Critères Communs pour l'évaluation de la sécurité des technologies de l'information

Partie 3 : Exigences d'assurance de sécurité

Août 1999

Version 2.1

CCIMB-99-033

Avant-propos

L'ISO (International Organisation for Standardisation, l'organisation internationale pour la normalisation) et l'IEC (International Electrotechnical Commission, la commission internationale électrotechnique) forment le système dédié à la normalisation mondiale. Les organisations nationales qui sont membres de l'ISO ou de l'IEC participent au développement des normes internationales par le biais de comités techniques établis par les organisations respectives pour traiter de domaines particuliers d'activités techniques. Les comités techniques de l'ISO et de l'IEC collaborent dans les domaines d'intérêt commun. D'autres organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'ISO et l'IEC, prennent également part au travail.

Dans le domaine des technologies de l'information, l'ISO et l'IEC ont établi un comité technique commun, l'ISO/IEC JTC 1. Les normes internationales provisoires (Draft International Standards) adoptées par le comité technique commun sont mises en circulation dans les organisations nationales pour être soumises à un vote. La publication comme norme internationale (International Standard) nécessite l'approbation d'au moins 75% des organisations nationales ayant voté.

La norme internationale ISO/IEC 15408 a été préparée par le comité technique commun ISO/IEC JTC 1, Technologies de l'Information, en collaboration avec le comité d'édition des critères communs (Common Criteria Implementation Board), une entité qui regroupe des membres des organisations commanditaires du projet Critères Communs. Le texte identique à la norme ISO/IEC 15408 est publié par les organisations commanditaires du projet Critères Communs sous le titre *Common Criteria for Information Technology Security Evaluation, version 2.0 (Critères Communs pour l'évaluation de la sécurité des technologies de l'information, version 2.0).* Des informations supplémentaires concernant le projet Critères Communs ainsi que les coordonnées des organisations commanditaires, sont fournies dans l'annexe A de la partie 1.

La norme ISO/IEC 15408, sous le titre général *Critères Communs pour l'évaluation de la sécurité des technologies de l'information,* comprend les parties suivantes :

Partie 1 : Introduction et modèle général

Partie 2 : Exigences fonctionnelles de sécurité

Partie 3 : Exigences d'assurance de sécurité

La présente NOTICE À CARACTÈRE LÉGAL a été introduite dans toutes les parties de la norme ISO/IEC 15408 sur demande :
Les sept organisations gouvernementales (collectivement dénommées les "organisations commanditaires du projet Critères Communs") citées ci-dessous et identifiées plus complètement dans l'Annexe A de la Partie 1, en tant que détentrices communes du copyright du document Critères Communs pour l'évaluation de la sécurité des technologies de l'information (Common Criteria for Information Technology Security Evaluation), version 2.0, comprenant les Parties 1 à 3 (appelé "CC 2.0"), accordent par la présente notice à l'ISO/IEC la licence non exclusive d'utilisation du document CC 2.0 pour le développement de la norme internationale ISO/IEC 15408. Cependant, les organisations commanditaires du projet Critères Communs conservent le droit d'utiliser, copier, distribuer ou modifier le document CC 2.0 quand elles le jugent bon.

- Allemagne :	*Bundesamt für Sicherheit in der Informationstechnik*
- Canada :	*Communications Security Establishment*
- Etats-Unis :	*National Institute of Standards and Technology*
- Etats-Unis :	*National Security Agency*
- France :	*Service Central de la Sécurité des Systèmes d'Information*
- Pays-Bas :	*Netherlands National Communications Security Agency*
- Royaume Uni :	*Communications-Electronics Security Group*

Table des matières

Part 3

Liste des figures

Liste des tableaux

1 Champ d'application

1 La présente partie 3 définit les exigences d'assurance des CC. Elle comprend les niveaux d'assurance de l'évaluation (EAL) qui définissent une échelle pour mesurer l'assurance, les composants individuels d'assurance à partir desquels sont composés les niveaux d'assurance et les critères pour l'évaluation des PP et des ST.

1.1 Organisation de la partie 3 des CC

2 Le chapitre 1 contient l'introduction et le paradigme de cette partie 3 des CC.

3 Le chapitre 2 décrit la structure adoptée pour la présentation des classes, familles, composants d'assurance et des niveaux d'assurance de l'évaluation ainsi que leurs relations. Il caractérise également les classes et familles d'assurance qui se trouvent dans les chapitres 8 à 14.

4 Les chapitres 3, 4 et 5 fournissent une brève introduction aux critères d'évaluation des PP et ST, suivie d'explications détaillées sur les familles et les composants qui sont utilisés pour ces évaluations.

5 Le chapitre 6 donne les définitions détaillées des EAL.

6 Le chapitre 7 fournit une brève introduction aux classes d'assurance ; il est suivi des chapitres 8 à 14 qui donnent les définitions détaillées de ces classes.

7 Les chapitres 15 et 16 fournissent une brève introduction aux critères d'évaluation pour la maintenance de l'assurance, suivie des définitions détaillées des familles et composants associés à la maintenance.

8 L'annexe A contient un résumé des dépendances entre les composants d'assurance.

9 L'annexe B contient les références croisées entre les EAL et les composants d'assurance.

1.2 Paradigme de l'assurance selon les CC

10 Le but de cette section est de documenter la philosophie qui étaye la manière adoptée par les CC pour aborder l'assurance. La compréhension de cette section permettra au lecteur d'appréhender l'argumentaire relatif aux exigences d'assurance de la partie 3 des CC.

1.2.1 Philosophie des CC

11 Selon la philosophie des CC, les menaces qui pèsent sur la sécurité et les règles en terme de politique de sécurité organisationnelle devraient être exprimées clairement

et il devrait être démontré que les mesures de sécurité proposées sont suffisantes pour atteindre le but prévu.

12 De plus, des mesures devraient être adoptées pour réduire la probabilité d’occurrence de vulnérabilités, ainsi que la capacité d’activer (i.e. d’exploiter intentionnellement ou bien de déclencher) de manière non intentionnelle une vulnérabilité, et enfin pour réduire l’étendue des dommages qui pourraient en résulter. En complément, des mesures qui facilitent l’identification ultérieure des vulnérabilités ainsi que leur élimination ou leur atténuation, ou la notification qu’une vulnérabilité a été exploitée ou déclenchée, devraient être adoptées.

1.2.2 Approche de l’assurance

13 Le principe de base des CC consiste à fournir une assurance basée sur une évaluation (une investigation active) du produit ou du système TI auquel on doit accorder sa confiance. L’évaluation a été le moyen traditionnel de fournir l’assurance et constitue la base sur laquelle reposent les critères d’évaluation qui ont précédé les CC. Les CC adoptent le même principe de base en mettant en commun les approches existantes. Les CC proposent de faire mesurer, par des experts en évaluation, la validité de la documentation et du produit ou système TI qui en résulte, avec un accent croissant sur le champ d’application, la profondeur et la rigueur.

14 Les CC n’excluent ni ne commentent la valeur relative d'autres moyens d’obtenir l’assurance. Les recherches se poursuivent en ce qui concerne les solutions alternatives pour acquérir l’assurance. Lorsque des démarches alternatives arriveront à maturité grâce à ces recherches, l’opportunité de leur inclusion dans les CC sera considérée, ces derniers étant structurés de façon à permettre leur introduction future.

1.2.2.1 Importance des vulnérabilités

15 On suppose qu’il existe des agents de menace qui chercheront activement à exploiter les occasions de violer les politiques de sécurité, aussi bien pour s’assurer des gains illicites que pour commettre des actions bien intentionnées, mais néanmoins non sûres. Les agents de menace peuvent aussi déclencher accidentellement des vulnérabilités de sécurité, causant ainsi des dommages à l’organisation concernée. Le besoin de traiter les informations sensibles et le manque de disponibilité de produits ou de systèmes auxquels on peut faire suffisamment confiance, entraînent l’existence de risques significatifs, dus aux défaillances dans les TI. Il est par conséquent vraisemblable que des violations de la sécurité des TI puissent conduire à des pertes importantes.

16 Des violations de la sécurité des TI surviennent pendant l’utilisation des TI pour des besoins professionnels, par l’exploitation intentionnelle ou le déclenchement accidentel de vulnérabilités.

17 Des mesures devraient être prises pour empêcher l’apparition de vulnérabilités dans les produits et systèmes TI. Dans la mesure du possible, les vulnérabilités devraient être :

a) éliminées — ce qui signifie que des mesures actives devraient être prises pour démasquer, puis supprimer ou neutraliser, toutes les vulnérabilités susceptibles d'être mises en œuvre ;

b) minimisées — ce qui signifie que des mesures actives devraient être prises pour réduire l'impact potentiel de l'exploitation d'une vulnérabilité quelconque à un niveau résiduel acceptable ;

c) contrôlées — ce qui signifie que des mesures devraient être prises pour garantir que toute tentative pour exploiter une vulnérabilité résiduelle sera détectée de façon à pouvoir prendre des mesures pour limiter les dégâts.

1.2.2.2 Origine des vulnérabilités

18 Les vulnérabilités peuvent apparaître par l'intermédiaire de défaillances dans :

a) les exigences — ce qui signifie qu'un produit ou système TI peut inclure toutes les fonctions et caractéristiques exigées et, malgré cela, contenir des vulnérabilités qui le rendent inadapté ou inefficace pour la sécurité ;

b) le développement — ce qui signifie qu'un produit ou système TI ne satisfait pas à ses spécifications ou que des vulnérabilités ont été introduites à cause de l'utilisation de méthodes de développement médiocres ou de choix de conception incorrects ;

c) l'exploitation — ce qui signifie qu'un produit ou système TI a été développé correctement sur la base de spécifications correctes, mais que des vulnérabilités ont été introduites à cause de contrôles non appropriés en exploitation.

