどうかを決定するために検査される。

開発者アクションエレメント:

ACO_DEV.1.1D

開発者は、基本コンポーネントの開発情報を提供しなければならない。

内容・提示エレメント:

ACO_DEV.1.1C

開発情報は、統合TOEで使用される基本コンポーネントの各インタフェースの目的を記述しなければならない。

ACO_DEV.1.2C

開発情報は、依存コンポーネントのTSFを支援するために、基本コンポーネントと依存コンポーネントの、統合TOEで使用されるインタフェース間の対応を示さなければならない。

評価者アクションエレメント:

ACO_DEV.1.1E

評価者は、情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_DEV.1.2E

評価者は、提供されたインタフェース記述が、依存コンポーネントに提供される依存情報と一貫していることを*決定しなければならない*。

15.3.5 ACO_DEV.2 設計の基本証拠

依存性：ACO_REL.1 基本依存情報

目的

依存コンポーネントが依存する基本コンポーネントのインタフェースの記述が要求される。この記述は、依存情報での依存コンポーネントが依存するインタフェースの記述と一貫しているかどうかを決定するために検査される。

さらに、依存コンポーネントのTSFを支援する基本コンポーネントのセキュリティのふるまいが記述される。

開発者アクションエレメント:

ACO_DEV.2.1D

開発者は、基本コンポーネントの開発情報を提供しなければならない。

内容・提示エレメント:

ACO_DEV.2.1C

開発情報は、統合TOEで使用される基本コンポーネントの各インタフェースの目的**及び使用方法**を記述しなければならない。

ACO_DEV.2.2C

開発情報は、依存コンポーネントのSFRの実施を支援する、基本コンポーネントのふるまいの上位レベルの記述を提供しなければならない。

ACO_DEV.2.3C

開発情報は、依存コンポーネントのTSFを支援するために、基本コンポーネントと依存コンポーネントの、統合TOEで使用されるインタフェース間の対応を示さなければならない。

評価者アクションエレメント:

ACO_DEV.2.1E

評価者は、情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_DEV.2.2E

評価者は、提供されたインタフェース記述が、依存コンポーネントに提供される依存情報と一貫していることを*決定しなければならない*。

15.3.6 ACO_DEV.3 設計の詳細証拠

依存性：ACO_REL.2 依存情報

目的

依存コンポーネントが依存する基本コンポーネントのインタフェースの記述が要求される。この記述は、依存情報での依存コンポーネントが依存するインタフェースの記述と一貫しているかどうかを決定するために検査される。

基本コンポーネントのアーキテクチャのインタフェース記述は、そのインタフェースが基本コンポーネントのTSFの一部であるかどうかを評価者が決定できるようにするために提供される。

開発者アクションエレメント:

ACO_DEV.3.1D

開発者は、基本コンポーネントの開発情報を提供しなければならない。

内容・提示エレメント:

ACO_DEV.3.1C

開発情報は、統合TOEで使用される基本コンポーネントの各インタフェースの目的及び使用方法を記述しなければならない。

ACO_DEV.3.2C

開発情報は、統合TOEで使用される基本コンポーネントのインタフェースを提供する基本コンポーネントのサブシステムを識別しなければならない。

ACO_DEV.3.3C

開発情報は、依存コンポーネントのSFRの実施を**支援する**、基本コンポーネントの**サブシステムの**ふるまいの上位レベルの記述を提供しなければならない。

ACO_DEV.3.4C

開発情報は、インタフェースから基本コンポーネントのサブシステムへのマッピングを提供しなければならない。

ACO_DEV.3.5C

開発情報は、依存コンポーネントのTSFを支援するために、基本コンポーネントと依存コンポーネントの、統合TOEで使用されるインタフェース間の対応を示さなければならない。

評価者アクションエレメント:

ACO_DEV.3.1E

評価者は、情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_DEV.3.2E

評価者は、提供されたインタフェース記述が、依存コンポーネントに提供される依存情報と一貫していることを*決定しなければならない*。

15.4 依存コンポーネントの依存(ACO_REL)

15.4.1 目的

このファミリの目的は、基本コンポーネントに対する依存コンポーネントの依存を記述する証拠を提供することである。この情報は、コンポーネントを他の評価されたITコンポーネントに統合して統合TOEを形成する担当者、及び結果としての統合のセキュリティ特性に関する洞察を提供する担当者にとって役立つ。

このファミリは、インタフェースが個々のコンポーネントTOEのTSFIでなかったために、個々のコンポーネントの評価中に分析されなかった可能性のある、統合TOEの依存コンポーネントと基本コンポーネント間のインタフェースの記述を提供する。

15.4.2 コンポーネントのレベル付け

このファミリのコンポーネントは、基本コンポーネントに対する依存コンポーネントの依存の記述に示されている詳細の量に従ってレベル付けされている。

15.4.3 適用上の注釈

依存コンポーネントの依存(ACO_REL)ファミリでは、依存コンポーネントがそのセキュリティ機能性の操作を支援するために基本コンポーネントのサービスを信頼している状況での、コンポーネント間の相互作用が考慮される。基本コンポーネントのサービスはコンポーネント評価の中でセキュリティに関連性があるとみなされていなかったため、基本コンポーネントのそれらサービスに対するインタフェースは基本コンポーネント評価の中で考慮されていなかった可能性がある。これは、サービス特有の目的(例えば、タイプフォントの調整)が原因であるか、又は関連するCCパート2のSFRが基本コンポーネントのSTで要求されていない(例えば、FIA：識別認証SFRが主張されていない場合のログインインタフェース)ことが原因である。基本コンポーネントに対するこれらのインタフェースは、基本コンポーネントの評価で機能インタフェースとみなされることが多く、機能仕様で考慮されるセキュリティインタフェース(TSFI)に追加されている。

要約すると、機能仕様で記述されているTSFIには、外部エンティティがTSFに対して行うコールと、それらのコールに対する応答のみが含まれる。TSFによって行われるコールは、コンポーネントの評価では明示的に考慮されず、依存コンポーネントの依存(ACO_REL)を満たすために提供される依存情報で記述される。

15.4.4 ACO_REL.1 基本依存情報

依存性：なし

開発者アクションエレメント:

ACO_REL.1.1D

開発者は、依存コンポーネントの依存情報を提供しなければならない。

内容・提示エレメント:

ACO_REL.1.1C

依存情報は、依存コンポーネントTSFが依存する基本コンポーネントハードウェア、ファームウェア及び/又はソフトウェアの機能性を記述しなければならない。

ACO_REL.1.2C

依存情報は、依存コンポーネントTSFが基本コンポーネントからサービスを要求するために使用する全ての相互作用を記述しなければならない。

ACO_REL.1.3C

依存情報は、依存TSFが、基本コンポーネントによる干渉及び改ざんから自分自身をどのように保護するかを記述しなければならない。

評価者アクションエレメント:

ACO_REL.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

15.4.5 ACO_REL.2 依存情報

依存性：なし

開発者アクションエレメント:

ACO_REL.2.1D

開発者は、依存コンポーネントの依存情報を提供しなければならない。

内容・提示エレメント:

ACO_REL.2.1C

依存情報は、依存コンポーネントTSFが依存する基本コンポーネントハードウェア、ファームウェア及び/又はソフトウェアの機能性を記述しなければならない。

ACO_REL.2.2C

依存情報は、依存コンポーネントTSFが基本コンポーネントからサービスを要求するために使用する全ての相互作用を記述しなければならない。

ACO_REL.2.3C

依存情報は、使用されるインタフェース及びそれらのインタフェースからの戻り値の観点から各相互作用を記述しなければならない。

ACO_REL.2.4C

依存情報は、依存TSFが、基本コンポーネントによる干渉及び改ざんから自分自身をどのように保護するかを記述しなければならない。

評価者アクションエレメント:

ACO_REL.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

15.5 統合TOEのテスト(ACO_CTT)

15.5.1 目的

このファミリは、統合TOEで使用されるように、統合TOEのテスト及び基本コンポーネントのテストが実行されることを要求する。

15.5.2 コンポーネントのレベル付け

このファミリのコンポーネントは、インタフェーステストの厳格さ、及び統合TSFが依存情報及び統合TOE SFRに従って動作することを実証するテストの十分性の分析の厳格さに基づいて、レベル付けされている。

15.5.3 適用上の注釈

このファミリに関連するテストには、次の2つの異なる側面が存在する:

a) 基本コンポーネント及び依存コンポーネント間の、セキュリティ機能性の実施のために依存コンポーネントが依存するインタフェースの互換性を実証するためのテスト。

b) 統合TOEのSFRに従ってTOEが動作することを実証するための統合TOEのテスト。

依存コンポーネントの評価中に使用されるテスト構成に「プラットフォーム」としての基本コンポーネントの使用が含まれ、TSFがSFRに従って動作することをテストの分析が十分に実証する場合、開発者は統合TOEの機能性のテストをさらに実行する必要はない。ただし、基本コンポーネントが依存コンポーネントのテストで使用されなかった場合、又はどちらかのコンポーネントの構成が変更された場合、開発者は、統合TOEのテストを実行する。これが、SFRに従って統合TOEのTSFが動作していることを十分に実証していれば、依存コンポーネントの依存コンポーネント開発者テストを繰り返す形式でもよい。

開発者は、統合で使用される基本コンポーネントインタフェースをテストすることの証拠を提供しなければならない。基本コンポーネントのTSFIの操作は、基本コンポーネントの評価中にATE: テストのアクティビティの一部としてテストされている可能性がある。このため、依存コンポーネントが必要とする全てのセキュリティ機能性がTSFに含まれている状態で、適切なインタフェースが基本コンポーネント評価のテストサンプル内に含まれ、基本コンポーネントが評価構成に従って動作していることが、統合の根拠(ACO_COR)で決定された場合は、基本コンポーネントATE: テスト判定の再利用によって評価者アクションACO_CTT.1.1Eを満たすことができる。

これに該当しない場合は、統合に関連して使用され、評価構成に対する変更及び追加のセキュリティ機能によって影響を受ける基本コンポーネントインタフェースが、期待されたふるまいを示すことを保証するためにテストされる。テストの対象である期待されるふるまいは、依存情報(依存コンポーネントの依存(ACO_REL)の証拠)で記述される。

15.5.4 ACO_CTT.1 インタフェーステスト

依存性： ACO_REL.1 基本依存情報

ACO_DEV.1 機能記述

目的

このコンポーネントの目的は、依存コンポーネントが依存する基本コンポーネントの各インタフェースがテストされることを保証することである。

開発者アクションエレメント:

ACO_CTT.1.1D

開発者は、統合TOEのテスト証拠資料を提供しなければならない。

ACO_CTT.1.2D

開発者は、基本コンポーネントのインタフェーステスト証拠資料を提供しなければならない。

ACO_CTT.1.3D

開発者は、テストのために統合TOEを提供しなければならない。

ACO_CTT.1.4D

開発者は、基本コンポーネントの基本コンポーネント開発者機能テストで使用されたものと同等の一連の資源を提供しなければならない。

内容・提示エレメント:

ACO_CTT.1.1C

統合TOE及び基本コンポーネントインタフェーステスト証拠資料は、テスト計画、期待されるテスト結果、及び実際のテスト結果から構成されなければならない。

ACO クラス: 統合

ACO_CTT.1.2C

開発者が実行した統合TOEテストのテスト証拠資料は、TSFが仕様どおりにふるまうことを実証しなければならない。

ACO_CTT.1.3C

開発者が実行した基本コンポーネントインタフェーステストのテスト証拠資料は、依存コンポーネントが依存する基本コンポーネントインタフェースが仕様どおりにふるまうことを実証しなければならない。

ACO_CTT.1.4C

基本コンポーネントは、テストに適していなければならない。

評価者アクションエレメント:

ACO_CTT.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていること*を確認しなければならない*。

ACO_CTT.1.2E

評価者は、開発者テスト結果を検証するために、テスト証拠資料内のテストのサンプルを*実行しなければならない*。

ACO_CTT.1.3E

評価者は、統合TSFが仕様どおりに動作することを確認するために、統合TOEのTSFインタフェースのサブセットを*テストしなければならない*。

15.5.5 ACO_CTT.2 厳格なインタフェーステスト

依存性： ACO_REL.2 依存情報

ACO_DEV.2 設計の基本証拠

目的

このコンポーネントの目的は、依存コンポーネントが依存する基本コンポーネントの各インタフェースがテストされることを保証することである。

開発者アクションエレメント:

ACO_CTT.2.1D

開発者は、統合TOEのテスト証拠資料を提供しなければならない。

ACO_CTT.2.2D

開発者は、基本コンポーネントのインタフェーステスト証拠資料を提供しなければならない。

ACO_CTT.2.3D

開発者は、テストのために統合TOEを提供しなければならない。

ACO_CTT.2.4D

開発者は、基本コンポーネントの基本コンポーネント開発者機能テストで使用されたものと同等の一連の資源を提供しなければならない。

内容・提示エレメント:

ACO_CTT.2.1C

統合TOE及び基本コンポーネントインタフェーステスト証拠資料は、テスト計画、期待されるテスト結果、及び実際のテスト結果から構成されなければならない。

ACO_CTT.2.2C

開発者が実行した統合TOEのテストによるテスト証拠資料は、TSFが仕様どおりに動作し、**完全である**ことを実証しなければならない。

ACO_CTT.2.3C

開発者が実行した基本コンポーネントインタフェーステストのテスト証拠資料は、依存コンポーネントが依存する基本コンポーネントインタフェースが仕様どおりに動作し、**完全である**ことを実証しなければならない。

ACO_CTT.2.4C

基本コンポーネントは、テストに適していなければならない。

評価者アクションエレメント:

ACO_CTT.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_CTT.2.2E

評価者は、開発者テスト結果を検証するために、テスト証拠資料内のテストのサンプルを*実行しなければならない*。

ACO_CTT.2.3E

評価者は、統合TSFが仕様どおりに動作することを確認するために、統合TOEのTSFインタフェースのサブセットを*テストしなければならない*。

15.6 統合の脆弱性分析(ACO_VUL)

15.6.1 目的

このファミリは、公知に利用できる脆弱性情報、及び統合の結果として生じる可能性がある脆弱性の分析を要求する。

15.6.2 コンポーネントのレベル付け

このファミリのコンポーネントは、公知の脆弱性情報及び独立脆弱性分析の検査の厳格さに基づいて、レベル付けされている。

15.6.3 適用上の注釈

開発者は、コンポーネントの評価で報告された残存脆弱性の詳細を提供する。これらの詳細は、コンポーネント開発者又はコンポーネントの評価報告書から得ることができる。これらは、運用環境における統合TOEの評価者による脆弱性分析への入力として使用される。

統合TOEの運用環境は、コンポーネント運用環境の(各コンポーネントSTで特定された)前提条件及び目的が統合TOEで満たされることを保証するために検査される。コンポーネント及び統合TOE ST間の前提条件と目的の一貫性の初期分析は、統合TOEのASEアクティビティの実施中に実行される。ただし、この分析は、例えば、依存コンポーネントST内の環境で扱われた依存コンポーネントの前提条件が、統合の結果として再度生じない(つまり、基本コンポーネントが、統合TOE内の依存コンポーネントSTの前提条件を適切に扱っている)こと

を保証するために、ACO_REL、ACO_DEV及びACO_CORアクティビティ中に得た知識に基づいて再び使用される。

評価者による各コンポーネントの問題の探索は、コンポーネントの評価の完了以降に公知に報告された潜在的脆弱性を識別する。その後、潜在的脆弱性はテストのサブジェクトとなる。

統合TOEで使用される基本コンポーネントが認証以後の保証継続性アクティビティの対象となった場合、評価者は、統合TOE脆弱性分析アクティビティ中に基本コンポーネントに加えられた変更を考慮する。

15.6.4 ACO_VUL.1 統合の脆弱性レビュー

依存性：ACO_DEV.1 機能記述

開発者アクションエレメント:

ACO_VUL.1.1D

開発者は、テストのために統合TOEを提供しなければならない。

内容・提示エレメント:

ACO_VUL.1.1C

統合TOEは、テストに適していなければならない。

評価者アクションエレメント:

ACO_VUL.1.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_VUL.1.2E

評価者は、基本コンポーネント及び依存コンポーネントで識別された残存脆弱性が、その運用環境の統合TOEで悪用不能であることを決定する分析を*実行しなければならない*。

ACO_VUL.1.3E

評価者は、統合TOEの運用環境で基本コンポーネントと依存コンポーネントを使用することで生じる可能性がある脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

ACO_VUL.1.4E

評価者は、基本的な攻撃能力を持つ攻撃者からの攻撃に統合TOEが耐えられることを実証するために、識別された脆弱性に基づいて侵入テストを*実施しなければならない*。

15.6.5 ACO_VUL.2 統合の脆弱性分析

依存性：ACO_DEV.2 設計の基本証拠

開発者アクションエレメント:

ACO_VUL.2.1D

開発者は、テストのために統合TOEを提供しなければならない。

内容・提示エレメント:

ACO_VUL.2.1C

統合TOEは、テストに適していなければならない。

評価者アクションエレメント:

ACO_VUL.2.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_VUL.2.2E

評価者は、基本コンポーネント及び依存コンポーネントで識別された残存脆弱性が、その運用環境の統合TOEで悪用不能であることを決定する分析を*実行しなければならない*。

ACO_VUL.2.3E

評価者は、統合TOEの運用環境で基本コンポーネントと依存コンポーネントを使用することで生じる可能性がある脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

ACO_VUL.2.4E

評価者は、統合TOEでの潜在的脆弱性を識別するために、ガイダンス証拠資料、依存情報、及び統合の根拠を使用して、統合TOEの独立脆弱性分析を*実行しなければならない*。

ACO_VUL.2.5E

評価者は、基本的な攻撃能力を持つ攻撃者からの攻撃に統合TOEが耐えられることを実証するために、識別された脆弱性に基づいて侵入テストを*実施しなければならない*。

15.6.6 ACO_VUL.3 強化基本的な統合の脆弱性分析

依存性：ACO_DEV.3 設計の詳細証拠

開発者アクションエレメント:

ACO_VUL.3.1D

開発者は、テストのために統合TOEを提供しなければならない。

内容・提示エレメント:

ACO_VUL.3.1C

統合TOEは、テストに適していなければならない。

評価者アクションエレメント:

ACO_VUL.3.1E

評価者は、提供された情報が、証拠の内容・提示に対する全ての要件を満たしていることを*確認しなければならない*。

ACO_VUL.3.2E

評価者は、基本コンポーネント及び依存コンポーネントで識別された残存脆弱性が、その運用環境の統合TOEで悪用不能であることを決定する分析を*実行しなければならない*。

ACO_VUL.3.3E

評価者は、統合TOEの運用環境で基本コンポーネントと依存コンポーネントを使用することで生じる可能性がある脆弱性を識別するために、公知の情報源の探索を*実行しなければならない*。

ACO_VUL.3.4E

評価者は、統合TOEでの潜在的脆弱性を識別するために、*ガイダンス*証拠資料、依存情報、及び統合の根拠を使用して、統合TOEの独立脆弱性分析を*実行しなければならない*。

ACO_VUL.3.5E

評価者は、**強化基本的な**攻撃能力を持つ攻撃者からの攻撃に統合TOEが耐えられることを実証するために、識別された脆弱性に基づいて侵入テストを*実施しなければならない*。

附属書A
(参考)

開発(ADV)

A.1 ADV_ARC: セキュリティアーキテクチャに関する補足資料

A.1.1 一般

この附属書は、ADV: 開発クラスのファミリで提示されたトピックをさらに説明して追加の例を提供するための補助資料を記載する。

セキュリティアーキテクチャは、TSFが示す特性のセットである。これらの特性には、自己保護、ドメイン分離、及び非バイパス性が含まれる。これらの特性を持つことで、TSFがセキュリティサービスを提供することへの信頼の基礎が提供される。この附属書では、セキュリティアーキテクチャ記述の内容について説明し、これらの特性に関する追加の資料も提供する。

この節の残りの部分では、最初にこれらの特性について説明し、その後TSFがそれらの特性をどのように示すかを記述するために必要な情報の種類について説明する。

A.1.2 セキュリティアーキテクチャの特性

*自己保護*とは、結果としてTSFが変更される場合もあるような外部のエンティティによる操作から自分自身を保護するTSFの能力を指す。これらの特性がないと、TSFはセキュリティサービスを実行できない場合がある。

TOEが、その機能を実行するために他のITエンティティから提供されるサービス又は資源を使用することはよくある(下層のオペレーティングシステムに依存するアプリケーションなど)。このような場合、TSFは、自ら使用するサービスの保護を他のITエンティティに依存するため、自らを完全には自力で保護しない。

*ドメイン分離*とは、TSFが、信頼できない能動的な各エンティティに対して与えられた資源で動作するために個別の*セキュリティドメイン*を作成し、エンティティが他のドメインで実行されないようにするために、ドメインを互いに分離した状態にする特性を指す。例えば、あるオペレーティングシステムTOEは、信頼できないエンティティに関連するプロセスごとに、1つのドメイン(アドレス空間、プロセスごとの環境変数)を提供する。

信頼できないエンティティのアクションは全てTSFによって仲介されるため、このようなドメインが存在しないTOEもある。パケットフィルタリングファイアウォールはこのようなTOEの一例である。このTOEには、信頼できないエンティティドメインが存在せず、TSFによって維持されるデータ構造のみが存在する。このように、ドメインの存在は、1)TOEのタイプ及び2)TOEに課せられるSFRに依存する。TOEが信頼できないエンティティにドメインを提供する場合、このファミリは、あるドメインの信頼できないエンティティが別の信頼できないエンティティのドメインからの(TSFによる仲介なしに影響を受ける)改ざんを回避するように、それらのドメインが互いに分離されることを要求する。

*非バイパス性*は、TSFのセキュリティ機能性(SFRで特定されたもの)が、その特定のメカニズムにとって適切なタイミングで常に呼び出され、回避できないという特性である。例えば、ファイルへのアクセス制御が、SFRを通じてTSFの能力として特定されている場合は、TSFのアクセス制御メカニズムを呼び出さずにファイルへのアクセスを可能にするインタフェース

が存在してはならない(ローディスクアクセスに使用されるインタフェースは、このようなインタフェースの一例であるかもしれない)。

自己保護と同様に、一部のTOEの本質が、TSFの非バイパス性において役割を果たすためにそれらの環境に依存する場合がある。例えば、セキュリティアプリケーションTOEは、下層のオペレーティングシステムに呼び出される必要がある。同様に、ファイアウォールは、内部及び外部のネットワーク間に直接の接続がないこと、及びそれらのネットワーク間の全てのトラフィックがファイアウォールを通る必要があるという事実に依存する。

A.1.3 セキュリティアーキテクチャ記述

セキュリティアーキテクチャ記述は、上記の特性がTSFでどのように示されるかについて説明する。ドメインがどのように定義され、TSFがそれらのドメインをどのように分離するかについて記述する。信頼できないプロセスがTSFにアクセスして変更することをどのようなことによって回避するかについて記述する。TSFの制御下にある全ての資源が適切に保護され、SFRに関連する全てのアクションがTSFによって仲介されることをどのようなことによって保証するかについて記述する。環境がこれらのいずれかにおいて果たす役割(例えば、下層環境によって正しく呼び出されることを想定した場合、セキュリティ機能がどのように呼び出されるか)を説明する。

セキュリティアーキテクチャ記述は、分解された記述の観点でTSFの自己保護、ドメイン分離、及び非バイパス性というTSFの特性を示す。この記述のレベルは、主張されているADV_FSP、ADV_TDS及びADV_IMP要件に必要なTSF記述に対応するものである。例えば、ADV_FSPが使用可能な唯一のTSF記述である場合は、TSFのあらゆる内部動作の詳細を得られないため、有意義なセキュリティアーキテクチャ記述を提供することが難しくなるであろう。

ただし、TOE設計が提供されていれば、最も基本的なレベル(ADV_TDS.1)であっても、TSFを構成するサブシステムに関するある程度の情報が示され、それらがどのように動作して自己保護、ドメイン分離、及び非バイパス性を実装するかが記述されているであろう。例えば、TOEに対するおそらく全ての利用者相互作用は、その利用者の全てのセキュリティ属性を使用して利用者に代わって作動するプロセスを通じて行われるものに制約される。セキュリティアーキテクチャ記述は、このようなプロセスがどのように発生し、そのプロセスのふるまいがTSFによってどのように(TSFを破壊できないように)制約され、そのプロセスの全てのアクションがTSFによってどのように調停されるか(それによってTSFをバイパスできない理由を説明)などを記述するであろう。

提供されるTOE設計がより詳細である(例えばモジュールレベル)場合、あるいは実装表現も提供される場合は、それに応じてセキュリティアーキテクチャ記述もより詳細になり、利用者のプロセスがTSFプロセスとどのように通信するか、複数の異なる要求をTSFがどのように処理するか、どのパラメタが渡されるか、どのようなプログラムによる保護が行われるか(バッファオーバーフロー防止、パラメタ境界チェック、チェック時/使用時についてのチェックなど)が説明されるであろう。同様に、STがADV_IMPコンポーネントを主張しているTOEは、実装固有の詳細が示されるであろう。

セキュリティアーキテクチャ記述で提供される説明は、それらの正確性をテストできるための十分な詳細さであることが期待される。つまり、単純な主張(「TSFはドメインを分離する」など)は、TSFが実際にドメインを作成して分離することを読者に納得させるために役立つ情報を提供しない。

A.1.3.1 ドメイン分離

TOEが完全に自力でドメイン分離を示す場合、これをどのように達成するかについて直接的に記述されるであろう。このセキュリティアーキテクチャ記述は、TSFで定義される各種のドメイン、それらの定義方法(各ドメインにどの資源が割り当てられるか)、保護されない資源をなくす方法、及びあるドメイン内の能動的なエンティティが、別のドメインの資源を改ざんできないようにドメインを分離する方法を説明するであろう。

TOEがドメイン分離で役割を果たすために他のITエンティティに依存する場合、その役割の共有は明確にしなければならない。例えば、単なるアプリケーションソフトウェアであるTOEは、TOEが定義するドメインを正しく具体化するために下層のオペレーティングシステムに依存する。つまり、TOEがドメインごとに個別の処理空間、メモリ空間などを定義すれば、TOEは下層のオペレーティングシステムに依存して正しくかつ悪意なく動作する(例えば、プロセスはTOEソフトウェアが要求した実行空間でのみ実行できる)。

例えば、ドメイン分離を実装するメカニズム(例えば、メモリ管理、ハードウェアが提供する保護された処理モード)は、識別され、記述される。又は、TSFはソフトウェアドメインの分離の実装に寄与する(利用者のアドレス空間とシステムのアドレス空間を明確に区別するなど)ソフトウェア保護構造やコーディング規則を実装する場合がある。

脆弱性分析及びテスト(AVA_VANを参照)アクティビティには、TSFの監視又は直接攻撃を利用することで、記述されたTSFドメイン分離を打ち負かす試みが含まれる可能性が高い。

A.1.3.2 TSF自己保護

TOEが完全に自力で自己保護を示す場合、この自己保護をどのように達成するかについて直接的に記述されるであろう。他の(利用者)ドメインから保護されるTSFドメインを定義するためにドメイン分離を提供するメカニズムが識別及び記述されるであろう。

TOEが自己保護で役割を果たすために他のITエンティティに依存する場合、その役割の共有は明確にしなければならない。例えば、単なるアプリケーションソフトウェアであるTOEは、正しくかつ悪意なく動作するために下層のオペレーティングシステムに依存する。つまり、アプリケーションはそれ自身を破壊する(例えば、実行可能コード又はTSFデータを上書きする)悪意のあるオペレーティングシステムから自分自身を保護できない。

セキュリティアーキテクチャ記述は、またTSFが利用者入力によって自分自身を破壊しないように、TSFが利用者入力をどのように処理するかもカバーする。例えばTSFは、特権の概念を実装し、特権モードのルーチンを使用して利用者データを処理することによって、自分自身を保護することができる。TSFは、TSFコード及びデータを利用者コードやデータから分離するためにプロセッサベースの分離メカニズム(例えば、特権レベルやリング)を使用する場合がある。TSFは(おそらくは利用者のアドレス空間とシステムのアドレス空間を明確に区別することにより)ソフトウェアの分離の実装に寄与するソフトウェア保護構造やコーディング規則を実装する場合がある。

低機能モード(例えば、設置者又は管理者にのみアクセスできる単一利用者モード)で立ち上げてから、評価されたセキュアな構成へ移行する(信頼できない利用者がログインして、TOEのサービスや資源を利用できるモード)TOEの場合、セキュリティアーキテクチャ記述には、評価構成で実行されないこの初期化コードからTSFをどのように保護するかについての説明も含まれる。このようなTOEの場合、セキュリティアーキテクチャ記述では、初期化中にのみ使用可能であるべきサービス(資源への直接アクセスなど)が評価構成でアクセス可能になるのを回避するための説明がなされるであろう。また、TOEが評価構成であるときに初期化コードが実行されるのを回避する方法についても説明するであろう。

TSFが初期のセキュアな状態であると信じ込ませるような結果を招く改変を初期化プロセスが検出できるように、信頼できる初期化コードがどのようにTSF(及びその初期化プロセス)の完全性を維持するかについても説明しなければならない。

脆弱性分析及びテスト(AVA_VANを参照)アクティビティには、TSFの改ざん、直接攻撃、又は監視を利用することで、記述されたTSF自己保護を打ち負かす試みが含まれる可能性が高い。

A.1.3.3 TSFの非バイパス性

非バイパス性の特性は、実施メカニズムのバイパスを許可するインタフェースに関係する。ほとんどの場合、この特性は実装によってもたらされる。つまりその実装では、オブジェクトをアクセス又は操作するインタフェースをプログラマが作成する場合に、そのプログラマが、オブジェクトに対するSFR実施メカニズムの一部であるインタフェースを使用し、それらのインタフェースを回避しないようにする責任を負う。そのため、非バイパス性に関する記述では、2つの広範な領域を扱わなければならない。

第 1 の領域は、SFR実施に対するインタフェースで構成される。これらのインタフェースの特性は、それらを使用してTSFをバイパスできる操作やモードを含んでいないという点である。この決定を行うために、ADV_FSP及びADV_TDSの証拠を大いに使用できることは十分に考えられる。非バイパス性は重要な問題であるため、TSFIを通じて利用できる操作の一部のみが(SFR実施操作であるために)証拠資料として提出され、それ以外の操作は証拠資料として提出されない場合は、TSFIのSFR支援及びSFR非干渉操作が、実施されている方針をバイパスする能力を信頼できないエンティティに与えないことを決定するためにADV_FSP及びADV_TDSで提示された情報に対する追加情報が必要かどうかを、開発者は考慮するべきである。このような情報が必要である場合は、セキュリティアーキテクチャ記述にその情報が組み込まれる。