1.2.2.3 Assurance procurée par les CC

19 L'assurance est le fondement de la confiance dans le fait qu'un produit ou système TI satisfait à ses objectifs de sécurité. L'assurance peut être déduite à partir de sources telles que des affirmations non prouvées, une expérience préalable appropriée ou bien une expérience spécifique. Néanmoins, les CC procurent l'assurance par l'intermédiaire d'investigations actives qui consistent en une évaluation du produit ou système TI pour déterminer ses propriétés de sécurité.

1.2.2.4 Assurance obtenue par l'évaluation

20 L'évaluation est le moyen traditionnel pour procurer l'assurance et constitue la base de l'approche CC. Les techniques d'évaluation peuvent comprendre les éléments suivants, sans que cette liste soit exhaustive :

a) l'analyse et la vérification du ou des processus et procédures ;

b) la vérification que le ou les processus et procédures sont appliqués ;

c) l’analyse des correspondances entre les représentations de conception de la TOE ;

d) l’analyse de chaque représentation de conception de la TOE par rapport aux exigences ;

e) la vérification de preuves ;

f) l’analyse de guides ;

g) l’analyse de tests fonctionnels développés et des résultats obtenus ;

h) les tests fonctionnels indépendants ;

i) l’analyse de vulnérabilités (y compris les hypothèses sur les anomalies) ;

j) les tests de pénétration.

1.2.3 L’échelle CC d’assurance de l’évaluation

21 La philosophie des CC affirme qu’une assurance plus élevée résulte de l’application d’efforts plus importants dans l’évaluation et que le but est de produire l’effort minimum exigé pour procurer le niveau d’assurance nécessaire. Le niveau d’effort croissant repose sur les éléments suivants :

a) le champ d’application — ce qui signifie que l’effort est plus important parce qu’une plus grande partie du produit ou système TI est concernée ;

b) le degré d’approfondissement — ce qui signifie que l’effort est plus important parce qu’il s’exerce à un niveau de détail plus fin sur la conception et l’implémentation ;

c) la rigueur — ce qui signifie que l’effort est plus important parce qu’il s’applique d’une manière plus structurée, plus formelle.

2 Exigences d’assurance de sécurité

2.1 Structures

22 Les sections suivantes décrivent les structures utilisées pour représenter les classes, les familles, les composants d’assurance, et les EAL, ainsi que les relations entre ces entités.

23 La figure 2.1 illustre les exigences d’assurance définies dans la présente partie 3 des CC. Il est à noter que le regroupement le plus abstrait d’exigences d’assurance est désigné par le terme “classe”. Chaque classe contient des familles d’assurance, qui contiennent des composants d’assurance, qui à leur tour contiennent des éléments d’assurance. Les classes et les familles sont utilisées pour fournir une taxinomie pour classer les exigences d’assurance, tandis que les composants sont utilisés pour spécifier les exigences d’assurance dans un PP ou une ST.

2.1.1 Structure d’une classe

24 La figure 2.1 illustre la structure d’une classe d’assurance.

2.1.1.1 Nom de la classe

25 À chaque classe d’assurance est assigné un nom unique. Le nom indique les thèmes couverts par cette classe d’assurance.

26 Une abréviation unique du nom de la classe d’assurance est également fournie. Cela constitue le moyen principal de faire référence à la classe d’assurance. La convention adoptée est de commencer par la lettre “A” suivie de deux lettres reliées au nom de la classe.

2.1.1.2 Introduction de la classe

27 Pour chaque classe d’assurance, une section introductive décrit la composition de la classe et explique le but recherché par la classe.

2.1.1.3 Familles d’assurance

28 Chaque classe d’assurance contient au moins une famille d’assurance. La structure des familles d’assurance est décrite dans la section suivante.

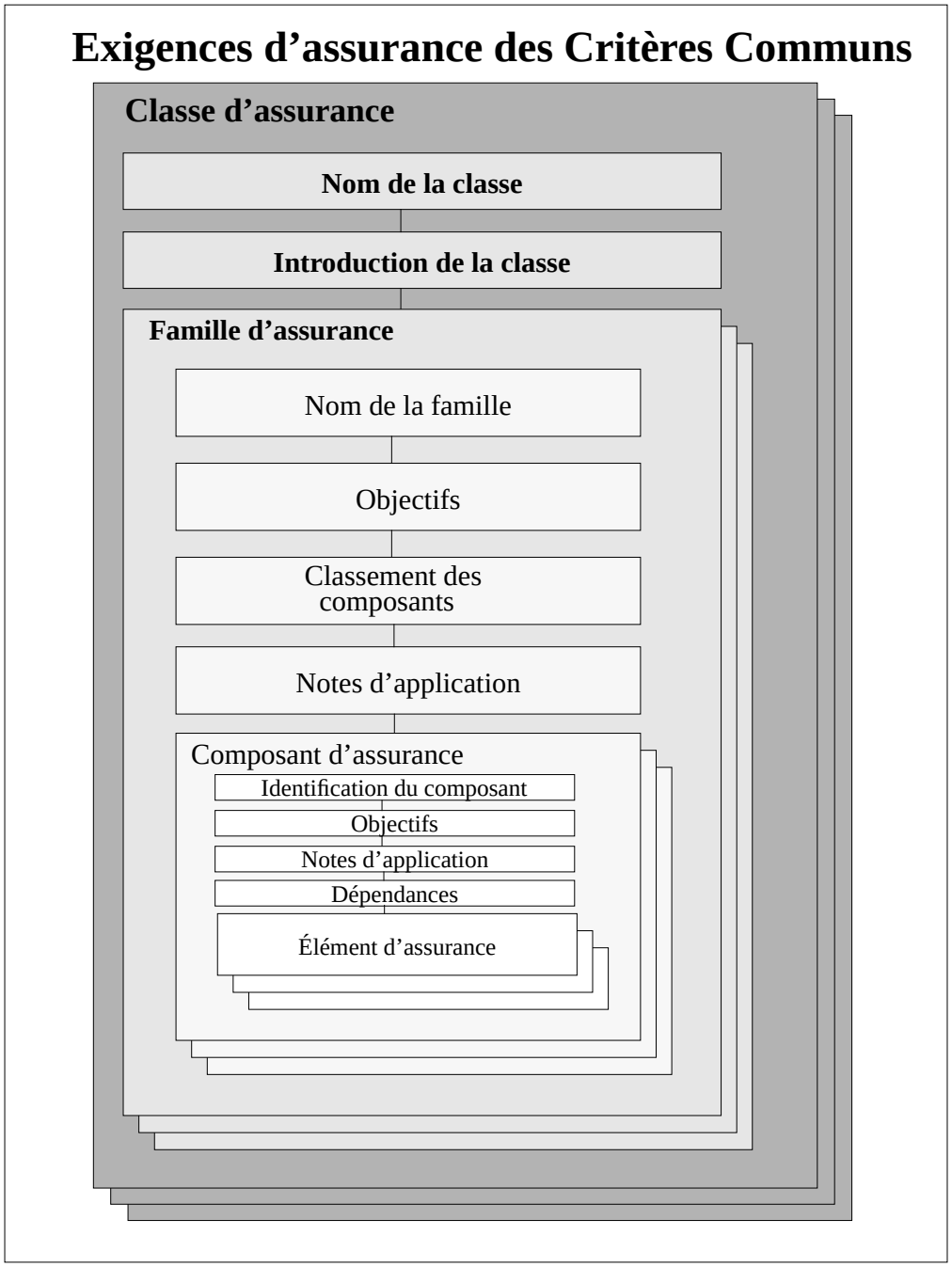

Figure 2.1 - Hiérarchie classe/famille/composant/élément d’assurance.

2.1.2 Structure d’une famille d’assurance

29 La figure 2.1 illustre la structure d’une famille d’assurance.

2.1.2.1 Nom de la famille

30 À chaque classe d'assurance est assigné un nom unique. Le nom donne des informations descriptives relatives aux thèmes couverts par la famille d'assurance. Chaque famille d'assurance est placée dans la classe d'assurance qui contient d'autres familles ayant le même but.

31 Une abréviation unique du nom de la famille d'assurance est également fournie. Cela constitue le moyen principal de faire référence à la famille d'assurance. La convention adoptée est d'utiliser l'abréviation du nom de la classe, suivie par le caractère "_" (underscore), et par trois lettres reliées au nom de la famille.

2.1.2.2 Objectifs

32 La section "objectifs" de la famille d'assurance présente le but de la famille d'assurance.

33 Cette section décrit les objectifs que la famille est censée couvrir, en particulier ceux qui ont un lien avec le paradigme de l'assurance selon les CC. La description de la famille d'assurance est faite à un niveau général. Les détails spécifiques exigés pour les objectifs sont donnés dans les composants d'assurance.

2.1.2.3 Classement des composants

34 Chaque famille d'assurance contient un ou plusieurs composants d'assurance. Cette section de la famille d'assurance décrit les composants disponibles et explique leurs différences. Le but principal est de différencier les composants d'assurance, une fois déterminé le fait que la famille d'assurance constitue une partie nécessaire ou utile des exigences d'assurance pour un PP ou une ST.

35 Lorsqu'une famille d'assurance contient plus d'un composant, une relation d'ordre est définie entre eux, et un argumentaire explique les relations entre ces composants. L'argumentation est faite en termes de champ d'application, de degré d'approfondissement ou de rigueur.

2.1.2.4 Notes d'application

36 La section "notes d'application" de la famille d'assurance, quand elle est présente, contient des informations supplémentaires sur cette famille d'assurance. Ces informations présentent de l'intérêt en particulier pour les utilisateurs de la famille d'assurance (e.g. les auteurs d'un PP ou d'une ST, les concepteurs de TOE, les évaluateurs). La présentation est faite de manière informelle et couvre, par exemple, les mises en garde relatives aux limitations d'utilisation et aux domaines où une attention spécifique peut être requise.

2.1.2.5 Composants d'assurance

37 Chaque famille d'assurance comprend au moins un composant d'assurance. La structure des composants d'assurance est donnée dans la section suivante.

2.1.3 Structure d’un composant d’assurance

38 La figure 2.2 illustre la structure d’un composant d’assurance.

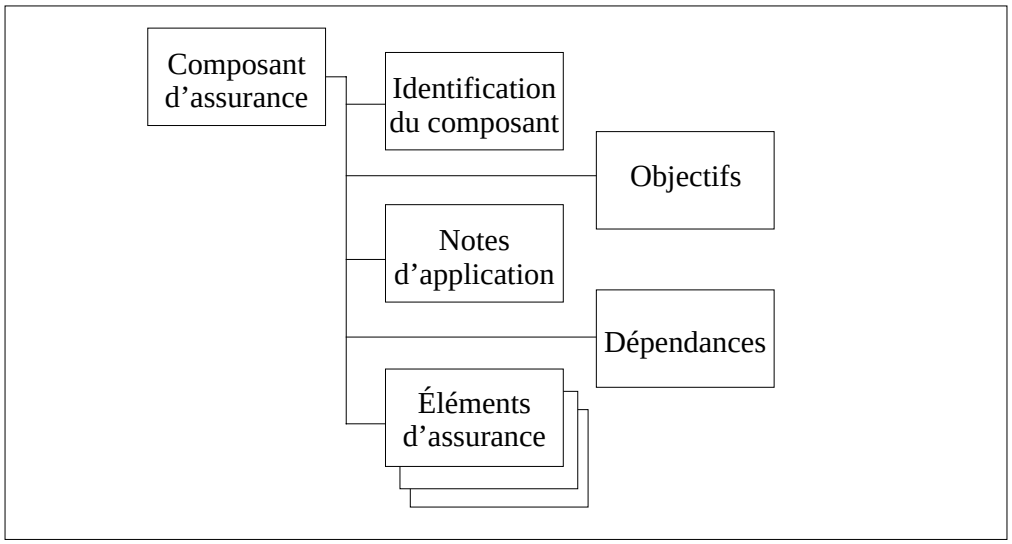

Figure 2.2 - Structure d’un composant d’assurance

39 Les relations entre les composants d’une famille sont mises en évidence par l’utilisation de caractères gras. Toutes les parties des exigences qui sont nouvelles, enrichies ou modifiées par rapport aux exigences du composant précédent d’une même hiérarchie sont présentées en caractères gras. La même convention (caractères gras) est aussi utilisée pour les dépendances.

2.1.3.1 Identification du composant

40 La section “identification d’un composant” fournit les informations descriptives nécessaires pour identifier, classer, enregistrer et référencer un composant.

41 À chaque composant d’assurance est assigné un nom unique. Le nom donne des informations descriptives relatives aux thèmes couverts par le composant d’assurance. Chaque composant d’assurance est placé dans la famille d’assurance qui partage ses objectifs de sécurité.

42 Une abréviation unique du nom du composant d’assurance est également donnée. Cela constitue le moyen principal de faire référence à un composant d’assurance. La convention adoptée est d’utiliser l’abréviation du nom de la famille, suivie d’un point puis d’un caractère numérique. Au sein de chaque famille, les caractères numériques sont assignés séquentiellement aux composants de la famille en commençant par le chiffre 1.

2.1.3.2 Objectifs

43 La section "objectifs" d'un composant d'assurance, quand elle est présente, contient des objectifs spécifiques pour le composant d'assurance concerné. Pour les composants d'assurance qui contiennent cette section, celle-ci présente le but spécifique du composant et une explication plus détaillée des objectifs.

2.1.3.3 Notes d'application

44 La section "notes d'application" d'un composant d'assurance, quand elle est présente, contient des informations supplémentaires pour faciliter l'utilisation du composant.

2.1.3.4 Dépendances

45 Des dépendances entre composants d'assurance apparaissent quand un composant ne se suffit pas à lui-même et dépend de la présence d'un autre composant.

46 Chaque composant d'assurance fournit une liste complète des dépendances vers d'autres composants d'assurance. Certains composants peuvent indiquer "pas de dépendances", exprimant par là qu'aucune dépendance n'a été identifiée. Les composants dont un composant dépend peuvent eux-mêmes dépendre d'autres composants.

47 La liste des dépendances d'un composant identifie l'ensemble minimum des composants d'assurance dont il dépend. Les composants qui sont hiérarchiques à un composant figurant dans la liste des dépendances peuvent également être utilisés pour satisfaire la dépendance.

48 Dans des situations spécifiques, les dépendances qui sont indiquées pourraient se révéler inapplicables. L'auteur du PP ou de la ST, en donnant l'argumentaire qui explique pourquoi une dépendance donnée n'est pas applicable, peut choisir de ne pas satisfaire cette dépendance.

2.1.3.5 Éléments d'assurance

49 Un ensemble d'éléments d'assurance est fourni pour chaque composant d'assurance. Un élément d'assurance est une exigence de sécurité qui, si elle était encore décomposée, n'aboutirait pas à un résultat d'évaluation significatif. C'est la plus petite exigence de sécurité reconnue dans les CC.

50 Chaque élément d'assurance est identifié comme appartenant à l'un des trois ensembles d'éléments d'assurance suivants :

a) tâches du développeur : il s'agit des activités qui doivent être réalisées par le développeur. Cet ensemble de tâches est précisé par des éléments de preuve référencés dans l'ensemble d'éléments suivant. Les exigences relatives aux tâches du développeur sont identifiées en rajoutant la lettre "D" au numéro de l'élément.

b) contenu et présentation des éléments de preuve : il s'agit des éléments de preuve exigés, de ce qu'ils doivent démontrer et des informations qu'ils doivent apporter. Les exigences pour le contenu et la présentation des éléments de preuve sont identifiées en ajoutant la lettre "C" au numéro de l'élément.

c) tâches de l'évaluateur : il s'agit des activités qui doivent être réalisées par l'évaluateur. Cet ensemble de tâches inclut de façon explicite la confirmation que les exigences prescrites dans le contenu et la présentation des éléments de preuve ont été satisfaites. Il inclut également les tâches et analyses qui doivent être exécutées en plus de celles déjà réalisées par le développeur. Les actions implicites de l'évaluateur doivent également être réalisées en réponse aux éléments de tâches du développeur qui ne sont pas couvertes par le contenu et la présentation des exigences de preuve. Les exigences relatives aux tâches de l'évaluateur sont identifiées en ajoutant la lettre "E" au numéro de l'élément.

51 Les tâches du développeur ainsi que le contenu et la présentation des éléments de preuve définissent les exigences d'assurance qui sont utilisées pour représenter les responsabilités d'un développeur relatives à la démonstration de l'assurance dans les fonctions de sécurité de la TOE. En satisfaisant à ces exigences, le développeur peut accroître la confiance dans le fait que la TOE satisfait aux exigences fonctionnelles et d'assurance d'un PP ou d'une ST.

52 Les tâches de l'évaluateur définissent les responsabilités de l'évaluateur en ce qui concerne les deux aspects de l'évaluation. Le premier aspect consiste à valider le PP ou la ST, en accord avec les classes APE et ASE des chapitres 4 et 5. Le second aspect consiste à vérifier la conformité de la TOE à ses exigences fonctionnelles et d'assurance. En démontrant que le PP ou la ST est valide et que les exigences sont satisfaites par la TOE, l'évaluateur renforce la confiance dans le fait que la TOE satisfera à ses objectifs de sécurité.

53 Les tâches du développeur, le contenu et à la présentation des éléments de preuve ainsi que les actions explicites de l'évaluateur identifient l'effort que l'évaluateur devra consacrer à la vérification des annonces de sécurité figurant dans la ST de la TOE.

2.1.4 Éléments d'assurance

54 Chaque élément représente une exigence à satisfaire. L'énoncé des exigences se veut clair, concis et sans ambiguïté. Par conséquent, il n'y figure pas de phrases complexes : chaque exigence distincte est formulée sous la forme d'un élément individuel.

55 Les éléments ont été rédigés avec la signification usuelle du dictionnaire pour les termes employés, plutôt que des abréviations correspondant au regroupement de termes prédéfinis, amenant des exigences implicites. En conséquence, les éléments sont rédigés sous forme d'exigences explicites, sans *termes réservés*.

56 Contrairement à la partie 2 des CC, aucune opération d'affectation ou de sélection n'est pertinente pour les éléments contenus dans la partie 3 des CC ; cependant, il est possible d'effectuer des raffinements sur des éléments de la partie 3, si nécessaire.

2.1.5 Structure d'un EAL

57 La figure 2.3 illustre les EAL et la structure associée, définie dans la présente partie 3. Cette figure montre le contenu des composants d'assurance ; il est cependant prévu que ces informations soient incluses dans un EAL par référence aux composants définis dans les CC.

2.1.5.1 Nom de l'EAL

58 À chaque EAL est assigné un nom unique. Le nom donne des informations descriptives relatives au but de l'EAL.

59 Une abréviation unique du nom de l'EAL est également donnée. Cela constitue le moyen principal de faire référence à l'EAL.