非バイパス性の第2の領域は、SFR実施に関連しない相互作用を持つインタフェースに関係している。主張されたADV_FSP及びADV_TDSコンポーネントによっては、これらのインタフェースに関する情報が、機能仕様及びTOE設計証拠資料に存在する場合と存在しない場合がある。このようなインタフェース(又はインタフェースのグループ)に対して提示される情報は、実施メカニズムのバイパスが不可能であることを読者が(ADV: 開発クラスで提供されるその他の証拠と同等の詳細レベルで)決定できるほど十分な内容であるべきである。

セキュリティ機能性をバイパスできないという特性は、全てのセキュリティ機能性に等しく適用される。つまり、設計記述は、SFR(FDP_*コンポーネントなど)で保護されるオブジェクト及びTSFが提供する機能性(監査など)を扱うべきである。また、この記述は、セキュリティ機能性に関連するインタフェースを識別するべきである。これには、機能仕様の情報が使用される場合がある。この記述は、オブジェクトマネージャなどのあらゆる設計の構成要素、及びそれらの使用方法も記述しているべきである。例えば、監査レコードを生成する標準マクロをルーチンが使用することになっている場合は、この規則が監査メカニズムの非バイパス性に寄与する設計の一部となる。この文脈での*非バイパス性*とは、「TSF実装の一部が、不当にセキュリティ機能性をバイパスできるか」という質問に回答する試みではなく、実装がいかにしてセキュリティ機能性をバイパスしないかを証拠資料として提供する試みであるという点に注意が必要である。

脆弱性分析及びテスト(AVA_VANを参照)アクティビティには、TSFの回避によって、記述された非バイパス性を打ち負かす試みが含まれる可能性が高い。

A.2 ADV_FSP: 機能仕様に関する補足資料

A.2.1 一般

TSFIの仕様を特定する目的は、テストの実施に必要な情報を提供することである。つまり、TSFとの相互作用を行うために可能な手段がわからなければ、TSFのふるまいを適切にテストすることはできない。

TSFIの仕様の特定には、2つの部分：識別及び記述がある。考えられるTOE、及びそれらの中のTSFが多様であるため、「TSFI」を構成する標準のインタフェースセットは存在しない。この附属書では、どのインタフェースがTSFIであるかを決定する要因についてのガイダンスを提供する。

A.2.2 TOEの非TSF部分

TSFは、セキュリティ機能性を信頼するために利用者が依存しなければならないTOEの全ての部分から構成される。

言い換えれば、TOEのTSFに属さない部分は、TOEのセキュリティ機能性に影響を与えることなく攻撃者によって変更される可能性がある。そうでない場合、TOEのこれらの部分はTSFに含まれなければならない。

TSFとTSFの実装が定義されていれば、TOEの非TSF部分として分類されるTOE部分がさらに存在するかどうかは明らかである。そのような部分はTSFの一部である必要はないが、TOEの一部であることに変わりはない。

TOEのTSF部分と非TSF部分の関係は、その定義とARC特性によって、以下のように与えられる。

— 非TSF部分は、TSFをバイパスしない。

— TSF部分は改ざんから自身を保護する。

TOEのサブシステムでTSFの一部でないものは、以下の(経験則[12]として記述される)条件を満たさなければならない。そのサブシステムは、攻撃者によって置き換えられたとしても、TOEのセキュリティに影響を及ぼしてはならない。

したがって、非TSF部分とTSF部分の間には、何らかの「分離メカニズム」を設けることが望ましい[13]と思われる。なぜなら、その「分離メカニズム」は非TSF部分からTSF部分に影響が及ばないという評価の根拠となり得るからである。

このような「分離メカニズム」は、セキュリティアーキテクチャによって実装することも、明確に実現された実装部分(例えば、TOEのTSFと非TSF部分の間のファイアウォール)によって実装することも可能である。

「分離メカニズム」は、VAN評価のレベルに応じてそれぞれの強度の攻撃者による攻撃に耐えなければならないため、「分離メカニズム」の分析は、脆弱性評定の対象となる。

---

[12] この規則はある程度までしか有効ではない。なぜなら「非 TSF 部分は TSF をバイパスしてはならない」という実際の要件は、与えられた経験則ほど強くないからである。

[13] 「分離メカニズム」は、ここでの提案に過ぎない。開発者は、TOE の TSF 部分と TOE の非 TSF 部分との非バイパス性を示す要件が満たされる限り、他の種類のセキュリティ実装を使用して自由に証拠を提供することができる。

開発者は、セキュリティアーキテクチャ記述において、非バイパス性と自己保護に関する証拠を提供しなければならず、評価者は、ADV_ARC.1に対するサブアクティビティにおいてこの証拠を分析し、脆弱性評価においてその有効性を評定しなければならない。

TOE設計証拠資料の目標は、TSFの境界を決定するための十分な情報を提供し、TSFがSFRをどのように実装するかを記述することである。ADV_TDSファミリは、TOE の非TSFサブシステムの識別のみを要求しているため、さらなる注意が必要である。ADV_FSP及びADV_TDSでは、これらのサブシステムに対するインタフェース記述は提供されない。これらのサブシステムのSFR非干渉性は前提とされるが、開発者によって実証されておらず、評価者によっても詳細には検査されない。しかし、TOE設計の観点からは、上記の分離メカニズムがあり、脆弱性評価でそれが十分な強度があることが確認されている限り、これはそれほど重要ではない。したがって、この「分離メカニズム」は、セキュリティ機能としてTSFを実装、あるいはARCの特性を実施する。しかし、非バイパス性は「純粋なアーキテクチャの特性」によって実施されることもある。

非TSFに分類されるTOEの部分は、(正当な利用者、あるいは攻撃者でさえもその部分を使用するかどうかにかかわらず)TSFをバイパスする手段を提供してはならず、TSFに貢献してはならない。開発者が明確な証拠を提供し、この要件がどのように満たされるかを実証することが重要である。

したがって、非TSFとしてのTOEサブシステムの識別(ADV_TDS.x.1参照)が正しく、その結果、これらのサブシステムの詳細な記述が必要ないことを、開発者は実証しなければならず、評価者は検査しなければならない。評価者の検査には、開発者によって提供されたADV_ARC証拠資料に記述されているARC特性の非バイパス性及び自己保護を含まなければならない(上記段落を参照)。

A.2.3 TSFIの決定

A.2.3.1 一般

TSFに対するインタフェースを識別するには、まずTSFを構成するTOEの部分を識別しなければならない。この識別は、実際にはTOE設計(ADV_TDS)分析の一部であるが、保証パッケージにTOE設計(ADV_TDS)が含まれていない場合は、開発者によるTSFIの識別と記述を通じて暗黙に実行される。この分析では、STのSFRを(全面的又は部分的に)満たすことに寄与しているTOEの一部分が、TSFにあるとみなさなければならない。これには、例えば、TSFの実行時の初期化に寄与するTOEの全ての部分が含まれる。このような部分には、SFRの実施がまだ開始されていないために(起動中など)、TSFが自己を保護できるようになる前に実行されるソフトウェアなどがある。また、アーキテクチャの原則であるTSFの自己保護、ドメイン分離、及び非バイパス性(セキュリティアーキテクチャ(ADV_ARC)を参照)に寄与するTOEの全ての部分も、TSFに組み込まれる。

TSFが定義されると、TSFIが識別される。TSFIは、外部エンティティ(又はTOEのサブジェクトでTSFの範囲外)が、TSFにデータを提供し、TSFからデータを受け取り、又はTSFからサービスを呼び出す、全ての手段で構成される。これらのサービス呼び出しと応答は、TSF境界を越えるための手段である。これらの多くはすぐに明らかになるが、明確にはわからないものもある。TSFIを決定する場合、「潜在的な攻撃者はSFRの破壊を試みる際にどのようにTSFと相互作用する可能性があるか」という点を考えるべきである。

したがって、評価の観点からは、TSFによって保護されている資産を侵害するために、攻撃者がインタフェースを悪用してセキュリティ機能性にアクセスすることができるかどうかも重要である。

攻撃者が使用する可能性のあるTSFのインタフェースは、(SFR実施、SFR支援、SFR非干渉のさらなる分類に関わらず)TSFIに属する。

TSFが外部からアクセスされるかどうか、TSFが外部資源にアクセスするかどうか(例えば、TSFがプラットフォームや利用者を呼び出す)は重要ではない。唯一の基準は、外部からのTSFへの潜在的な干渉があるかどうかである。

様々な状況でのTSFI定義の適用について以下で説明する。

A.2.3.2 電気インタフェース

スマートカードなどのTOEでは、相手側がTOEに論理的にアクセスできるだけでなく、物理的にも完全にアクセスできるため、TSF境界は物理的な境界である。したがって、接触可能な電気インタフェースは、その操作がTSFのふるまいに影響する可能性があるため、TSFIとみなされる。このため、これら全てのインタフェース(電気的な接点)について、適用される様々な電圧などを記述する必要がある。

A.2.3.3 ネットワークプロトコルスタック

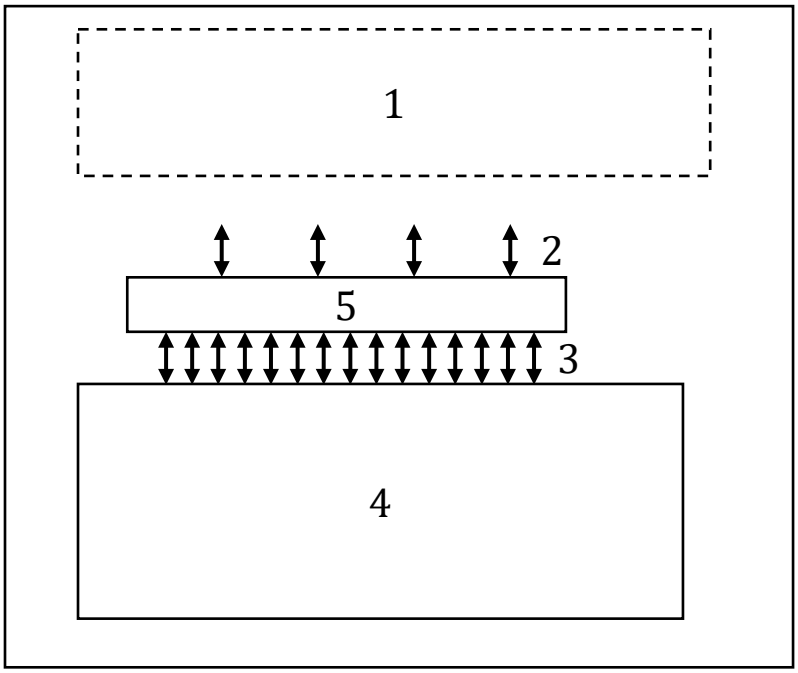
プロトコル処理を実行するTOEのTSFIは、攻撃者が直接アクセスするプロトコル層であろう。これは、プロトコルスタック全体である必要はないが、プロトコルスタック全体になる場合もある。

例えば、TOEが、潜在的な攻撃者がプロトコルスタックの全てのレベルに影響を与えることができる(つまり、任意の信号、任意の電圧、任意のパケット、任意のデータグラムを送信する)ある種のネットワーク装置である場合、TSF境界はスタックの各層に存在する。したがって、機能仕様は、スタックの全ての層における全てのプロトコルを扱う必要があるであろう。

ただし、TOEが、内部のネットワークをインターネットから保護するファイアウォールである場合、潜在的な攻撃者には、TOEに入る電圧を直接操作する手段がないであろう(一切の過剰電圧はインターネットを通じて簡単には渡せないだろうから)。つまり、攻撃者はインターネット層又はさらに上位の層のプロトコルにのみアクセスできるであろう。TSF境界はスタックの各階層に存在する。したがって、機能仕様は、インターネット層と上位の層のプロトコルのみを扱う必要があるであろう(つまり、ファイアウォールへの接触が可能な各通信層を、境界上に現れる適格な入力の構成、及び適格な入力と不適格な入力の結果という点から記述するであろう)。例えば、インターネットプロトコル層の記述では、適格なIPパケットの構成内容、及び適格なパケットと不適格なパケットを受信したときの結果が記述されるであろう。同様に、TCP層の記述では、正常なTCP接続、及び正常な接続が確立されたときの結果と、接続を確立できなかったとき、又は意図せずに接続を切断したときの結果が記述されるであろう。ファイアウォールの目的がアプリケーションレベルのコマンド(例えば、FTPやtelnet)を選別することであると仮定すると、アプリケーション層の記述では、ファイアウォールによって認識され選別されるアプリケーションレベルのコマンド、及び未知のコマンドに遭遇した場合の結果が記述されるであろう。

これらの層の記述では、使用される公開通信標準(例えば、telnet、FTP、TCP)の参照と、どの利用者定義のオプションが選択されたかが記述されるであろう。

A.2.3.4 ラッパー



キー

1 アプリケーション
2 API
3 システム
4 カーネル(TSF)
5 ラッパー

図A.1 — ラッパー

「ラッパー」は、複雑な一連の相互作用を単純化された共通のサービスに変換する。例えば、オペレーティングシステムがアプリケーションで使用できるAPIを作成する場合がこれにあたる(図A.1を参照)。TSFIがシステムコールであるかAPIであるかは、何がアプリケーションから使用可能かに依存するであろう。アプリケーションがシステムコールを直接使用できる場合、システムコールがTSFIとなる。一方、システムコールを直接使用することが何らかの方法で禁止され、全ての通信をAPIを通じて行う必要がある場合、APIがTSFIになるであろう。

グラフィカルユーザインタフェースも同様に、マシンで理解可能なコマンドを利用者が理解しやすいグラフィックに変換する。同様に、TSFIは、利用者がコマンドにアクセスできる場合はコマンドとなり、利用者がグラフィックしか使用できない場合はグラフィック(プルダウンメニュー、チェックボックス、テキストフィールド)になるであろう。

これらの例はどちらも、利用者がよりプリミティブなインタフェース(例えば、システムコール又はコマンド)の使用を禁じられている場合、この制約及び実施の記述が、セキュリティアーキテクチャ記述に含まれるであろうという点で注目に値する(A.1を参照)。また、ラッパーはTSFの一部になるであろう。

A.2.3.5 アクセス不可能なインタフェース

ある特定のTOEでは、全てのインタフェースが*アクセス可能*とは限らない場合がある。つまり、(STにおける)運用環境のセキュリティ対策方針によって、それらのインタフェースへの

アクセスが阻止される場合や、それらを事実上アクセス不可にする方法でアクセスが制限される場合がある。このようなインタフェースは、TSFIとはみなされないであろう。以下に例を挙げる。

a) スタンドアロンファイアウォールに対する運用環境のセキュリティ対策方針で、「ファイアウォールは、訓練を受けた信頼できる人員のみがアクセスできるサーバルーム環境で稼働し、かつ(停電に備えて)無停電電源装置が搭載される」と記述されている場合、物理的なインタフェースと電源インタフェースはアクセス不可能となる。これは、訓練を受けた信頼できる人員が、ファイアウォールを分解したり、その電源装置の機能を無効にしたりすることがないからである。

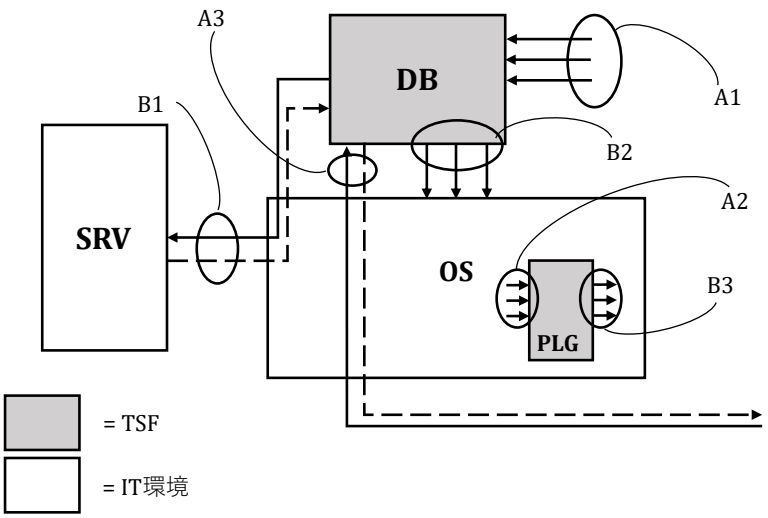
b) ソフトウェアファイアウォール(アプリケーション)に対する運用環境のセキュリティ対策方針で、「OS及びハードウェアは、他のプログラムから改ざんされることのないようにアプリケーションにセキュリティドメインを提供する」と記述されている場合、OS上の他のアプリケーションを介してファイアウォールにアクセスできる(例えば、ファイアウォール実行可能ファイルの削除や修正、ファイアウォールのメモリ空間に対する直接的な読み書き)インタフェースはアクセス不可能になる。これは、運用環境のOS/ハードウェアの部分が、このインタフェースをアクセス不可能にするからである。

c) ソフトウェアファイアウォールに対する運用環境のセキュリティ対策方針で、OSとハードウェアがTOEのコマンドを忠実に実行し、TOEをいかなる方法によっても改ざんしないことが追加で記述されている場合、ファイアウォールがOS及びハードウェアからプリミティブな機能性を取得する(マシンコード命令、ファイルの作成、読み取り、書き込み、削除などを行うOS API、グラフィカルAPIなどを実行する)ようなインタフェースはアクセス不可能になる。これは、OS/ハードウェアがそのインタフェースにアクセスできる唯一のエンティティであり、完全に信頼されているからである。

上記の全ての例で、これらのアクセス不可能なインタフェースはTSFIではないであろう。

A.2.4 例: 複雑なDBMS

図A.2は、複雑なTOEの例として、TOEの境界の外部にあるハードウェアとソフトウェア(これ以降は*IT環境*と呼ぶ)に依存するデータベース管理システムを示している。この例を単純化するために、TOEはTSFと同一になっている。陰影の付いたボックスはTSFを、陰影の付いていないボックスは環境のITエンティティを表している。TSFは、データベースエンジン及び管理GUI(図の*DB*というラベルが付いたボックス)と、OSの一部として実行されてセキュリティ機能を実行するカーネルモジュール(図の*PLG*というラベルが付いたボックス)で構成されている。TSFのカーネルモジュールには、OSの仕様によって定義される入口点がある。OSは、この入口点を呼び出すことによって、特定の機能(デバイスドライバや認証モジュールなど)を呼び出す。ここで重要なのは、この着脱可能なカーネルモジュールが、STの機能要件によって特定されているセキュリティサービスを提供するという点である。



図A.2 — DBMSシステムのインタフェース

IT環境は、オペレーティングシステム自体(*OS*というラベルのボックス)及び外部サーバ(*SRV*というラベルのボックス)で構成される。この外部サーバは、OSと同様に、TSFが依存しているサービスを提供するため、IT環境に含まれている必要がある。この図では、TSFIが*Ax*というラベルで示され、ACO: 統合で証拠資料として提出されるであろうその他のインタフェースが*Bx*というラベルで示されている。次に、これらのインタフェースのグループについてそれぞれ説明する。

インタフェースグループA1は、最も明白なTSFIのセットを表す。これらは、利用者がデータベース及びそのセキュリティ機能性と資源に直接アクセスするために使用するインタフェースである。

インタフェースグループA2は、着脱可能なモジュールによって提供される機能性を取得するためにOSが呼び出すTSFIを表す。これと対照的なのがインタフェースグループB3で、これは、IT環境からサービスを取得するために着脱可能なモジュールが行う呼び出しを表す。

インタフェースグループA3は、IT環境を通過するTSFIを表している。この場合、DBMSは、アプリケーションレベルの専用プロトコルを使用してネットワーク通信を行う。様々なサポートプロトコル(例えば、イーサネット、IP、TCP)を提供する責任はIT環境にあるが、DBMSからサービスを取得するために使用されるアプリケーション層のプロトコルはTSFIであり、そのように証拠資料に記述されなければならない。図の点線は、ネットワーク接続を通じてTSFから返される戻り値やサービスを示している。

*Bx*というラベルのインタフェースは、IT環境内の機能性に対するインタフェースを表す。これらのインタフェースはTSFIではなく、ACOクラスに関連するアクティビティの一部としての統合評価でTOEが使用されるときに、説明又は分析される必要がある。

A.2.5 機能仕様の例

A.2.5.1 一般

ファイアウォールの例は、内部及び外部のネットワーク間で使用される。この例では、受信データの発信元アドレスを検証する(外部データが、内部データから発信しているかのようになりすまそうとしていないことを保証するため)。なりすましの試みが検出された場合、監査ログにその違反の試みが保存される。管理者は、内部ネットワークからファイアウォールに対するtelnet接続を確立してファイアウォールに接続する。管理者のアクションは、認

証、パスワードの変更、監査ログのレビュー、及び内部ネットワークと外部ネットワークのアドレスの設定や変更で構成される。

ファイアウォールの例では、内部ネットワークへの以下のインタフェースがある:

a) IPデータグラム

b) 管理者コマンド

また、外部ネットワークへの以下のインタフェースがある:

a) IPデータグラム

A.2.5.2 インタフェース記述: IPデータグラム

データグラムは、RFC 791で特定されている形式に従う。

— 目的：発信元ホストから宛先ホスト(ともに固定長のアドレスで識別される)にデータのブロック(「データグラム」)を送信する。また、長いデータグラムについては、小さいパケットネットワークを通じて送信できるように、必要に応じて断片化及び再組み立ても提供する。

— 使用方法：これらは下位レベル(データリンクなど)のプロトコルから送られる。

— パラメタ：IPデータグラムヘッダーのフィールド(発信元アドレス、宛先アドレス、断片化禁止フラグ)

— パラメタ記述：[RFC 791の3.1節(「インターネットヘッダーフォーマット」)の定義に従う]

— アクション：なりすまされていないデータグラムを送信する。必要に応じて大きいデータグラムを断片化する。複数の断片をデータグラムに再組み立てする。

— 誤りメッセージ：(なし)。保証された信頼性がない(上位レベルのプロトコルで提供される信頼性)、配信不可能なデータグラム(送信するために断片化する必要があるが、断片化禁止フラグが設定されている)は破棄。

A.2.5.3 インタフェース記述: 管理者コマンド

管理者コマンドは、ファイアウォールとの相互作用の手段を管理者に提供する。これらの**コマンド及び応答は、内部ネットワーク上の任意のホストから確立されたtelnet (RFC 854)接続上で実行される**。使用可能なコマンドは以下のとおりである:

— **Passwd**

  — 目的：管理者パスワードを設定する

  — 使用方法：**Passwd** <*パスワード*>

  — パラメタ：パスワード

  — パラメタ記述：新しいパスワードの値

  — アクション：パスワードを新しく指定された値に変更する。制限はない。

— 誤りメッセージ：なし。

— **Readaudit**

— 目的：管理者に監査ログを提示する

— 使用方法：**Readaudit**

— パラメタ：なし

— パラメタ記述：なし

— アクション：監査ログのテキストを提供する

— 誤りメッセージ：なし。

— **Setintaddr**

— 目的：内部ネットワークのアドレスを設定する。

— 使用方法：**Setintaddr** <*アドレス*>

— パラメタ：アドレス

— パラメタ記述：IPアドレスの先頭から3つ目までのフィールド(RFC 791の定義に従う)。例：123.123.123。

— アクション：内部ネットワークを定義する変数の内部値、つまり、なりすましが試行されているかどうかを判断するために使用される値を変更する。

— 誤りメッセージ：「アドレスは使用中」：識別された内部ネットワークが、外部ネットワークと同一であることを示す。

— **Setextaddr**

— 目的：外部ネットワークのアドレスを設定する

— 使用方法：**Setextaddr** <*アドレス*>

— パラメタ：アドレス

— パラメタ記述：IPアドレスの先頭から3つ目までのフィールド(RFC 791の定義に従う)。例：123.123.123。

— アクション：外部ネットワークを定義する変数の内部値を変更する。

— 誤りメッセージ：「アドレスは使用中」：識別された外部ネットワークが、内部ネットワークと同一であることを示す。

A.3 ADV_INT: TSF内部構造に関する補足資料

A.3.1 一般

TOEの多様性から、「適切に構成された」又は「最小の複雑さ」よりも具体的なものを体系化することは不可能である。構造と複雑さに関する判断は、TOEで使用される特定の技術から導き出されることが期待される。例えば、ソフトウェアエンジニアリングの分野で挙げられる特性を示す場合、ソフトウェアは適切に構成されたものとみなされる可能性が高い。

この附属書では、TSFの手続きベースのソフトウェア部分の構造及び複雑さの評定に関する補足資料を提供する。この資料はソフトウェアエンジニアリングの文献で容易に入手できる情報に基づいている。その他の種類の内部構造(例えば、ハードウェア、オブジェクト指向のコードなどの非手続き型ソフトウェア)については、規範に関する対応する文献を参照するべきである。

A.3.2 手続き型ソフトウェアの構造

A.3.2.1 一般

手続き型ソフトウェアの構造は、従来、そのモジュール性に従って評定される。モジュール設計を使用して作成されたソフトウェアでは、理解のしやすさの実現を支援するために、モジュール間の依存性を明確化し(*結合度*)、互いに強い関連を持つタスクのみをモジュールに組み込む(*凝集度*)。モジュール設計の使用によって、TSFに含まれる要素間の相互依存性が減り、それによって1つのモジュールでの変更又は誤りが、TOE全体に影響を及ぼすリスクが軽減される。その使用によって、設計がより明解になり、予期しない結果が起きないことの保証が高まる。モジュール分解のもう1つの望ましい特性は、冗長なコードや不要なコードの量が減ることである。

TSFにおける機能性の量を最小化することで、評価者及び開発者がSFRの実施に必要な機能性のみに集中できるようになり、理解のしやすさのさらなる向上及び設計又は実装の誤りが発生する可能性のさらなる軽減に役立つ。

モジュール分解、階層化、及び最小化を、設計と実装のプロセスに統合する際には、適切なソフトウェアエンジニアリングについて考慮しなければならない。実用的で有用なソフトウェアシステムは、たいてい、モジュール間の望ましくない結び付き、関連性の弱い機能を含んだモジュール、及びモジュール設計の難解さと複雑さを内包する。理想的なモジュール分解からのこれらの逸脱は、パフォーマンス、互換性、将来予定されている機能性、又はその他の要因に関連する目標や制約を達成するために必要と判断されることが多く、それらの逸脱に対して開発者が提供する正当化に基づいて受け入れられる場合がある。このクラスの要件を適用する際には、適切なソフトウェアエンジニアリングの原則について十分に検討しなければならないが、理解のしやすさを達成するという全体的な目的も達成されなければならない。

A.3.2.2 凝集度

凝集度とは、単一のソフトウェアモジュールによって実行されるタスクが相互に関連する方法とその度合いである。凝集度には、偶発的、通信的、機能的、論理的、連続的、時間的の各タイプがある。以下に、これらの凝集度のタイプを望ましいものから順にリストし、その特徴を示す。

a) *機能的*凝集度：機能的凝集度を持つモジュールは、単一の目的に関連するアクティビティを実行する。この特性を持つモジュールは、単一の目的に関連するアクティビティを実行する。機能的に凝集するモジュールは、スタックマネージャやキューマネージャのように、単一タイプの入力を単一タイプの出力に変換する。

b) *連続的*凝集度：連続的凝集度を持つモジュールでは、モジュールに含まれる各機能の出力がモジュールに含まれるその次の機能の入力となる。連続的に凝集するモジュールの

例としては、監査レコードを書き出す機能及び特定タイプの監査違反の累積数をカウントし続ける機能を含んだモジュールが挙げられる。

c) *通信的*凝集性：通信的凝集性を持つモジュールでは、ある機能が同じモジュール内の他の機能に対して出力を生成するか、又は他の機能からの出力を使用する。通信的に凝集するモジュールの例としては、強制チェック、任意チェック、及び権限(capability)チェックを含んだアクセスチェックモジュールが挙げられる。

d) *時間的*凝集度：時間的凝集度を持つモジュールでは、大体同時に実行する必要がある複数の機能を含む。時間的に凝集するモジュールの例としては、初期化、回復、シャットダウンなどのモジュールが挙げられる。

e) *論理的*(又は*手続き的*)凝集度：論理的凝集度を持つモジュールは、類似するアクティビティを異なるデータ構造に対して実行する。モジュールの機能が、別々の入力に対して、関連しているが異なっている操作を実行する場合、そのモジュールは論理的凝集度を示す。

f) *偶発的*凝集度：偶発的凝集度を持つモジュールは、関連がまったくない、又はほとんどない複数のアクティビティを実行する。

A.3.2.3 結合度

結合度は、ソフトウェアモジュール間の相互依存の方法とその度合いである。結合には、コール、共通、内容の各タイプがある。以下に、これらの結合度のタイプを望ましいものから順にリストし、その特徴を示す。

a) コール: 2つのモジュールが、厳密にそれぞれの証拠資料に記述された機能コールの使用を通して通信する場合、これらのモジュールはコール結合されている。コール結合の例としては、次に定義するデータ、スタンプ、制御がある。

   1) *データ*: 2つのモジュールが、厳密に単一のデータ項目を表すコールパラメタの使用を通して通信する場合、それらのモジュールはデータ結合されている。

   2) *スタンプ*: 2つのモジュールが、複数のフィールドからなるコールパラメタ、又は意味のある内部構造を持つコールパラメタの使用を通して通信する場合、それらのモジュールはスタンプ結合されている。

   3) *制御*: 2つのモジュールの一方が、他方の内部ロジックに影響するように意図された情報を渡す場合、それらのモジュールは制御結合されている。

b) *共通*: 2つのモジュールが共通データ領域又は共通システム資源を共有する場合、それらのモジュールは共通結合されている。グローバル変数は、それを使用するモジュールが共通結合されていることを示す。グローバル変数による共通結合は、一般に許可されているが、その程度は限定される。例えば、グローバル領域に置かれているが、単一のモジュールのみが使用する変数は、配置が不適切であり、削除するべきである。このほかに、グローバル変数の適切性を評定する際には、次の要因を検討する必要がある:

   1) グローバル変数を改変するモジュールの数: 一般に、グローバル変数の内容を制御する責任は1つのモジュールのみに割り当てるべきであるが、第2のモジュールと責任を共有する状況も発生する。このような場合は、十分な正当性を提示する必要がある。2つより多いモジュール間でこの責任を共有することは受け入れられない。

(この評定を行う際には、変数の内容について実際に責任を負うモジュールを注意して決定するべきである。例えば、単一のルーチンを使用して変数を改変するが、そのルーチンが単にその呼び出し側から要求された改変を実行する場合、呼び出し側モジュールが責任を負うことになり、責任を負うモジュールが複数になる可能性がある)。さらに、複雑さ決定の一環として、2つのモジュールがグローバル変数の内容について責任を負う場合は、それらのモジュール間で改変がどのように調整されるかが明確に示されるべきである。