2.1.5.2 Objectifs

60 La section "objectifs" de l'EAL présente le but de l'EAL.

2.1.5.3 Notes d'application

61 La section "notes d'application" de l'EAL, quand elle est présente, contient des informations présentant un intérêt particulier pour les utilisateurs de l'EAL (e.g. les auteurs d'un PP ou d'une ST, les concepteurs de TOE visant cet EAL, les évaluateurs). La présentation est faite de manière informelle et couvre, par exemple, les avertissements relatifs aux limitations d'utilisation et les domaines où une attention spécifique peut être requise.

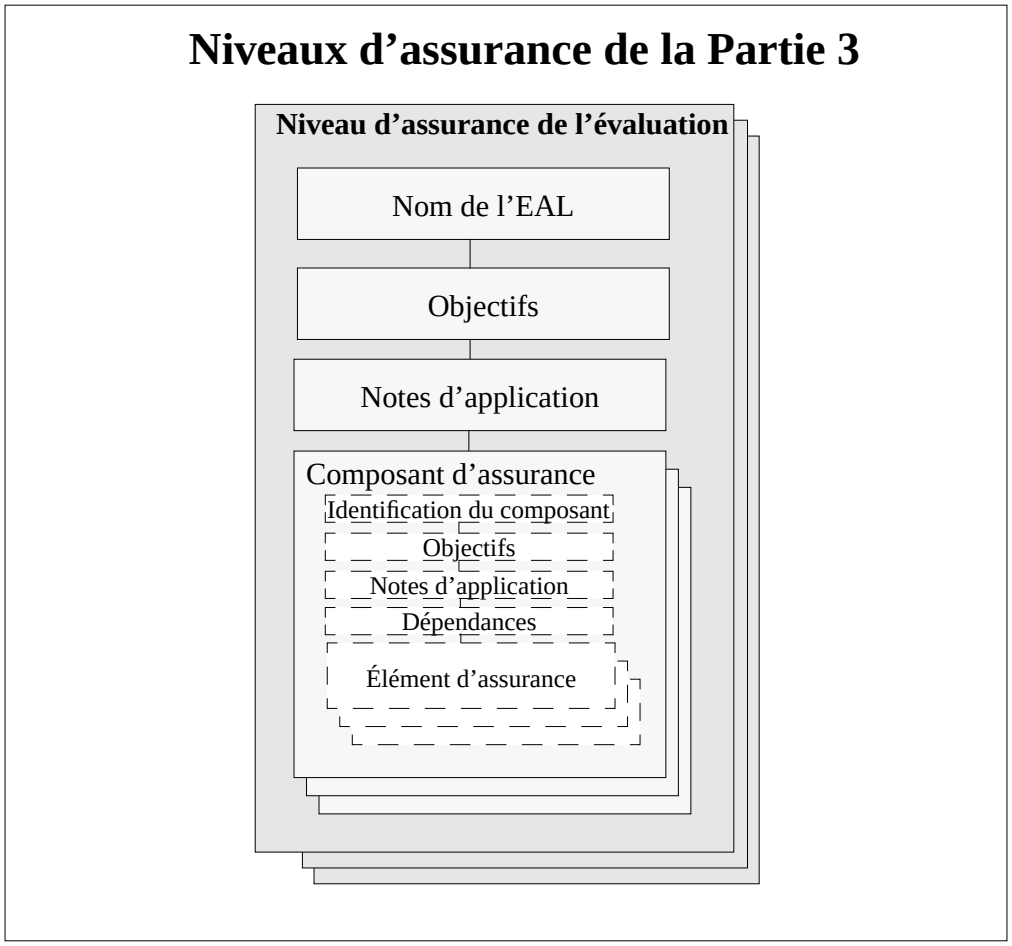

Figure 2.3 - Structure d’un EAL

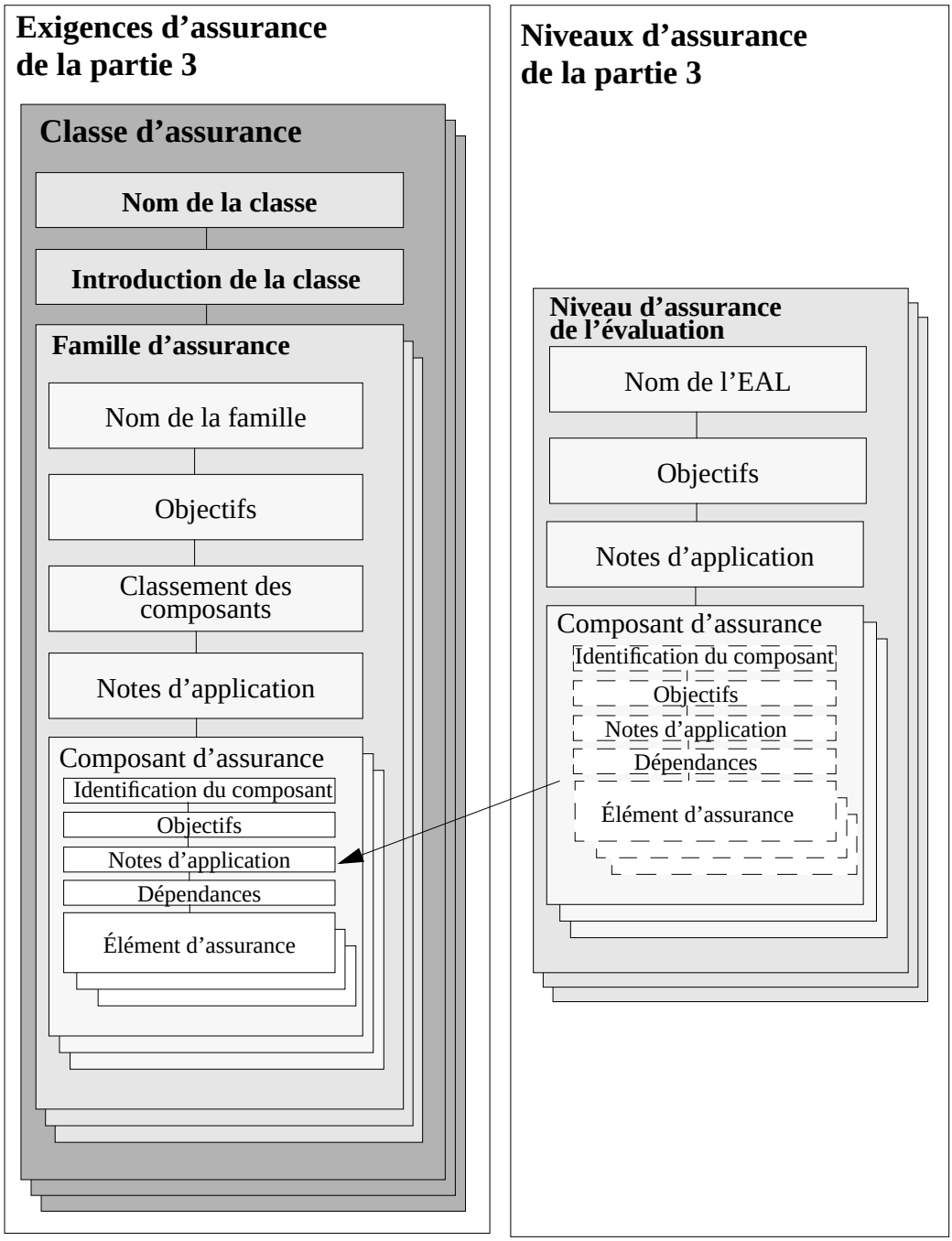

Figure 2.4 - Correspondance entre exigences et niveaux d’assurance

2.1.5.1 Composants d’assurance

62 Un ensemble de composants d’assurance a été sélectionné pour chaque EAL.

63 Un niveau d’assurance plus élevé que celui procuré par un EAL donné peut être obtenu :

a) en incluant des composants d’assurance supplémentaires pris dans d’autres familles d’assurance,

b) ou en remplaçant un composant d’assurance par un composant d’assurance de niveau plus élevé pris dans la même famille d’assurance.

2.1.6 Relation entre exigences et niveaux d’assurance

64 La figure 2.4 illustre la relation entre les exigences d’assurance et les niveaux de l’assurance définis dans la partie 3 des CC. Alors que les composants d’assurance se décomposent encore en éléments d’assurance, les éléments d’assurance ne peuvent pas être référencés individuellement par les niveaux d’assurance. Il est à noter que la flèche dans la figure représente une référence d’un EAL vers un composant d’assurance à l’intérieur de la classe où il est défini.

2.2 Taxinomie d’un composant

65 La présente partie 3 contient des classes de familles et composants qui sont groupés selon l'assurance qu'ils procurent. Au début de la présentation de chaque classe figure un diagramme qui indique les familles contenues dans cette classe et les composants contenus dans chaque famille.

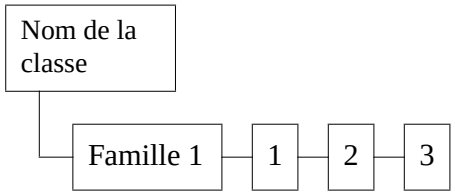

Figure 2.5 - Exemple de diagramme de décomposition d’une classe

66 Dans la figure 2.5 ci-dessus, la classe qui est montrée contient une seule famille. La famille contient trois composants qui ont des relations hiérarchiques linéaires (i.e. le composant 2 est plus exigent que le composant 1, en terme de tâches spécifiques, d’éléments de preuve spécifiques, ou de rigueur dans les tâches ou dans les éléments de preuve). Les familles d’assurance dans la présente partie 3 présentent toutes des relations hiérarchiques linéaires, bien que la linéarité ne constitue pas un critère obligatoire pour les familles d’assurance qui pourraient être ajoutées dans le futur.