2) グローバル変数を参照するモジュールの数: 一般に、グローバル変数を参照するモジュールの数に制限はないが、多数のモジュールが参照する場合は、有効性と必要性を検査すべきである。

c) *内容*: 2つのモジュールの一方が他方の内部を直接参照できる場合、それらのモジュールは内容結合されている(例えば、他方のモジュールのコードを改変する場合やその内部ラベルを参照する場合)。その結果、一方のモジュールの内容の一部又は全部が、他方のモジュールに実質的に包含される。内容結合は非通知型モジュールインタフェースを使用しているとみなすことができる。これは、通知型モジュールインタフェースを使用するコール結合とは対照的である。

A.3.3 手続き型ソフトウェアの複雑さ

複雑さとは、コードが実行される際の決定ポイント及び論理パスの尺度である。ソフトウェアエンジニアリングの文献では、複雑さは、コードのロジックと流れの理解を妨げるため、ソフトウェアの否定的な特性として挙げられる。コードの理解を妨げるもう1つのものとして、使用されない又は冗長という点で不要なコードな存在が挙げられる。

階層化の使用によって抽象化のレベルを分離し、循環的な依存性を最小化することで、TSFがさらに理解しやすくなり、TOEのSFRが実装で正確かつ完全に実現されることの保証が高まる。

複雑さの軽減には、相互依存性の軽減/排除という概念も含まれる。これは、同じ層にあるモジュールと別々の層にあるモジュールの両方に関係する。相互に依存するモジュールは互いに依存して1つの結果を導き出すが、それがデッドロック状態を招いたり、さらに悪い場合は、最終的な結論が決まらずに、ある瞬間の計算環境の影響を受ける競合状態(例えば、チェックのタイミングと使用のタイミングの問題)に陥る可能性がある。

設計の複雑さの最小化は、リファレンス確認メカニズムの主要な特質であり、その目的は、容易に理解できるTSFを実現して、TSFを完全に分析できるようにすることである(リファレンス確認メカニズムには、他にもTSFの自己保護や非バイパス性などの重要な特性がある。これらの特性は、ADV_ARCファミリの要件で扱われる)。

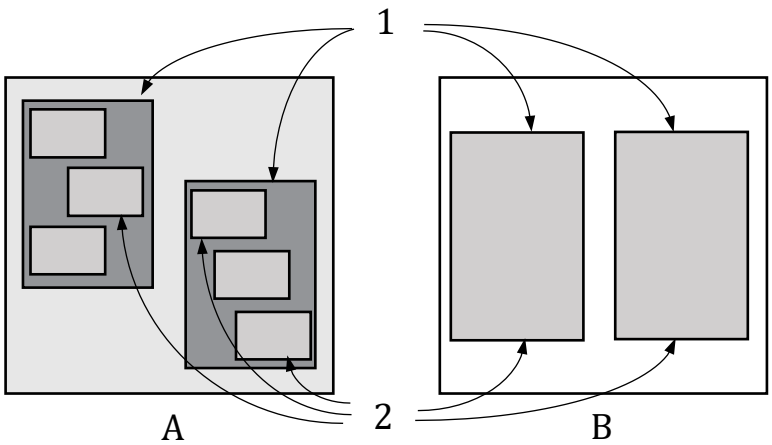
A.4 ADV_TDS: サブシステム及びモジュール

A.4.1 一般

この節では、TDSファミリに関する追加のガイダンスを提供し、このファミリでの「サブシステム」と「モジュール」という用語の用法を示す。その後に、提供される詳細レベルが高くなると、低い詳細レベルに対する要件が縮小されるしくみについて説明する。

A.4.2 サブシステム

図A.3は、TSFの複雑さに応じて、設計がサブシステムとモジュールの観点から記述される場合(サブシステムの抽象レベルがモジュールより高い場合)と、単にある抽象レベル(例えば、下位の保証レベルでの*サブシステム*、上位の保証レベルでの*モジュール*)の観点から記述される場合があることを示している。下位レベルの抽象(モジュール)が提示される場合、上位レベルの抽象(サブシステム)に課せられる要件は、基本的に何もしなくても満たされる。この概念については、この後のサブシステムとモジュールに関する説明でさらに詳しく述べる。



キー

A TOE1(複雑)
B TOE2(単純)
1 *サブシステム*
2 *モジュール*

図A.3 — サブシステム及びモジュール

開発者は、*サブシステム*の観点からTOEの設計を記述することが期待される。「サブシステム」は、特に曖昧になるように選択された用語であるため、TOE固有の単位(例えば、サブシステムやモジュール)を指す可能性もある。サブシステムは、サブシステムの記述に関する要件が満たされている限り、範囲さえも均一でなくてよい。

サブシステムの第1の用法は、TSFの境界、つまりTSFを構成するTOEの部分を区別することである。一般に、サブシステムがいずれかのSFRの正しい動作に影響する(設計又は実装によって)能力を持つ場合、そのサブシステムはTSFの一部である。例えば、ソフトウェアが様々なハードウェア実行モードに応じてドメイン分離(A.1を参照)を提供し、その1つのドメインでSFR実施コードが実行される場合、そのドメインで実行される全てのサブシステムはTSFの一部とみなされるであろう。同様に、そのドメインの外部にあるサーバがSFRを実装する場合(例えば、そのサーバが管理するオブジェクト上でアクセス制御方針を実施する場合)は、そのサーバもTSFの一部とみなされるであろう。

サブシステムの第2の用法は、TSFを記述するための構造を提供することで、その記述レベルは、TSFの作用を記述する一方で、モジュール記述(後で説明)に見られる下位レベルの実装詳細は必ずしも含んでいない。サブシステムは、上位レベル(実装の詳細がほとんど記述されない)又は詳細レベル(実装に対するより深い洞察を提供)のいずれかで記述される。サブシステムに提供される記述のレベルは、サブシステムがSFRの実装にどの程度責任を持っているかによって決まる。

*SFR実施*サブシステムは、任意のSFRのエレメントを実施するためのメカニズムを提供するサブシステム、又はSFRの実施に責任を持つサブシステムを直接支援するサブシステムである。サブシステムがSFR実施TSFIを提供(実装)する場合、そのサブシステムはSFR実施である。

サブシステムには、*SFR支援*又は*SFR非干渉*と識別されるものもある。SFR支援サブシステムは、SFRを実装するためにSFR実施サブシステムが依存しているサブシステムであるが、SFR実施サブシステムほど直接的な役割を果たさない。SFR非干渉サブシステムは、支援の役割においても実施の役割においても、SFRを実装するために依存されないサブシステムである。

A.4.3 モジュール

モジュールは、一般的に、TSF内部構造(ADV_INT)で説明される特性の観点から特徴を表すことができる比較的小さいアーキテクチャの単位である。ADV_TDS.3基本モジュール設計(又はそれ以上)の要件及びTSF内部構造(ADV_INT)の要件の両方がPP又はSTに存在する場合、TOE設計(ADV_TDS)の要件の観点での「モジュール」は、TSF内部構造(ADV_INT)の要件に対する「モジュール」と、同じものを指す。サブシステムとは異なり、モジュールは、実装表現のレビューに対するガイドとしての役割を果たすことができる詳細レベルで実装を記述する。

TOEによっては、モジュールとサブシステムが同じ抽象概念を指す場合があるので注意が必要である。ADV_TDS.1基本設計とADV_TDS.2アーキテクチャ設計(モジュールレベルでの記述を要求しない)の場合、サブシステム記述はTSFに関する最下位レベルの詳細を提供する。ADV_TDS.3基本モジュール設計(モジュール記述を要求する)の場合、これらの記述は最下位レベルの詳細を提供する一方で、サブシステム記述(別のものとして存在する場合)は、単にモジュール記述の枠組みを示すために使用される。つまり、モジュール記述が存在する場合は、詳細なサブシステム記述を提供する必要がない。きわめて単純なTOEでは、独立した「サブシステム記述」が必要ない。つまり、モジュールによって提供される証拠資料で要件を満たすことができる。複雑なTOEでの(TSFに関する)サブシステム記述の目的は、読者がそれぞれの分析の焦点を適切に絞り込むことができるように枠組みを提供することである。図A.3には、この違いが示されている。

SFR実施モジュールは、STのSFRを完全に又は部分的に実装するモジュールである。このようなモジュールはSFR実施TSFIを実装するかもしれないが、SFRで表現されている一部の機能性(例えば、監査機能やオブジェクト再使用機能性)が単一のTSFIに直接結び付いていないかもしれない。サブシステムの場合と同様に、SFR支援モジュールは、SFR実施モジュールから依存されているが、SFRを直接実装する責任を負わないモジュールである。SFR非干渉モジュールは、SFRの実施を直接的にも間接的にも扱わないモジュールである。

「直接実装する」が何を意味するかの判断は、やや主観的になるので注意が必要である。最も狭義には、要件を実装する比較やゼロ化操作などを実際に実行する1、2行のコードを意味すると解釈できる。解釈を広げると、SFR実施TSFIに応答して呼び出されるモジュール、及びそのモジュールによって呼び出されることがある全てのモジュール(呼び出しが完了するまで続く)までが含まれるかもしれない。どちらの解釈も特に十分ではない。なぜなら、最初の解釈は意味が狭いために、重要なモジュールが間違ってSFR支援と分類される可能性があり、2番目の解釈では、実際にはSFR実施でないモジュールがSFR実施と分類されてしまうからである。

モジュールの記述は、その記述からモジュールの実装を作成できるものであるべきであり、その結果の実装は1)提示されるインタフェースの観点から実際のTSF実装と同一であり、2)設計で記述されるインタフェースの使用において同一であり、3)TSFモジュールの目的の記述と機能的に同等である。例えば、RFC 793はTCPプロトコルの上位レベル記述を提供する。これは、必ずしも実装には依存していない。これは、豊富な詳細を提供するが、実装の特定

ではないため、適切な設計記述ではない。実際の実装はRFCで指定されているプロトコルを追加でき、実装の選択(例えば、実装の様々な部分で、グローバルデータとローカルデータのどちらを使用するか)は実行される分析に対して影響を与える可能性がある。TCPモジュールの設計記述は、(RFC 793で定義されたインタフェースだけではなく)実装によって提示されるインタフェース、及びTCPを(TSFの一部であったと想定して)実装しているモジュールに関連する処理のアルゴリズム記述をリストするであろう。

設計では、モジュールが提供する機能(目的)、モジュールが提示するインタフェース(基準によって求められる場合)、それらのインタフェースからの戻り値、モジュールが使用するインタフェース(他のモジュールが提示)(記述されることが要求されるそれらのインタフェースの提供)、及びモジュールの実装方法に対して適切な技法を用いて、モジュールがその機能性の提供方法を示す記述の観点から、モジュールが詳細に記述される。

モジュールの目的は、モジュールが提供する機能を示しながら記述されるべきである。また、アーキテクチャにおけるモジュールの機能を読者が全般的に把握できるように十分に記述されるべきである。

モジュールが提示するインタフェースは、提供されている機能性を呼び出すために他のモジュールが使用するインタフェースである。インタフェースには、*明示的な*インタフェース(例えば、他のモジュールによって呼び出されるコーリングシーケンス)及び*暗黙の*インタフェース(例えば、モジュールによって操作されるグローバルデータ)の両方が含まれる。インタフェースは、どのように呼び出されるかという観点から、及び戻される全ての値の観点から記述される。この記述には、パラメタのリスト、及びこれらのパラメタの記述が含まれるであろう。あるパラメタが値のセット(例えば「フラグ」パラメタ)であることを期待されていた場合、処理しているモジュールに影響を与えるパラメタがとり得る値の完全なセットが特定されるであろう。同様に、データ構造を表すパラメタは、データ構造の各フィールドが識別及び記述されるように記述される。グローバルデータは、それらの目的を理解するのに求められる程度に記述されるべきである。グローバルデータ構造に求められる記述のレベルは、モジュールインタフェースに対する記述のレベルと同一であることを必要とする。そこでは、入力パラメタと戻り値が、データ構造における個々のフィールドとそれらの可能値に対応するグローバルデータ構造は、更新されるグローバルデータ構造、又は、グローバルデータ構造から取り出される情報について、モジュール設計が十分な情報を含んでいる限り、それらを操作したり読んだりするモジュールとは別に記述してもよい。

プログラミング言語によっては、明白とはならない追加の「インタフェース」を持つ可能性がある。この例として挙げられるのは、C++における演算子/関数のオーバーロードがあるだろう。クラス記述におけるこの「暗黙のインタフェース」は、モジュール設計の一部としても記述されるであろう。モジュールは1つのインタフェースのみを提示する可能性があるが、関連するインタフェースの小規模なセットをモジュールが提示することのほうがより一般的である。

モジュールによって使用されるインタフェースを記述することが要求される場合、モジュールの設計記述又は呼び出されるモジュールの目的のいずれかから、呼び出されるモジュールから期待されるサービスは何かが明確でなければならない。例えば、モジュールAが記述されており、それがモジュールBのバブルソートルーチンを使用する場合、なぜモジュールBのバブルソートルーチンが呼ばれ、そしてこの呼び出しがSFRの実現に寄与するものは何か、モジュール間の相互作用の記述が特定することを可能にしなければならない。モジュールBのインタフェース(ADV_TDSのレベルとモジュールBの分類が、インタフェースの記述を要求する場合に提供される)の一部として、モジュールBのバブルソートルーチンのインタフェースと目的が記述されなければならない、そしてモジュールAは、単に、このルーチンを使って、どんなデータがソートされる必要があるかを、特定する必要がある。適切な記述は、

「モジュールAは、ユーザ名をアルファベット順にソートするために、モジュールBのインタフェース*double_bubble()*を呼び出す。」であろう。

ユーザ名のソートが、どのSFRの実施においても重要でない場合（例：単なる速度向上、アルゴリズム上同一なモジュールAの実装方法でもユーザ名のソートが回避可能）、モジュールBのバブルソートルーチンはSFR実施ではなく、モジュールAの記述において、ユーザ名が性能強化のためにアルファベット順にソートされることを説明することで十分であることに注意すること。モジュールBは“SFR支援”のみと分類されてもよく、選択されたADV_TDSのレベルは、SFR支援モジュールのインタフェースは記述される必要があるか、あるいはモジュールBの目的を単に記述するだけで十分であるかを示す。

すでに説明したように、モジュールのアルゴリズム記述は、アルゴリズム方式でモジュールの実装を記述するべきである。これは、擬似コード、フローチャート、又は(ADV_TDS.3基本モジュール設計での)非形式的説明文で実現することができる。この記述では、モジュール入力と呼び出される関数をどのように使用してモジュールの機能が達成されるかが説明される。また、グローバルデータ、システム状態、モジュールによって生成される戻り値に対する影響について言及する。この記述は、TOEの実際の実装にきわめて似た実装を導き出せる程度の詳細レベルにある。

ソースコードは、モジュール証拠資料の要件を満たさないことに注意するべきである。モジュール設計は実装を記述するが、実装そのものでは*ない*。ソースコードの周囲にあるコメントがソースコードの意図を説明している場合は、それらが十分な証拠資料となることがある。単に各コード行の処理内容を記述するインラインコメントは、モジュールが達成するべき内容を説明しないので役に立たない。

以下のエレメントでは、サブシステムとモジュールについて検討されたラベル(SFR実施、SFR支援、及びSFR非干渉)が、開発者が提供する必要がある情報の量とタイプを記述するために使用される。エレメントは、開発者が特定された情報*のみ*を提供することを期待しないように構成されている。つまり、開発者が提供するTSFの証拠資料が以下の要件の情報を提供する場合、開発者が自らの証拠資料を更新し、サブシステムとモジュールをSFR実施、SFR支援、又はSFR非干渉とラベル付けすることは期待されない。このラベル付けの主な目的は、成熟した開発方法(及び詳細なインタフェース及び設計証拠資料などの関連する資料)を確立していない開発者が、過度なコストをかけずに必要な証拠を提供できるようにすることである。