2.3 Structure de la classe “critères d’évaluation d’un profil de protection et d’une cible de sécurité”

67 Les exigences pour l’évaluation d’un profil de protection et d’une cible de sécurité sont traitées comme des classes d’assurance et sont présentées en utilisant une

structure semblable à celle utilisée pour les autres classes d'assurance, décrites ci-dessous. Une différence notable concerne l'absence d'une section "classement des composants" dans la description des familles associées. La raison en est que chaque famille ne comporte qu'un seul composant et que, par conséquent, aucun classement n'a eu lieu.

68 Les tableaux 3.1, 3.2, 3.3 et 3.4 contenus dans le chapitre 3 de la présente partie 3 résument, à la fois pour les classes APE et ASE, leurs familles constituantes et leurs abréviations respectives. Des résumés narratifs concernant les familles de la classe APE figurent dans les sections B.2.2 à B.2.6 de l'annexe B de la partie 1 des CC, tandis que les résumés narratifs concernant les familles de la classe ASE se trouvent dans les sections C.2.2 à C.2.8 de l'annexe C de la partie 1 des CC.

2.4 Utilisation des termes dans la partie 3

69 La liste suivante contient des termes qui sont utilisés avec un sens précis dans la présente partie 3. Ils ne méritent pas d'être incorporés dans le glossaire car ce sont à l'origine des termes usuels et leur utilisation, bien que restreinte par les explications données ci-après, est conforme aux définitions du dictionnaire. Cependant, les explications de ces termes ont été utilisées comme guide pour le développement de la présente partie 3 et devraient être utiles pour la compréhension générale.

70 **Cohérent (consistent)** — Ce terme décrit une relation entre deux entités ou plus et indique qu'il n'y a pas de contradictions apparentes entre ces entités.

71 Qui possède une **cohérence interne (internally consistent)** — Il n'y a pas de contradictions apparentes entre les aspects d'une entité. Pour une documentation, cela signifie qu'il n'y a pas d'énoncés dans la documentation pouvant être considérés comme contradictoires entre eux.

72 **Complet (complete)** — Toutes les parties nécessaires d'une entité ont été fournies. Pour une documentation, cela signifie que toutes les informations pertinentes figurent dans la documentation, à un niveau de détail tel qu'aucune explication supplémentaire n'est exigée au niveau d'abstraction concerné.

73 **Confirmer (confirm)** — Ce terme est utilisé pour indiquer que quelque chose doit être revu en détail et qu'une détermination indépendante de sa suffisante doit être réalisée. Le niveau de rigueur exigé dépend de la nature du sujet. Ce terme s'applique seulement aux tâches de l'évaluateur.

74 **Contrer (counter** (verb)) — Ce terme est typiquement utilisé dans le contexte où un objectif de sécurité contre une menace particulière, mais n'indique pas nécessairement qu'il en résulte que la menace est complètement éradiquée.

75 **Contrôler (verify)** — Ce terme est similaire au terme "confirmer", mais possède des connotations plus rigoureuses. Utilisé dans le contexte des tâches de l'évaluateur, ce terme indique qu'un effort indépendant est exigé de la part de l'évaluateur.

76 **Décrire (describe)** — Ce terme exige que certains détails spécifiques à une entité soient fournis.

77 **Démontrer (demonstrate)** — Ce terme fait référence à une analyse menant à une conclusion, ce qui est moins rigoureux qu’une “preuve (proof)”.

78 **Déterminer (determine)** — Ce terme exige que soit menée une analyse indépendante, avec pour objectif de parvenir à une certaine conclusion. L’utilisation de ce terme diffère de celle de “confirmer (confirm)” ou de “contrôler (verify)”, puisque ces autres termes impliquent qu’une analyse a déjà été effectuée et doit être revue, alors que l’utilisation de “déterminer (determine)” implique que soit menée une analyse véritablement indépendante, en général sans qu'aucune analyse préalable n'ait été effectuée.

79 **Exhaustif (exhaustive)** — Ce terme est utilisé dans les CC pour ce qui concerne la conduite d’une analyse ou d’une autre activité. Il a une signification proche de celle de “systématique (systematic)” mais considérablement renforcée, du fait qu’il indique non seulement qu’une approche méthodique a été adoptée pour effectuer l’analyse ou l’activité concernée selon un plan non ambigu, mais également que le plan qui a été suivi est suffisant pour garantir que toutes les voies possibles ont été explorées.

80 **Expliquer (explain)** — Ce terme diffère à la fois de “décrire” et de “démontrer”. Il est censé répondre à la question “Pourquoi?” sans essayer réellement de prétendre que la ligne de conduite qui a été choisie était nécessairement optimale.

81 **Garantir (ensure)** — Ce terme, utilisé seul, implique une forte relation de cause à effet entre une action et ses conséquences. Ce terme est précédé typiquement de l’expression “aide à (helps)”, qui indique que la conséquence n’est pas absolument certaine, sur la base de cette seule action.

82 **Intelligible (coherent)** — Une entité est logiquement ordonnée et possède une signification discernable. Pour une documentation, cela concerne à la fois le texte lui-même et la structure du document, à savoir s’il est compréhensible par l’audience visée.

83 **Justification (justification)** — Ce terme fait référence à une analyse aboutissant à une conclusion, mais il se veut plus rigoureux que le terme “démonstration” car il exige une rigueur significative pour expliquer très soigneusement et complètement chaque étape d’un argument logique.

84 **Prouver (prove)** — Ce terme fait référence à une analyse formelle au sens mathématique. Il a une signification de rigueur totale dans tous les cas. Typiquement, “prouver” est utilisé lorsqu’on désire montrer la correspondance entre deux représentations de la TSF avec un niveau élevé de rigueur.

85 Qui se **soutiennent mutuellement (mutually supportive)** — Ce terme décrit une relation au sein d’un groupe d’entités et indique que les entités possèdent des propriétés qui n’entrent pas en conflit et qui peuvent aider les autres entités à remplir leurs tâches. Il n’est pas nécessaire de déterminer que chaque entité

individuelle en question apporte une aide directe à d'autres entités de ce groupe ; en fait, c'est une détermination plus générale qui est faite.

86 **Spécifier (specify)** — Ce terme est utilisé dans le même contexte que "décrire", mais est censé être plus rigoureux et plus précis. Il ressemble beaucoup à "définir".

87 **Tracer (trace** (verb)) — Ce terme est utilisé pour indiquer qu'une correspondance informelle est exigée entre deux entités avec seulement un niveau de rigueur minimal.

88 **Vérifier (check)** — Ce terme a une signification similaire à "confirmer (confirm)" ou "contrôler (verify)", mais il se veut moins rigoureux . Ce terme requiert qu'une décision rapide soit prise par l'évaluateur, exigeant seulement une analyse superficielle ou peut-être même aucune analyse.

2.5 Décomposition de l'assurance

89 Les classes, les familles d'assurance et les abréviations utilisées pour chaque famille sont présentées dans le tableau 2.1.

2.6 Vue d'ensemble des classes et des familles d'assurance

90 Les sections suivantes résument les classes et familles d'assurance présentées dans les chapitres 8 à 14. Ces résumés des classes et familles sont présentés dans le même ordre que leur ordre d'apparition dans les chapitres 8 à 14.

2.6.1 Classe ACM : Gestion de configuration

91 La classe "Gestion de configuration (CM)" aide à garantir que l'intégrité de la TOE est préservée, en exigeant une discipline et des contrôles dans les processus de raffinement et de modification de la TOE ainsi que d'autres informations associées. La CM empêche les modifications, additions ou suppressions non autorisées de la TOE, procurant ainsi l'assurance que la TOE et la documentation utilisées pour l'évaluation sont bien celles qui sont préparées pour la distribution.

2.6.1.1 Automatisation de la CM (ACM_AUT)

92 L'automatisation de la CM établit le niveau d'automatisation utilisé pour contrôler les objets gérés en configuration.

2.6.1.2 Capacités de la CM (ACM_CAP)

93 Les capacités de la CM définissent les caractéristiques du système de gestion de configuration.

Classe d'assurance	Famille d'assurance	Abréviation
Classe ACM : Gestion de configuration	Automatisation de la CM	ACM_AUT
	Capacités de la CM	ACM_CAP
	Portée de la CM	ACM_SCP
Classe ADO : Livraison et exploitation	Livraison	ADO_DEL
	Installation, génération et démarrage	ADO_IGS
Classe ADV : Développement	Spécifications fonctionnelles	ADV_FSP
	Conception de haut niveau	ADV_HLD
	Représentation de l'implémentation	ADV_IMP
	Parties internes de la TSF	ADV_INT
	Conception de bas niveau	ADV_LLD
	Correspondance des représentations	ADV_RCR
	Modélisation de la politique de sécurité	ADV_SPM
Classe AGD : Guides	Guide de l'administrateur	AGD_ADM
	Guide de l'utilisateur	AGD_USR
Classe ALC : Support au cycle de vie	Sécurité du développement	ALC_DVS
	Correction d'anomalies	ALC_FLR
	Définition du cycle de vie	ALC_LCD
	Outils et techniques	ALC_TAT
Classe ATE : Tests	Couverture	ATE_COV
	Profondeur	ATE_DPT
	Tests fonctionnels	ATE_FUN
	Tests indépendants	ATE_IND
Classe AVA : Estimation des vulnérabilités	Analyse des canaux cachés	AVA_CCA
	Utilisation impropre	AVA_MSU
	Résistance des fonctions de sécurité de la TOE	AVA_SOF
	Analyse de vulnérabilités	AVA_VLA

Tableau 2.1 -Décomposition en familles d'assurance et correspondances

2.6.1.3 Portée de la CM (ACM_SCP)

94 La portée de la CM indique les éléments de la TOE qui doivent être contrôlés par le système de gestion de configuration.