A.4.4 レベル付けアプローチ

何がSFR実施で何がSFR支援かを決定(さらに場合によっては何がSFR非干渉かも決定)する際には主観が影響するため、このファミリでは次のパラダイムが採用されている。このファミリの前半のコンポーネントでは、開発者がサブシステムをSFR実施などに分類するための決定を行い、適切な情報を提供する。また、評価者がこの主張を支援するために検査する追加の証拠はほとんどない。望まれる保証のレベルが高くなると、開発者はやはり分類に関する決定を行うが、評価者には、開発者の分類を確認するために使用する証拠がより多く提供されるようになる。

評価者の分析をTOEのSFR関連部分に集中させるために、特に低いレベルの保証では、最初にSFR実施アーキテクチャエンティティについてのみ詳細な情報が要求されるように、ファミリのコンポーネントがレベル付けされる。保証のレベルが高まるにつれて、SFR支援エンティティ及び(最終的には)SFR非干渉エンティティについて、より多くの情報が要求されるようになる。完全な情報が要求されている場合でも、この情報全てを同じ詳細レベルで分析することは要求されないので注意するべきである。いずれの場合も、*必要な*情報が提供され、分析されるかどうかに焦点を置くべきである。

表A.1は、記述されるアーキテクチャエンティティについて各ファミリコンポーネントで要求される情報を要約したものである。

表A.1 — 記述の詳細に関するレベル付け

	TSFサブシステム			TSFモジュール		
	SFR実施	**SFR支援**	**SFR非干渉**	**SFR実施**	**SFR支援**	**SFR非干渉**
ADV_TDS.1 基本設計(非形式的表現)	構造、SFR実施のふるまいの要約、相互作用	指定支援[a]	指定支援			
ADV_TDS.2 アーキテクチャ設計(非形式的表現)	構造、SFR実施のふるまいの詳細な記述、その他のふるまいの要約、相互作用	構造、他のふるまいの要約、相互作用	指定支援、相互作用			
ADV_TDS.3 基本モジュール設計(非形式的表現)	記述、相互作用	記述、相互作用	記述、相互作用	目的、SFRインタフェース[b]	相互作用、目的	相互作用、目的
ADV_TDS.4 準形式的モジュール設計(準形式的表現)	記述、相互作用	記述、相互作用	記述、相互作用	目的、SFRインタフェース	目的、SFRインタフェース	相互作用、目的
ADV_TDS.5 完全な準形式的モジュール設計(準形式的表現)	記述、相互作用	記述、相互作用	記述、相互作用	目的、全インタフェース[c]	目的、全インタフェース	目的、全インタフェース
ADV_TDS.6 形式的な上位レベルの設計提示を伴う完全な準形式的モジュール設計(準形式的表現、追加の形式的表現)	記述、相互作用	記述、相互作用	記述、相互作用	目的、全インタフェース	目的、全インタフェース	目的、全インタフェース

[a] *指定支援*とは、サブシステム/モジュールの分類を支援するのに十分な証拠資料のみが必要であることを意味する。

[b] *SFRインタフェース*とは、各SFR関連インタフェースに対し、戻り値と他のモジュールによばれるインタフェースを含むモジュール記述を意味する。

[c] *全インタフェース*とは、各インタフェースに対し、戻り値と他のモジュールに呼ばれるインタフェースを含むモジュール記述を意味する。

A.4.5 セキュリティの関連性

CCは、記述、証拠及び分析をTOEのセキュリティ機能性に集中させる。そのためには、TOEの機能的及び物理的な部分のセキュリティ関連性の特徴付けが必要となる。インタフェース、サブシステム、モジュールは、(暗黙的又は明示的に)「SFR実施」、「SFR支援」又は「SFR非干渉」に分類できる。

開発者証拠と評価分析はTOEに関連し、TSFとそのSFR実施及びSFR支援の実装に焦点を当てる。セキュリティアーキテクチャ記述は、TOEの識別された非TSFサブシステムがTSFをバイパスしておらず、TSFが非TSFコード又はエンティティによる破壊から自身を保護していることを実証しなければならない。開発者は、TOE設計におけるSFR非干渉インタフェース、

サブシステム及びモジュールを記述し、それらが目的、相互作用又は資源の分離のためにTSFに干渉しないことを実証しなければならない。

インタフェース、サブシステム又はモジュールは、以下のとおりである。

— SFRを直接実装している場合、SFR実施。

— SFRの適切な機能をサポートするために、機能的に正しく動作する必要がある場合、SFR支援。

— SFRの実装に関係しない場合、SFR非干渉。

セキュリティ実施機能性とセキュリティ支援機能性に焦点を当てると、他の機能性が非干渉であることの証拠が必要となる。正しく実装されたセキュリティ実施機能及びセキュリティメカニズムであっても、**バイパス、回避、無効化、破壊又は直接攻撃**される可能性がある。非干渉は、TSFが悪用できず、TSF実装の資源への不正アクセスが防止されるか不可能であることを意味する。したがって、インタフェース、サブシステム及びモジュールのセキュリティの関連性を分類し、この分類を脆弱性分析に使用する場合、非バイパス性と自己保護のセキュリティアーキテクチャの側面は重要である。

TSF自己保護は、非TSFのコード又はエンティティによって、TSFが破損されることのないセキュリティアーキテクチャ特性である。これには、TOEの非TSFサブシステムやIT製品の非TOE部分が含まれる。これはSFR非干渉のサブシステム/モジュールの証拠と同様である。

セキュリティドメインは、信頼できないエンティティによる使用のため、互いに分離され、保護されるように、TSFによって提供される環境である。

したがって、評価中の非干渉の分析は、TOEのセキュリティアーキテクチャ(ADV_ARC)の検査を必要とし、ADV_TDS.x.1に規定されるサブシステムの観点でのTOE構造だけでなく、非TSFサブシステムに関するより多くの情報を必要とするかもしれない。開発者は、TSFが正しく定義されている根拠と、以下のような、SFR非干渉モジュールの目的及び他のモジュールとの相互作用の観点からの分析を提供しなければならない。

— **目的**：モジュールがどのようにその機能を提供するか、それ以上の設計上の決定は必要ない。

— **相互作用**：サブシステムやモジュールが通信する理由、渡される情報の特徴(インタフェースの場合よりも詳細が少ない)。

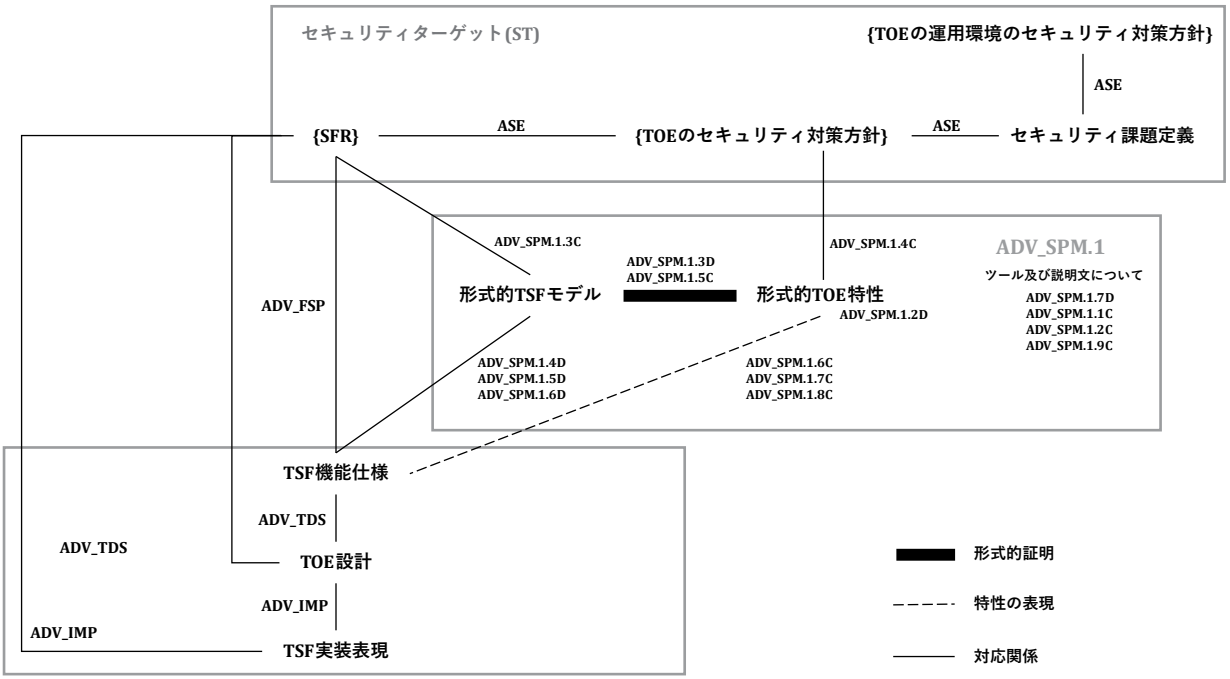
評価中には、機能仕様及びTOE設計の検査、及び脆弱性分析の一環として、非干渉性を分析しなければならない。インタフェース、サブシステム及びモジュールをSFR実施、SFR支援及びSFR非干渉に分類することは、機能仕様、設計及びテストの特定の検査を意味する。TSFの全てのアクセス可能な外部インタフェースとしてTSFIを解釈すると、この分析に役立つであろう。全てのTSFサブシステム(ATE_DPT.1以上)及び全てのTSFモジュール(ATE_DPT.3以上)の機能テストは、それらのセキュリティの分類の正しさを示す証拠を提供すべきである。

A.5 形式的な方法に関する補足資料

形式的な方法は、TSF及びそのふるまいの数学的表現を提供し、ADV_SPM.1(形式的TSFモデル)、ADV_FSP.6(追加の形式的仕様を伴う完全な準形式的機能仕様)及びADV_TDS.6(形式的な上位レベルの設計提示を伴う完全な準形式的モジュール設計)の各コンポーネントで必要

とされる。CEM附属書Cの形式的スタイルでは、形式的な方法に関する補足資料が提供されている。

図A.4は、ADV_SPM.1に規定されるSPMと、ST及び機能仕様によって提供されるTSFの表現との間の関係を示している。



図A.4 — ADV_SPMと他のファミリ及び構成要素との関係

ASEクラスは、SFRとTOEのセキュリティ対策方針との対応に関する要件、SPDエレメントとTOE及び運用環境のセキュリティ対策方針との対応に関する要件を、それぞれ定義する。

TSF表現に固有の各保証ファミリ、すなわち機能仕様(ADV_FSP)、TOE設計(ADV_TDS)、及びTSF実装表現(ADV_IMP)は、その特定のTSF表現とその直上のTSF表現の対応のための開発者アクションに関わる要件、及び特定のTSF表現とSFRのセットの対応の評価のための要件の両方を定義する。

ADV_SPMファミリは、セキュリティの本質的な側面(すなわちTSF)とTOEのふるまいとの関係の形式的な表現である、形式的セキュリティモデルに焦点を当てる。具体的には、形式的TSFモデルは、STに記述されたSFRの全セットによって定義される評価対象システムの形式的記述である。この形式的モデルに対して定義される形式的TOE特性のセットは、TOEのセキュリティ対策方針を全て網羅する。この目的のために、ADV_SPMファミリは以下を定義する。

— TSFを形式的モデルにするための開発者アクションに関する要件(ADV_SPM.1.1D)及び形式的TOE特性のセット(ADV_SPM.1.2D)

— 以下の間の対応関係の内容及び提示に関する要件

  — 形式的TSFモデルとSFRの完全なセット(ADV_SPM.1.3C)

  — 形式的TOE特性とTOEのセキュリティ対策方針(ADV_SPM.1.4C)

保証は、形式的TSFモデルが形式的TOE特性を満たすことを形式的に証明することにより提供される。このため、ADV_SPMファミリは、この形式的証明の要件を定義する(ADV_SPM.1.3D及びADV_SPM.1.5C)。形式的モデルの特性を形式的に証明することによって得られる信頼は、形式的モデルとTSF機能仕様の間の対応の根拠を定義することによって得られる信頼を伴う(ADV_SPM.1.4D)。対応の根拠は、TSF機能仕様の形式的側面にマッピングする際の形式的証明で構成される(ADV_SPM.1.6D)。機能仕様が準形式的なスタイルで記述されている場合は、準形式的な実証からなる(ADV_SPM.1.5D)。ADV_SPMファミリは、TSF機能仕様による形式的TOE特性の保持に関する対応の根拠に関する内容の要件を定義する(ADV_SPM.1.6C/7C/8C)。

ADV_SPMファミリは、基礎となる数学的理論(ADV_SPM.1.1C)、形式的モデリングと証明に使用するツール(ADV_SPM.1.7DとADV_SPM.1.9C)、さらに、各エレメントをサポートし証拠資料に記載する説明文(ADV_SPM.1.2C)に関する要求を含む。

ADV_FSPは、開発者がTSF機能仕様とSFRの対応関係を確立することを要求している。この要件はSPMから独立しているが、ADV_SPM.1が使用される場合、この対応は、一方ではSFRと正式なTSFモデルとの対応、他方ではモデルと機能仕様との対応の副産物となるものである。

図A.4は、機能仕様とST(SFRとTOE特性)の間の関係における形式的TSFモデルの役割を示し、それは、その後、ADV_TDSとADV_IMPファミリの要件によって設計と実装表現に伝播される。

附属書B
(参考)

統合(ACO)

B.1 一般

この附属書の目標は、構成評価とACO基準の背景にある概念を説明することである。本附属書ではASE基準の定義は行わない。定義は、9章にある。

B.2 統合TOE評価の必要性

IT市場は全体として、特定のタイプの製品/技術を提供するベンダで構成されている。PCのハードウェアベンダがアプリケーションソフトウェア及び/又はオペレーティングシステムも提供する場合や、チップメーカーが自社のチップセット専用のオペレーティングシステムを開発する場合のように、部分的な重複も見られるが、1つのITソリューションが様々なベンダによって実装されることは少なくない。

個々のコンポーネントの保証に加えてコンポーネントの組み合わせ(統合)における保証が必要な場合がある。コンポーネントの技術的な統合に必要な一定の資料を配布する際にはベンダ間で協力が行われるが、詳細な設計情報及び開発プロセス/手続きの証拠を提供するところまで合意が拡大することはまれである。あるコンポーネントが依存しているコンポーネントの開発者からこの情報を入手できないことは、依存コンポーネントの開発者がEAL2以上の依存コンポーネントと基本コンポーネント両方の評価を実行するために必要な情報にアクセスできないことを意味する。したがって、依存コンポーネントの評価はどの保証レベルでも実行できるが、EAL2以上のレベルの保証でコンポーネントを構成するには、評価証拠及びコンポーネント開発者向けに行なわれた評価の結果を再使用する必要がある。

ACO基準は、あるITエンティティが別のITエンティティにセキュリティサービスの提供を依存している状況で適用できるように意図されている。サービスを提供するエンティティは「基本コンポーネント」と呼ばれ、そのサービスを受けるエンティティは「依存コンポーネント」と呼ばれる。この関係が存在する状況はいくつかある。例えば、アプリケーション(依存コンポーネント)が、オペレーティングシステム(基本コンポーネント)の提供するサービスを使用する場合がある。あるいは、共通のオペレーティングシステム環境内又は別々のハードウェアプラットフォーム上で稼働する2つのリンクされたアプリケーションという意味では、この関係はピアツーピアの可能性もある。比較的重要でないピアにサービスを提供する主要なピアが存在する場合は、その主要なピアが基本コンポーネント、比較的重要でないピアが依存コンポーネントとみなされる。ピアが互いにサービスを提供し合う場合は、それぞれのピアが、提供されるサービスについては基本コンポーネントとみなされ、要求されるサービスについては依存コンポーネントとみなされる。この場合、ACOコンポーネントの繰返しによって、各タイプのコンポーネントピアに全ての要件を適用する必要がある。

また、この基準は、より複雑な関係で段階的により広く適用されるように意図されているが(この場合、依存コンポーネントと基本コンポーネントそのもので構成される統合TOEは、別の統合TOEの基本コンポーネントになる)、これにはさらなる解釈が必要となる場合がある。

統合評価は、個々のコンポーネント評価の結果に基づいているため、統合TOE評価では、個々のコンポーネントがそれぞれ単独で評価される必要もある。統合TOE評価が開始される

ときに、依存コンポーネントの評価が進行中でも構わない。ただし、統合TOE評価が完了する前に依存コンポーネント評価が完了する必要がある。

統合評価アクティビティは、依存コンポーネント評価と同時に行われる場合がある。これは次の2つの要因による:

a) 経済/ビジネス上の要因：統合評価アクティビティには、依存コンポーネント評価からの評価用提供物件が必要であるため、依存コンポーネントの開発者は、統合評価アクティビティのスポンサーとなるか、又はそれらのアクティビティを支援することになる。

b) 技術的な要因：コンポーネントは、依存コンポーネントが最近コンポーネント評価を受けており(評価中であり)、かつ評価に関連する全ての評価用提供物件が入手可能であることを了解したうえで、基本コンポーネントから必要な保証が提供されているかどうかを考慮する(例えば、コンポーネント評価完了後の基本コンポーネントに対する変更を考慮)。したがって、統合中に依存コンポーネント評価アクティビティの再検証を要求するアクティビティは発生しない。また、依存コンポーネントの評価中に基本コンポーネントによって依存コンポーネントのテスト構成(の1つ)が作成され、ACO_CTTによってこの構成で基本コンポーネントが考慮されているか検証される。

依存コンポーネントの評価から得られた評価証拠は、統合TOE評価アクティビティに対して必要な入力である。統合TOE評価アクティビティに対して必要な入力である基本コンポーネントの評価から得られる唯一の評価資料は以下のとおりである:

— 基本コンポーネント評価で報告される、基本コンポーネントにおける残存脆弱性。これは、ACO_VULアクティビティに必要となる。

基本コンポーネントのコンポーネント評価で得た評価結果は再使用されるべきであるため、統合TOE評価には、基本コンポーネントアクティビティで得られた他の評価証拠は必要ないようにすべきである。統合TOEのTSFに、基本コンポーネントのコンポーネント評価中にTSFとみなされた部分よりも多くのものを含んでいる場合、基本コンポーネントの追加情報が必要になる場合がある。

基本コンポーネント及び依存コンポーネントのコンポーネント評価は、ACOコンポーネントに対して最終的な判定が行われる時点までに完了していると想定される。

ACO_VULコンポーネントでは、最高でも強化基本的な攻撃能力を持つ攻撃者に対する抵抗力のみが考慮される。これは、基本コンポーネントが、依存コンポーネントが依存するサービスをACO_DEVアクティビティの適用を通してどのように提供するかに関して提供できる設計情報のレベルに起因している。したがって、CAPを使用して統合TOE評価から得られる信頼は、EAL4コンポーネントTOE評価で得られる信頼と同様のレベルに制限される。ただし、統合TOEを構成するコンポーネント内の保証は、EAL4よりも高いレベルになる場合がある。

B.3 統合TOEに対するセキュリティターゲット(ST)評価の実行

統合TOE(評価コンポーネント+依存コンポーネント)の評価のために、開発者からSTが提出される。このSTは、統合TOEに適用される保証パッケージを識別し、コンポーネント評価で得られた保証を利用することで統合エンティティでの保証を提供する。

ST内でコンポーネントの統合を考慮する目的は、環境と要件の両方の観点からコンポーネントの両立性の有効性を確認することと、統合TOEのSTがコンポーネントST及びそれらのSTで表現されているセキュリティ方針と一貫していることを評価することである。これには、コンポーネントST及びそれらのSTで表現されているセキュリティ方針が両立できることの判断も含まれる。

コンポーネントSTが統合TOEのSTでどのように表現されているかの根拠を提供しつつ、統合TOEのSTはコンポーネントSTの内容を参照してもよいし、ST作成者は統合TOEのST内でコンポーネントSTの記述を繰り返してもよい。

統合TOEのSTに対するASE_CCL評価アクティビティの実施中に、評価者は、統合TOEのSTでコンポーネントSTが正確に表現されていることを判断する。これは、統合TOEのSTがコンポーネントTOEのSTと適合することが論証可能であることを決定することで達成される。また、評価者は、運用環境に対する依存コンポーネントの依存性が、統合TOEで十分に満たされていることを決定する必要がある。

統合TOEの記述では、統合ソリューションが記述される。統合ソリューションの論理的及び物理的な範囲と境界が記述され、コンポーネント間の論理的な境界も識別される。この記述は、各コンポーネントから提供されるセキュリティ機能性を識別する。

統合TOEに対するSFRのステートメントは、SFRを満たすコンポーネントを識別する。SFRが両方のコンポーネントによって満たされる場合は、SFRの様々な側面を満たすコンポーネントが識別できるよう記述される。同様に、統合TOEの要約仕様は、記述されているセキュリティ機能性を提供するコンポーネントを識別する。

統合TOEのSTに適用されるASE: ST評価要件のパッケージは、コンポーネント評価で使用されるASE: ST評価要件のパッケージと一致しているべきである。

統合TOEのSTがコンポーネントSTを直接参照している場合は、コンポーネントSTの評価で得た評価結果を再利用できる。例えば、統合TOEのSTがSFRのそのステートメントの一部についてコンポーネントSTを参照している場合、評価者は、全ての割付操作と選択操作(ASE_REQ.*.3Cで述べられているもの)の完了に対する要件が、コンポーネント評価で満たされていると推察できる。

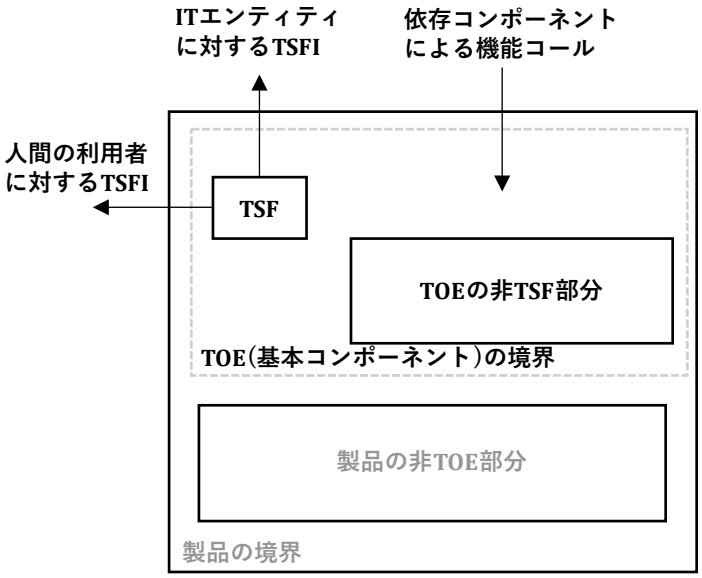
B.4 統合ITエンティティ間の相互作用

基本コンポーネントのTSFは、その統合での可能な適用における依存性に関する知識なしに定義されることが少なくない。この基本コンポーネントのTSFは、基本コンポーネントSFRを実施するために依存しなければならない基本コンポーネントの全ての部分を含むように定義される。これには、基本コンポーネントのSFRの実装に必要な基本コンポーネントの全ての部分が含まれる。

この基本コンポーネントのTSFIは、TSFのサービスを呼び出すためにSFRのステートメントで定義された外部エンティティにTSFが提供するインタフェースを表す。これには、人間の利用者とのインタフェースに加え、外部ITエンティティとのインタフェースも含まれる。ただし、TSFIにはTSFに対するインタフェースのみが含まれるため、TSFIは必ずしも外部エンティティと基本コンポーネント間で利用可能な全てのインタフェースを網羅するインタフェース仕様ではない。基本コンポーネントは、セキュリティ関連とみなされていなかったサービスに対するインタフェースを提示する場合がある。その理由は、サービス特有の目的(例えば、タイプフォントの調整)によるか、又は関連するCCパート2のSFRが基本コンポーネントのSTで主張されていない(例えば、FIA: 識別認証SFRが主張されていない場合のログインインタフェース)ことによる。

基本コンポーネントが提供する機能インタフェースは、セキュリティインタフェース(TSFI)に追加されるもので、基本コンポーネント評価で考慮することは要求されない。機能インタフェースには、基本コンポーネントが提供するサービスを依存コンポーネントが呼び出す場合に使用されるインタフェースが含まれることが多い。

基本コンポーネントには、TSFIをコールすることができる間接的なインタフェースも含まれるかもしれない。例えば、TSFのサービスの呼び出しに使用できるAPIがあり、これは基本コンポーネントの評価中に考慮されていない。

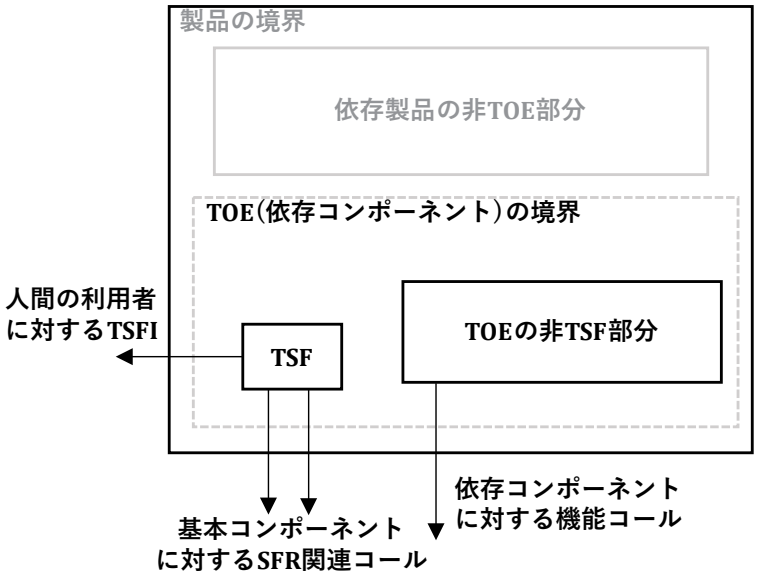


図B.1 — 基本コンポーネントの抽象概念

基本コンポーネントに依存する依存コンポーネントも同様に定義される。コンポーネントSTのSFRで定義される外部エンティティへのインタフェースは、TSFIとして分類され、ADV_FSPで検査される。図B.1にこれを示す。

依存TSFからSFRを支援する環境に対して行われる全てのコールは、宣言された依存コンポーネントSFRの実施を満たすために依存TSFが環境に対して何らかのサービスを要求することを示す。これらのサービスは、依存コンポーネントの境界外にあり、基本コンポーネントは、依存コンポーネントSTで外部エンティティとして定義される可能性はあまりない。このため、依存TSFから下層のプラットフォーム(基本コンポーネント)に対して行われるコールは、機能仕様(ADV_FSP)アクティビティの一環として分析されることはない。基本コンポーネントにおけるこれらの依存性は依存コンポーネントSTにおいて環境のセキュリティ対策方針として表現される。

この依存コンポーネントとインタフェースの抽象概念を、次の図B.2に示す。

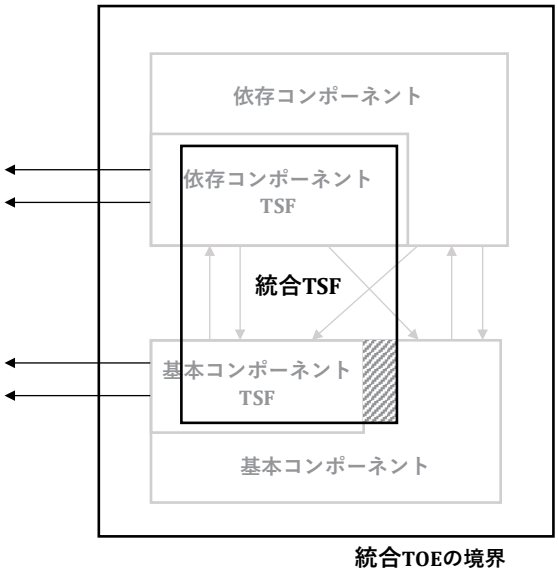


図B.2 — 依存コンポーネントの抽象概念

基本コンポーネントと依存コンポーネントの統合を考慮する際に、依存コンポーネントのTSFがSFRの実装をサポートするために基本コンポーネントのサービスを要求する場合は、サービスに対するインタフェースを定義する必要がある。このサービスが基本コンポーネントのTSFによって提供される場合、そのインタフェースは基本コンポーネントのTSFIであるべきであるため、基本コンポーネントの機能仕様で既に定義されているであろう。

ただし、依存コンポーネントのTSFによって呼び出されるサービスが基本コンポーネントのTSFから提供されない(つまり、そのサービスが基本コンポーネントの非TSF部分で実装されるか、場合によっては基本コンポーネントの非TOE部分(図B.3には示されていない)で実装される)場合は、そのサービスが基本コンポーネントのTSFによって仲介されない限り、そのサービスに関連する基本コンポーネントのTSFIである可能性は低い。依存コンポーネントから運用環境へのこれらのサービスに対するインタフェースは、依存コンポーネントの依存(ACO_REL)ファミリで考慮される。

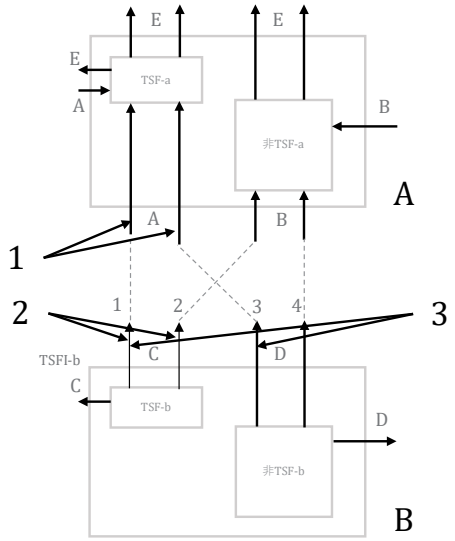
基本コンポーネントの非TSF部分は、依存コンポーネントのSFRを支援するために依存コンポーネントが基本コンポーネントに対して持つ依存性のために、統合TOEのTSFに取り込まれることもある。したがってこのような場合は、統合TOEのTSFが、コンポーネントのTSFを単純に合計したものよりも大きくなる。



図B.3 — 統合TOEの抽象概念

基本コンポーネントのTSFIが、基本コンポーネント評価では予期されなかった方法で呼び出される場合がある。このため、基本コンポーネントのTSFIを追加でテストする必要が生じる。

想定されるインタフェースについては、次の図(図B.4)及び補足説明で詳しく記述されている。



キー

A 依存コンポーネント-a
B 基本コンポーネント-b
1 ACO_REL (コンポーネント-a)
2 ADV_FSP (コンポーネント-b)
3 ACO_DEV (コンポーネント-b)