2.6.2 Classe ADO : Livraison et exploitation

95 La classe d'assurance ADO définit des exigences pour les mesures, procédures et normes qui traitent de livraison, d'installation et d'utilisation opérationnelle sûres de la TOE, garantissant que la protection en terme de sécurité offerte par la TOE

n'est pas compromise pendant son transfert, son installation, son démarrage et son exploitation.

2.6.2.1 Livraison (ADO_DEL)

96 La livraison couvre les procédures utilisées pour maintenir la sécurité pendant le transfert de la TOE vers l'utilisateur, aussi bien pour la livraison initiale que pour celles faisant suite à des modifications. Elle inclut des procédures ou des opérations spéciales nécessaires pour démontrer l'authenticité de la TOE livrée. De telles procédures et mesures constituent la base pour garantir que la protection en terme de sécurité offerte par la TOE n'est pas compromise pendant son transfert. Alors que la conformité avec les exigences de livraison ne peut pas toujours être déterminée au moment de l'évaluation de la TOE, il est possible d'évaluer les procédures qu'un développeur a élaborées pour distribuer la TOE aux utilisateurs.

2.6.2.2 Installation, génération et démarrage (ADO_IGS)

97 L'installation, la génération et le démarrage exigent que l'exemplaire de la TOE soit configuré et mis en œuvre par l'administrateur afin de présenter les mêmes propriétés de protection que l'exemplaire original de la TOE. Les procédures d'installation, de génération et de démarrage procurent la confiance dans le fait que l'administrateur aura connaissance des paramètres de configuration de la TOE et de la manière dont ils peuvent affecter la TSF.

2.6.3 Classe ADV : Développement

98 La classe d'assurance ADV définit des exigences pour le raffinement pas-à-pas de la TSF depuis les spécifications globales de la TOE dans la ST jusqu'à l'implémentation effective. Chacune des représentations de la TSF qui résulte de ce processus fournit des informations qui aident l'évaluateur à déterminer si les exigences fonctionnelles de la TOE ont été satisfaites.

2.6.3.1 Spécifications fonctionnelles (ADV_FSP)

99 Les spécifications fonctionnelles décrivent la TSF et doivent constituer une instantiation complète et exacte des exigences fonctionnelles de sécurité de la TOE. Les spécifications fonctionnelles détaillent également les interfaces externes de la TOE. Les utilisateurs de la TOE sont censés interagir avec la TSF au travers de ces interfaces.

2.6.3.2 Conception de haut niveau (ADV_HLD)

100 La conception de haut niveau consiste en des spécifications de conception de haut niveau qui raffinent les spécifications fonctionnelles de la TSF pour aboutir aux principaux constituants de la TSF. La conception de haut niveau identifie la structure de base de la TSF et les principaux éléments matériels, microprogrammés et logiciels.

2.6.3.3 Représentation de l’implémentation (ADV_IMP)

101 La représentation de l’implémentation correspond à la représentation la moins abstraite de la TSF. Elle intègre les fonctionnements internes détaillés de la TSF en termes de code source, schémas des matériels, etc., selon les cas.

2.6.3.4 Parties internes de la TSF (ADV_INT)

102 Les exigences concernant les parties internes de la TSF spécifient la structuration interne requise pour la TSF.

2.6.3.5 Conception de bas niveau (ADV_LLD)

103 La conception de bas niveau consiste en des spécifications de conception détaillées qui raffinent la conception de haut niveau à un niveau de détail qui peut servir de base pour la programmation ou pour la réalisation du matériel.

2.6.3.6 Correspondance des représentations (ADV_RCR)

104 La correspondance des représentations est une démonstration des liens entre toutes les paires adjacentes de représentations de la TSF disponibles, depuis les spécifications globales de la TOE jusqu’à la représentation la moins abstraite de la TSF qui est fournie.

2.6.3.7 Modélisation de la politique de sécurité (ADV_SPM)

105 Les modèles de politique de sécurité sont des représentations structurées des politiques de sécurité de la TSP et sont utilisés pour fournir une assurance accrue dans le fait que les spécifications fonctionnelles correspondent aux politiques de sécurité de la TSP et finalement aux exigences fonctionnelles de sécurité de la TOE. Cela est réalisé au moyen des correspondances entre les spécifications fonctionnelles, le modèle de politique de sécurité et les politiques de sécurité qui sont modélisées.

2.6.4 Classe AGD : Guides

106 La classe d’assurance AGD définit des exigences destinées à permettre la compréhension, la couverture et la complétude de la documentation d’exploitation fournie par le développeur. Cette documentation, qui offre deux catégories d’informations, pour les utilisateurs et pour les administrateurs, constitue un facteur important pour l’exploitation sûre de la TOE.

2.6.4.1 Guide de l’administrateur (AGD_ADM)

107 Les exigences sur le guide de l'administrateur contribuent à garantir que les contraintes environnementales peuvent être comprises par les administrateurs et les opérateurs de la TOE. Le guide de l'administrateur constitue le principal moyen à la disposition du développeur pour donner aux administrateurs de la TOE des informations détaillées et exactes sur la façon d’administrer la TOE de façon sûre et d’utiliser efficacement les privilèges et les fonctions de protection de la TSF.

2.6.4.2 Guide de l'utilisateur (AGD_USR)

108 Les exigences sur le guide de l'utilisateur contribuent à garantir que les utilisateurs sont capables d'exploiter la TOE de façon sûre (e.g. les contraintes d'utilisation dont l'existence est supposée dans le PP ou la ST doivent être clairement expliquées et illustrées). Le guide de l'utilisateur constitue le principal moyen à la disposition du développeur pour donner aux utilisateurs de la TOE le contexte et les informations spécifiques nécessaires sur la manière d'utiliser correctement les fonctions de protection de la TOE. Le guide de l'utilisateur doit offrir des explications sur les deux aspects suivants. Premièrement, il doit expliquer ce que font les fonctions de sécurité visibles par les utilisateurs et comment elles doivent être utilisées, de telle façon que les utilisateurs soient capables de protéger de manière cohérente et efficace leurs informations. Deuxièmement, il doit expliquer le rôle de l'utilisateur dans le maintien de la sécurité de la TOE.

2.6.5 Classe ALC : Support au cycle de vie

109 La classe d'assurance ALC définit des exigences pour obtenir une assurance au moyen de l'adoption d'un modèle de cycle de vie bien défini qui couvre toutes les étapes du développement de la TOE, y compris les procédures et les politiques de correction d'anomalies, l'utilisation correcte d'outils et de techniques et les mesures de sécurité utilisées pour protéger l'environnement de développement.

2.6.5.1 Sécurité du développement (ALC_DVS)

110 La sécurité du développement couvre les mesures de sécurité physiques, organisationnelles, touchant au personnel et autres, utilisées dans l'environnement de développement. Elle inclut la sécurité physique concernant le ou les sites de développement et les contrôles relatifs à la sélection et au recrutement du personnel de développement.

2.6.5.2 Correction d'anomalies (ALC_FLR)

111 La correction d'anomalies garantit que les anomalies découvertes par les utilisateurs de la TOE seront prises en compte et corrigées tant que la TOE est suivie par le développeur. Alors que la conformité future avec les exigences de correction d'anomalies ne peut pas être déterminée au moment de l'évaluation de la TOE, il est possible d'évaluer les procédures et les politiques qu'un développeur a mises en place pour prendre en compte et corriger les anomalies et pour distribuer les corrections aux utilisateurs.

2.6.5.3 Définition du cycle de vie (ALC_LCD)

112 La définition du cycle de vie établit que les méthodes de développement utilisées par un développeur pour produire la TOE inclut les considérations et les activités identifiées dans le processus de développement et les exigences pour le support d'exploitation. La confiance dans la correspondance entre les exigences et la TOE est plus élevée si l'analyse de sécurité et la production des éléments de preuve sont faites sur une base habituelle, faisant intégralement partie du processus de développement et des activités de support d'exploitation. Il n'est pas dans

l'intention de ce composant d'imposer un processus de développement spécifique quelconque.

2.6.5.4 Outils et techniques (ALC_TAT)

113 Les outils et techniques traitent du besoin de définir les outils de développement utilisés pour analyser et implémenter la TOE. Ils incluent les exigences relatives aux outils de développement et aux options dépendant de l'implémentation de ces outils.

2.6.6 Classe ATE : Tests

114 La classe d'assurance ATE formule des exigences de tests qui démontrent que la TSF satisfait aux exigences fonctionnelles de sécurité de la TOE.

2.6.6.1 Couverture (ATE_COV)

115 La couverture traite de la complétude des tests fonctionnels réalisés par le développeur sur la TOE. Elle concerne l'étendue des tests appliqués aux fonctions de sécurité de la TOE.

2.6.6.2 Profondeur (ATE_DPT)

116 La profondeur traite du niveau de détail avec lequel le développeur teste la TOE. Le test des fonctions de sécurité est basé sur le caractère de plus en plus détaillé des informations déduites de l'analyse des représentations de la TSF.

2.6.6.3 Tests fonctionnels (ATE_FUN)

117 Les tests fonctionnels permettent d'établir que la TSF possède les propriétés nécessaires pour satisfaire aux exigences de sa ST. Les tests fonctionnels procurent l'assurance que la TSF satisfait au moins aux exigences des composants fonctionnels choisis. Cependant, les tests fonctionnels n'établissent pas que la TSF ne fait pas autre chose que ce qui est attendu. Cette famille est focalisée sur les tests fonctionnels réalisés par le développeur.

2.6.6.4 Tests indépendants (ATE_IND)

118 Les tests indépendants spécifient le niveau avec lequel les tests fonctionnels de la TOE doivent être réalisés par une partie différente du développeur (e.g. une tierce partie). Cette famille offre une valeur ajoutée par l'introduction de tests qui ne font pas partie des tests du développeur.