図B.4 — 統合コンポーネントのインタフェース

a) 「依存コンポーネント-a」に向かう矢印(A及びB) = コンポーネントは環境がサービス要求に応じることを期待する(依存コンポーネントから環境に対するコールに応答する)。

b) 「基本コンポーネント-b」から出ている矢印(C及びD) = 基本コンポーネントから環境に対して提供されるサービスのインタフェース。

c) コンポーネント間の破線 = 対を成すインタフェース間の通信のタイプ。

d) その他の(グレーの)矢印 = 特定の基準で記述されているインタフェース。

次に、単純化した例で考慮の必要な事項を説明する。

a (「依存コンポーネント-a」)及びb (「依存コンポーネント-b」)というコンポーネントがある。TSF-aから出ている矢印は、TSF-aが提供するサービスであるため、TSFI(a)とする。同様に、TSF-bから出ている矢印(「C」)はTSFI(b)とする。これらは、それぞれの機能仕様で詳しく説明されている。コンポーネント-aは、その環境からサービスを要求する。TSF(a)が必要とするサービスは「A」というラベルで示され、それ以外のサービス(TSF-aに関連しないもの)は「B」というラベルで示される。

コンポーネント-aがコンポーネント-bと結合されると、破線(対を成すインタフェース間の通信のタイプ)で示されるように、{コンポーネント-aが必要とするサービス}と{コンポーネント-bが提供するサービス}の組み合わせが4通り発生する。このいずれのセットも、特定の統合に存在する可能性がある:

— TSF-aがTSF-bの提供するサービスを必要とする(「A」が「C」に連結される)場合: これは直接的であり、「C」に関する詳細がコンポーネント-bのFSP内にある。この場合は、全てのインタフェースがコンポーネント-bの機能仕様で定義されているべきである。

— 非TSF-aがTSF-bの提供するサービスを必要とする(「B」が「C」に連結される)場合: これは直接的である(同じく「C」に関する詳細がコンポーネント-bのFSP内にある)が、セキュリティ上は重要でない。

— 非TSF-aが非TSF-bの提供するサービスを必要とする(「B」が「D」に連結される)場合: Dに関する詳細はわからないが、これらのインタフェースの使用はセキュリティに影響しないため、開発者にとってはこれらのインタフェースが統合上の課題になる可能性があるが、これらを評価で考慮する必要はない。

— TSF-aが非TSF-bの提供するサービスを必要とする(「A」が「D」に連結される)場合: これは、コンポーネント-aとコンポーネント-bで「セキュリティサービス」の意味するところが異なっている場合に生じる。例えば、コンポーネント-bはI&Aに関する主張を行っていない(そのSTにFIA SFRがない)が、コンポーネント-aはその環境が提供する認証を必要とする。「D」インタフェースに関する詳細はない(これらはTSFI(b)ではないため、コンポーネント-bのFSP内にない)。

注意: 上述のケースdで説明されている種類の相互作用が存在する場合、統合TOEのTSFは、TSF-a + TSF-b + 非TSF-bとなる。存在しない場合、統合TOEのTSFはTSF-a + TSF-bとなる。

図B.4のインタフェースタイプ2及び4は、統合TOEの評価に直接関連しない。インタフェース1及び3は、各種ファミリを適用する際に考慮される:

— 機能仕様(ADV_FSP) (コンポーネント-b用)は、Cインタフェースを記述する。

— 依存コンポーネントの依存(ACO_REL)は、Aインタフェースを記述する。

— 開発証拠(ACO_DEV)は、連結タイプ1のCインタフェースと、連結タイプ3のDインタフェースを記述する。

統合が適用される典型的な例は、下層のオペレーティングシステム(OS)に依存しているデータベース管理システム(DBMS)である。DBMSコンポーネントの評価中は、(評価で使用されている保証コンポーネントによって指示されている厳格さの度合いまで)そのDBMSのセキュリティ特性についての評定が実施される。例えば、TSF境界が識別され、機能仕様がTSFによって提供されるセキュリティサービスに対するインタフェースを記述するかどうかが評定される。TSFに関する追加情報(設計、アーキテクチャ、内部構造)が提供されたり、TSFがテストされたり、そのライフサイクルの側面とガイダンス証拠資料が評定されたりする可能性もある。

ただし、DBMS評価は、DBMSがOSに対して持つ依存性に関する証拠を要求しない。DBMSのSTは、前提条件の節でOSに関する前提条件を述べたり、環境の節でOSに対するセキュリティ対策方針を述べるケースが多い。DBMSのSTは、OSに対するSFRの観点から環境に対するそれらの対策方針の具体化までを行う場合もある。ただし、機能仕様、アーキテクチャ記述、又はDBMSのその他のADV証拠における詳細を反映するOSの仕様は存在しない。依存コンポーネントの依存(ACO_REL)は、その要件を満たす。

依存コンポーネントの依存(ACO_REL)は、サービスの提供のために基本コンポーネントにコールを行う依存TOEのインタフェースを記述する。これらは、基本コンポーネントが応答するインタフェースである。インタフェース記述は、依存コンポーネントの観点から提供される。

開発証拠(ACO_DEV)は、基本コンポーネントより提供される依存コンポーネントのサービス要求に応答するインタフェースを記述する。これらのインタフェースは、依存情報で識別される関連する依存コンポーネントのインタフェースにマッピングされる(このマッピングの完全性(記述された基本コンポーネントのインタフェースが全ての依存コンポーネントインタフェースを表すかどうか)は、ここでは検証されないが統合の根拠(ACO_COR)で検証される)。ACO_DEVの上位レベルで、インタフェースを提供するサブシステムが記述される。

基本コンポーネントで記述されない依存コンポーネントが必要とするインタフェースは全て、統合の根拠(ACO_COR)の根拠で報告される。根拠は、依存コンポーネントが依存している基本コンポーネントのインタフェースが、基本コンポーネント評価内で考慮されているかどうかについても報告する。基本コンポーネント評価で考慮されなかった全てのインタフェースについて、基本コンポーネントTSFでそのインタフェースを使用した場合の影響についての根拠が提供される。

附属書C
(参考)

保証コンポーネントの依存性の相互参照

7章及び9章から15章のコンポーネントに記述されている依存性は、保証コンポーネント間の直接的な依存性である。

次の保証コンポーネントに対する依存性の表は、それぞれの直接的、間接的、あるいはオプションの依存性を示す。ある保証コンポーネントが依存する個々のコンポーネントは、列に配置される。各保証コンポーネントは、行に配置される。表のセルにおける値は、列に書かれたコンポーネントが、行に書かれたコンポーネントによって、直接的に要求されるか(クロス「X」で表示)、間接的に要求されるか(ダッシュ「-」で表示)、あるいはオプション的に要求されるか(「O」で表示)を示す。文字が表示されていない場合、そのコンポーネントは他のコンポーネントに依存しない。

表C.1 — ADV: 開発クラスの依存性の表

ADV	ADV_FSP.1	ADV_FSP.2	ADV_FSP.3	ADV_FSP.4	ADV_FSP.5	ADV_FSP.6	ADV_IMP.1	ADV_TDS.1	ADV_TDS.3	ALC_CMC.5	ALC_CMS.1	ALC_DVS.2	ALC_LCD.1	ALC_TAT.1
ADV_ARC.1	X	–						X						
ADV_COMP.1														
ADV_FSP.1														
ADV_FSP.2		–						X						
ADV_FSP.3		–						X						
ADV_FSP.4		–						X						
ADV_FSP.5		–		–			X	X	–					–
ADV_FSP.6		–		–			X	X	–					–
ADV_IMP.1		–		–			–	–	X					X
ADV_IMP.2		–		–			–	–	X	X	–	–	–	X
ADV_INT.1		–		–			X	–	X					X
ADV_INT.2		–		–			X	–	X					X
ADV_INT.3		–		–			X	–	X					X
ADV_SPM.1				X				–						
ADV_TDS.1		X						–						
ADV_TDS.2		–	X					–						
ADV_TDS.3		–		X				–						
ADV_TDS.4		–		–	X		–	–	–					–
ADV_TDS.5		–		–	X		–	–	–					–
ADV_TDS.6		–		–		X	–	–	–					–

表C.2 — AGD: ガイダンス文書クラスの依存性の表

AGD	ADV_FSP.1

AGD_OPE.1	X
AGD_PRE.1	

表C.3 — ALC: ライフサイクルサポートクラスの依存性の表

ALC	ADV_FSP.2	ADV_FSP.4	ADV_IMP.1	ADV_TDS.1	ADV_TDS.3	ALC_CMS.1	ALC_CMS.3	ALC_DVS.1	ALC_DVS.2	ALC_LCD.1	ALC_TAT.1
ALC_CMC.1						X					
ALC_CMC.2						X					
ALC_CMC.3						X		X		X	
ALC_CMC.4						X		X		X	
ALC_CMC.5						X			X	X	
ALC_CMS.1											
ALC_CMS.2											
ALC_CMS.3											
ALC_CMS.4											
ALC_CMS.5											
ALC_COMP.1											
ALC_DEL.1											
ALC_DVS.1											
ALC_DVS.2											
ALC_FLR.1											
ALC_FLR.2											
ALC_FLR.3											
ALC_LCD.1											
ALC_LCD.2											
ALC_TAT.1	–	–	X	–	–						–
ALC_TAT.2	–	–	X	–	–						–
ALC_TAT.3	–	–	X	–	–						–
ALC_TDA.1											
ALC_TDA.2							X				
ALC_TDA.3	–	–	X	–	–		X				X

表C.4 — APE: プロテクションプロファイル評価クラスの依存性の表

APE	APE_ECD.1	APE_INT.1	APE_OBJ.2	APE_REQ.1	APE_SPD.1
APE_CCL.1	X	X		X	
APE_ECD.1					
APE_INT.1					
APE_OBJ.1					

APE_OBJ.2					X
APE_REQ.1	X				
APE_REQ.2	X		X		–
APE_SPD.1					

表C.5 — ACE: PP構成評価クラスの依存性の表

ACE	ACE_CCL.1	ACE_ECD.1	ACE_INT.1	ACE_MCO.1	ACE_OBJ.1	ACE_OBJ.2	ACE_REQ.1	ACE_REQ.2	ACE_SPD.1	APE_ECD.1
ACE_CCL.1		X	X		–		O	O	–	–
ACE_CCO.1	X	X	X	X	O	O	O	O	X	–
ACE_ECD.1										
ACE_INT.1										
ACE_MCO.1		–	X		O	O	O	O	X	–
ACE_OBJ.1										
ACE_OBJ.2									X	
ACE_REQ.1									X	X
ACE_REQ.2		X				X				
ACE_SPD.1										

表C.6 — ASE: セキュリティターゲット評価クラスの依存性の表

ASE	ADV_ARC.1	ADV_FSP.1	ADV_FSP.2	ADV_TDS.1	ASE_ECD.1	ASE_INT.1	ASE_OBJ.2	ASE_REQ.1	ASE_SPD.1
ASE_CCL.1					X	X		X	
ASE_COMP.1									
ASE_ECD.1									
ASE_INT.1									
ASE_OBJ.1									
ASE_OBJ.2									X
ASE_REQ.1					X				
ASE_REQ.2					X		X		–
ASE_SPD.1									
ASE_TSS.1		X			–	X		X	
ASE_TSS.2	X	–	–	–	–	X		X	

表C.7 — ATE: テストクラスの依存性の表

ATE	ADV_ARC.1	ADV_FSP.1	ADV_FSP.2	ADV_FSP.3	ADV_FSP.4	ADV_FSP.5	ADV_IMP.1	ADV_TDS.1	ADV_TDS.2	ADV_TDS.3	ADV_TDS.4	AGD_OPE.1	AGD_PRE.1	ALC_TAT.1	ATE_COV.1	ATE_FUN.1
ATE_COMP.1																

ATE_COV.1			X					–							–	X
ATE_COV.2			X					–							–	X
ATE_COV.3			X					–							–	X
ATE_DPT.1	X	–	–	–				–	X						–	X
ATE_DPT.2	X	–	–		–			–		X					–	X
ATE_DPT.3	X	–	–		–	–	–	–		–	X			–	–	X
ATE_DPT.4	X	–	–		–	–	X	–		–	X			–	–	X
ATE_FUN.1			–					–							X	–
ATE_FUN.2			–					–							X	–
ATE_IND.1		X										X	X			
ATE_IND.2		–	X					–				X	X		X	X
ATE_IND.3		–	–		X			–				X	X		X	X

表C.8 — AVA: 脆弱性評定クラスの依存性の表

AVA	ADV_ARC.1	ADV_FSP.1	ADV_FSP.2	ADV_FSP.3	ADV_FSP.4	ADV_IMP.1	ADV_TDS.1	ADV_TDS.2	ADV_TDS.3	AGD_OPE.1	AGD_PRE.1	ALC_TAT.1	ATE_COV.1	ATE_DPT.1	ATE_FUN.1
AVA_COMP.1															
AVA_VAN.1		X								X	X				
AVA_VAN.2	X	–	X				X			X	X				
AVA_VAN.3	X	–	–	–	X	X	–	–	X	X	X	–	–	X	–
AVA_VAN.4	X	–	–	–	X	X	–	–	X	X	X	–	–	X	–
AVA_VAN.5	X	–	–	–	X	X	–	–	X	X	X	–	–	X	–

表C.9 — ACO: 統合クラスの依存性の表

ACO	ACO_DEV.1	ACO_DEV.2	ACO_DEV.3	ACO_REL.1	ACO_REL.2	ALC_CMC.1	ALC_CMS.1
ACO_COR.1	X			X		X	–
ACO_CTT.1	X			X			
ACO_CTT.2		X		–	X		
ACO_DEV.1				X			
ACO_DEV.2				X			
ACO_DEV.3					X		
ACO_REL.1							
ACO_REL.2							
ACO_VUL.1	X			–			
ACO_VUL.2		X		–			
ACO_VUL.3			X		–		

参考文献

[1] ISO 10007:2017, *Quality management — Guidelines for configuration management*

---

[i] 【訳注】b)の例は、「TOE の構成の各段階」の例である。

[ii] 【訳注】c)の例は、転送される「構成要素」の例である。

[iii] 【訳注】e)の例は、「TOE の統合」の例である。

[iv] 【訳注】原文では”configuration management list”。

[v] 【訳注】APE_OBJ.1.2D について、原文では”security objectives rationale objectives”となっている。

[vi] 【訳注】ACE_CCL.1.4C 中の「CC パート 3」について、原文では”this document”であるが、このステートメントは別の文書(CEM)にも転記されており、その場合”this document”が何であるか明確でない。

[vii] 【訳注】ACE_CCO.1.10C 中の「CC パート 3」について、原文では”this document”であるが、このステートメントは別の文書(CEM)にも転記されており、その場合”this document”が何であるか明確でない。

[viii] 【訳注】10.8.3 中の i.の「基本コンポーネント」は原文では”base document”。

[ix] 【訳注】図 10 の「ALC_TDA:TOE 開発成果物」について、原文では”Development Artifacts”となっており、12.8 にあるような TOE の記述が無い。

[x] 【訳注】原文では ALC_CMC.3 の依存性と目的の間に、ALC_LCD の記述が挿入されてしまっている。

[xi] 【訳注】原文では ALC_CMC.4 の依存性と目的の間に、ALC_LCD の記述が挿入されてしまっている。

[xii] 【訳注】原文では ALC_CMC.5 の依存性と目的の間に、ALC_LCD の記述が挿入されてしまっている。

[xiii] 【訳注】12.5.3 の「脅威」について、原文では”threads”。

[xiv] 【訳注】12.7.3 の箇条書きについて、原文ではここで改行がある。

[xv] 【訳注】原文では”acceptance subsequent to internal transportations in Development security (ALC_DVS)”となっており、environment が記述されていない。

[xvi] 【訳注】図 12 の AVA_VAN について、原文では”Vulnerability assesment”(脆弱性評定)となっている。