2.6.7 Classe AVA : Estimation des vulnérabilités

119 La classe d'assurance AVA définit des exigences destinées à l'identification des vulnérabilités exploitables. Elle concerne, de façon spécifique, les vulnérabilités introduites pendant la construction, l'exploitation, l'utilisation impropre ou la configuration incorrecte de la TOE.

2.6.7.1 Analyse des canaux cachés (AVA_CCA)

120 L'analyse des canaux cachés est destinée à découvrir et à analyser les canaux de communication non prévus qui peuvent être exploités pour violer la TSP prévue.

2.6.7.2 Utilisation impropre (AVA_MSU)

121 L'analyse relative à l'utilisation impropre a pour but d'examiner si un administrateur ou un utilisateur pourrait raisonnablement être capable de déterminer, avec une bonne compréhension des guides, si la TOE est configurée et exploitée d'une manière non sûre.

2.6.7.3 Résistance des fonctions de sécurité de la TOE (AVA_SOF)

122 L'analyse de la résistance des fonctions concerne les fonctions de sécurité de la TOE qui sont réalisées par un mécanisme faisant appel au calcul des probabilités ou des permutations (e.g. un mot de passe ou une fonction de hachage). Même si de telles fonctions ne peuvent pas être court-circuitées, désactivées ou altérées, il peut toujours être possible de les mettre en échec par une attaque directe. Un niveau ou une métrique spécifique peut être annoncé pour la résistance de chacune de ces fonctions. L'analyse de la résistance des fonctions est menée dans le but de déterminer si de telles fonctions satisfont à l'annonce ou la dépassent. Par exemple, l'analyse de la résistance des fonctions d'un mécanisme de mot de passe peut démontrer que la fonction de mot de passe satisfait à la résistance annoncée en montrant que l'espace des mots de passe est suffisamment large.

2.6.7.4 Analyse de vulnérabilités (AVA_VLA)

123 L'analyse de vulnérabilité consiste à identifier les anomalies qui ont pu être introduites dans les différentes étapes du raffinement effectué pendant le développement. Elle aboutit à la définition des tests de pénétration grâce à la collecte des informations nécessaires relatives : (1) à la complétude de la TSF (est-ce-que la TSF contre toutes les menaces annoncées ?) et (2) aux dépendances entre toutes les fonctions de sécurité. Ces vulnérabilités potentielles sont estimées en effectuant des tests de pénétration pour déterminer si elles pourraient en pratique être exploitables pour compromettre la sécurité de la TOE.

2.7 Classification de la maintenance

124 Les exigences pour la maintenance de l'assurance sont traitées de la même façon qu'une classe d'assurance et sont présentées en utilisant la structure de classe définie ci-dessous.

125 Les familles relatives à la maintenance de l'assurance et l'abréviation de chaque famille sont présentées dans le tableau 2.2.

Classe d’assurance	Famille d’assurance	Abréviation
Maintenance de l’assurance	Plan de maintenance de l’assurance	AMA_AMP
	Rapport de classification des composants de la TOE	AMA_CAT
	Preuve de la maintenance de l’assurance	AMA_EVD
	Analyse d’impact sur la sécurité	AMA_SIA

Tableau 2.2 -Décomposition de la classe “Maintenance de l’assurance”

2.8 Vue d’ensemble de la classe et des familles de la maintenance de l'assurance

126 Les paragraphes suivants résument la classe et les familles d’assurance du chapitre 16. Les résumés de la classe et des familles sont présentés dans le même ordre d’apparition qu’au chapitre 16.

2.8.1 Classe AMA : Maintenance de l’assurance

127 La classe d’assurance AMA est destinée à maintenir l’assurance que la TOE continuera à satisfaire à sa cible de sécurité quand des changements sont effectués sur la TOE ou son environnement. Chacune des familles dans cette classe identifie les tâches du développeur et de l’évaluateur qui doivent être appliquées *après* que la TOE ait été évaluée avec succès, bien que certaines exigences peuvent être appliquées au moment de l’évaluation.

2.8.1.1 Plan de maintenance de l’assurance (AMA_AMP)

128 Le plan de maintenance de l'assurance identifie les plans et les procédures qu’un développeur doit mettre en œuvre pour garantir que l’assurance établie pour la TOE évaluée est maintenue quand des changements sont effectués sur la TOE ou son environnement.

2.8.1.2 Rapport de classification des composants de la TOE (AMA_CAT)

129 Le rapport de classification des composants de la TOE fournit une classification des composants d’une TOE (e.g. les sous-systèmes de la TSF) selon leur pertinence pour à la sécurité. Cette classification permet de cibler l’analyse d’impact sur la sécurité réalisée par le développeur.

2.8.1.3 Preuve de la maintenance de l’assurance (AMA_EVD)

130 La preuve de la maintenance de l’assurance est destinée à établir la confiance dans le fait que l’assurance dans la TOE est maintenue par le développeur, en accord avec le plan de maintenance de l’assurance.

2.8.1.4 Analyse d'impact sur la sécurité (AMA_SIA)

131 L'analyse d'impact sur la sécurité a pour but d'établir la confiance dans le fait que l'assurance a été maintenue dans la TOE au moyen d'une analyse réalisée par le développeur de l'impact sur la sécurité de toutes les modifications qui ont affecté la TOE depuis son évaluation.

3 Critères d'évaluation d'un profil de protection et d'une cible de sécurité

3.1 Généralités

132 Le présent chapitre introduit les critères d'évaluation des PP et des ST. Les critères d'évaluation sont ensuite complètement présentés dans le chapitre 4, Classe APE : Evaluation d'un profil de protection et dans le chapitre 5, Classe ASE : Evaluation d'une cible de sécurité.

133 Ces critères sont les premières exigences présentées dans cette partie 3 car l'évaluation du PP et de la ST sera normalement effectuée avant l'évaluation de la TOE. Ils jouent un rôle particulier du fait que les informations relatives à la TOE sont appréciées et que les exigences fonctionnelles et d'assurance sont évaluées pour déterminer si le PP ou la ST constitue une base significative pour l'évaluation d'une TOE.

134 Bien que ces critères d'évaluation diffèrent quelque peu des exigences contenues dans les chapitres 7 à 14, ils sont présentés de façon similaire parce que les activités du développeur et de l'évaluateur sont comparables pour les évaluations de PP, de ST et de TOE.

135 Les classes relatives au PP et à la ST sont différentes des classes relatives à la TOE du fait que toutes les exigences contenues dans la classe relative au PP ou à la ST doivent être considérées pour l'évaluation d'un PP ou d'une ST, tandis que les exigences présentées dans les classes relatives à la TOE couvrent une large gamme de sujets qui ne sont pas tous à prendre en compte pour une TOE donnée.

136 Les critères d'évaluation pour les PP et les ST sont basés sur les informations fournies dans les annexes B et C de la partie 1 des CC. Des informations utiles sur le contexte des exigences contenues dans les classes APE et ASE, telles que présentées dans les chapitres suivants, peuvent y être trouvées.

3.2 Vue d'ensemble des critères relatifs à un profil de protection

3.2.1 Évaluation d'un profil de protection

137 Le but de l'évaluation d'un PP est de démontrer que le PP est complet, cohérent, techniquement correct et par conséquent convient pour servir d'énoncé des exigences pour une ou plusieurs TOE évaluables. Un tel PP remplit les conditions requises pour faire partie d'un registre de PP.

3.2.2 Relation avec les critères d'évaluation d'une cible de sécurité

138 Comme décrit dans les annexes B et C de la partie 1 des CC, il existe de nombreuses similitudes dans la structure et le contenu entre le PP générique et la ST spécifique à une TOE. En conséquence, les critères pour évaluer les PP contiennent des

exigences qui sont semblables à nombre d'exigences pour les ST, et les critères relatifs aux deux sont présentés de la même façon.

3.2.3 Tâches de l'évaluateur

3.2.3.1 Tâches de l'évaluateur pour une évaluation basée uniquement sur les exigences des CC

139 Les évaluateurs qui effectuent l'évaluation d'un PP n'incluant pas d'exigences extérieures à la norme doivent appliquer les exigences de la classe APE telles qu'elles sont décrites dans le tableau 3.1.

Classe	Famille	Abréviation
Classe APE : Evaluation d'un profil de protection	Profil de Protection, Description de la TOE	APE_DES
	Profil de protection, Environnement de sécurité	APE_ENV
	Profil de protection, Introduction du PP	APE_INT
	Profil de protection, Objectifs de sécurité	APE_OBJ
	Profil de protection, Exigences de sécurité des TI	APE_REQ

Tableau 3.1 -Familles relatives au profil de protection - exigences des CC uniquement

3.2.3.2 Tâches de l'évaluateur pour une évaluation CC-étendue

140 Les évaluateurs qui effectuent l'évaluation d'un PP incluant des exigences ne provenant pas de la norme doivent appliquer les exigences de la classe APE telles qu'elles sont décrites dans le tableau 3.2.

Classe	Famille	Abréviation
Classe APE : Evaluation d'un profil de protection	Profil de Protection, Description de la TOE	APE_DES
	Profil de protection, Environnement de sécurité	APE_ENV
	Profil de protection, Introduction du PP	APE_INT
	Profil de protection, Objectifs de sécurité	APE_OBJ
	Profil de protection, Exigences de sécurité des TI	APE_REQ
	Profil de protection, Exigences de sécurité des TI explicitement énoncées	**APE_SRE**

Tableau 3.2 -Familles relatives au profil de protection - exigences CC-étendues

3.3 Vue d'ensemble des critères relatifs à une cible de sécurité

3.3.1 Évaluation d'une cible de sécurité

141 Le but de l'évaluation d'une ST est de démontrer que la ST est complète, cohérente, techniquement correcte et par conséquent convient pour servir de base à l'évaluation de la TOE correspondante.

3.3.2 Relation avec les autres critères d'évaluation de cette partie 3

142 Il y a deux étapes bien identifiées pour l'évaluation d'une TOE ; l'évaluation de la ST et l'évaluation de la TOE correspondante. Les exigences pour les évaluations de ST sont présentées dans ce chapitre et dans le chapitre 6, tandis que les exigences pour les évaluations de TOE sont contenues dans les chapitres 7 à 14.

143 L'évaluation d'une ST inclut l'évaluation des annonces de conformité à un PP. Si la ST ne prétend pas être conforme à un PP, la partie "annonces de conformité à un PP" de la ST doit contenir une déclaration indiquant que la TOE ne prétend pas être conforme à un quelconque PP.

3.3.3 Tâches de l'évaluateur

3.3.3.1 Tâches de l'évaluateur pour une évaluation basée uniquement sur les exigences des CC

144 Les évaluateurs qui effectuent l'évaluation d'une ST n'incluant pas d'exigences extérieures à la norme doivent appliquer les exigences de la classe ASE telles qu'elles sont décrites dans le tableau 3.3.

Classe	Famille	Abréviation
Classe ASE : Evaluation d'une cible de sécurité	Cible de sécurité, Description de la TOE	ASE_DES
	Cible de sécurité, Environnement de sécurité	ASE_ENV
	Cible de sécurité, Introduction de la ST	ASE_INT
	Cible de sécurité, Objectifs de sécurité	ASE_OBJ
	Cible de sécurité, Annonce de conformité à un PP	ASE_PPC
	Cible de sécurité, Exigences de sécurité des TI	ASE_REQ
	Cible de sécurité, Spécifications globales de la TOE	ASE_TSS

Tableau 3.3 -Familles relatives à la cible de sécurité - exigences des CC uniquement

3.3.3.2 Tâches de l'évaluateur pour une évaluation CC-étendue

145 Les évaluateurs qui effectuent l'évaluation d'une ST incluant des exigences extérieures à la norme doivent appliquer les exigences de la classe ASE telles qu'elles sont décrites dans le tableau 3.4.

Classe	Famille	Abréviation
Classe ASE : Evaluation d'une cible de sécurité	Cible de sécurité, Description de la TOE	ASE_DES
	Cible de sécurité, Environnement de sécurité	ASE_ENV
	Cible de sécurité, Introduction de la ST	ASE_INT
	Cible de sécurité, Objectifs de sécurité	ASE_OBJ
	Cible de sécurité, Annonce de conformité à un PP	ASE_PPC
	Cible de sécurité, Exigences de sécurité des TI	ASE_REQ
	Cible de sécurité, Exigences de sécurité des TI explicitement énoncées	**ASE_SRE**
	Cible de sécurité, Spécifications globales de la TOE	ASE_TSS

Tableau 3.4 -Familles relatives à la cible de sécurité - exigences CC-étendues

4 Classe APE : Evaluation d’un profil de protection

146 Le but de l’évaluation d’un PP est de démontrer que le PP est complet, cohérent et techniquement correct. Un PP évalué convient pour servir de base au développement de ST. Un tel PP remplit les conditions requises pour figurer dans un registre.

147 La figure 4.1 présente les familles contenues dans cette classe.

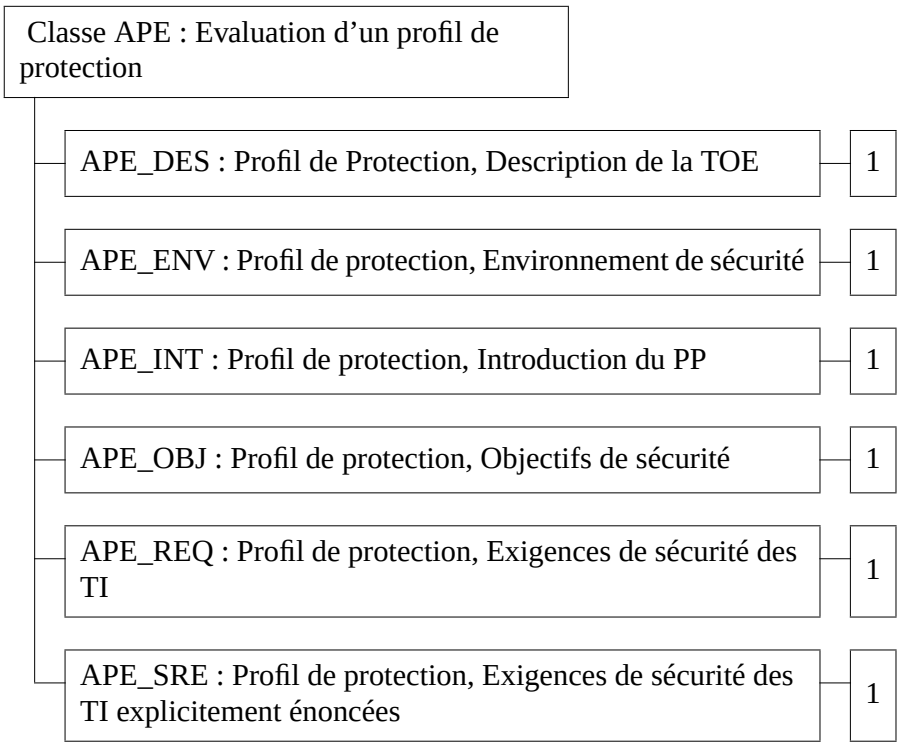

Figure 4.1 - Décomposition de la classe “Evaluation d’un profil de protection”

4.1 Description de la TOE (APE_DES)

Objectifs

148 La description de la TOE constitue une aide à la compréhension des exigences de sécurité de la TOE. L'évaluation de la description de la TOE est exigée pour montrer qu'elle est intelligible, qu'elle possède une cohérence interne et qu'elle est cohérente avec toutes les autres parties du PP.

APE_DES.1 Profil de protection, description de la TOE, exigences d'évaluation

Dépendances :

APE_ENV.1 Profil de protection, environnement de sécurité, exigences d'évaluation

APE_INT.1 Profil de protection, introduction du PP, exigences d'évaluation

APE_OBJ.1 Profil de protection, objectifs de sécurité, exigences d'évaluation

APE_REQ.1 Profil de protection, exigences de sécurité des TI, exigences d'évaluation

Tâches du développeur :

APE_DES.1.1D **Le développeur du PP doit fournir une description de la TOE en tant que partie du PP.**

Contenu et présentation des éléments de preuve :

APE_DES.1.1C **La description de la TOE doit au minimum décrire le type du produit et les caractéristiques générales des TI de la TOE.**

Tâches de l'évaluateur :

APE_DES.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

APE_DES.1.2E **L'évaluateur doit confirmer que la description de la TOE est intelligible et possède une cohérence interne.**

APE_DES.1.3E **L'évaluateur doit confirmer que la description de la TOE est cohérente avec les autres parties du PP.**

4.2 Environnement de sécurité (APE_ENV)

Objectifs

149 Afin de déterminer si les exigences de sécurité des TI dans le PP sont suffisantes, il est important que le problème de sécurité à résoudre soit parfaitement compris par toutes les parties participant à l'évaluation.

APE_ENV.1 Profil de protection, environnement de sécurité, exigences d'évaluation

Dépendances :

Pas de dépendances.

Tâches du développeur :

APE_ENV.1.1D **Le développeur du PP doit fournir un énoncé de l'environnement de sécurité de la TOE en tant que partie du PP.**

Contenu et présentation des éléments de preuve :

APE_ENV.1.1C **L'énoncé de l'environnement de sécurité de la TOE doit identifier et expliquer toute hypothèse relative à l'utilisation prévue de la TOE et à l'environnement d'utilisation de la TOE.**

APE_ENV.1.2C **L'énoncé de l'environnement de sécurité de la TOE doit identifier et expliquer toutes les menaces connues ou présumées contre les biens pour lesquels une protection sera exigée, soit par la TOE soit par son environnement.**

APE_ENV.1.3C **L'énoncé de l'environnement de sécurité de la TOE doit identifier et expliquer toutes les politiques de sécurité organisationnelles auxquelles la TOE doit satisfaire.**

Tâches de l'évaluateur :

APE_ENV.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

APE_ENV.1.2E **L'évaluateur doit confirmer que l'énoncé de l'environnement de sécurité de la TOE est intelligible et possède une cohérence interne.**

4.3 Introduction du PP (APE_INT)

Objectifs

150 L'introduction du PP contient les informations concernant la gestion du document et les informations générales nécessaires au maintien d'un registre de PP. L'évaluation de l'introduction du PP est exigée pour démontrer que le PP est correctement identifié et qu'elle est cohérente avec toutes les autres parties du PP.

APE_INT.1 Profil de protection, introduction du PP, exigences d'évaluation

Dépendances :

APE_DES.1 Profil de protection, description de la TOE, exigences d'évaluation

APE_ENV.1 Profil de protection, environnement de sécurité, exigences d'évaluation

APE_OBJ.1 Profil de protection, objectifs de sécurité, exigences d'évaluation

APE_REQ.1 Profil de protection, exigences de sécurité des TI, exigences d'évaluation

Tâches du développeur :

APE_INT.1.1D **Le développeur du PP doit fournir une introduction du PP en tant que partie du PP.**

Contenu et présentation des éléments de preuve :

APE_INT.1.1C **L'introduction du PP doit contenir une identification du PP qui donne les informations de désignation et de description nécessaires pour identifier, cataloguer, enregistrer et faire référence au PP.**

APE_INT.1.2C **L'introduction du PP doit contenir une vue d'ensemble du PP qui résume le PP sous forme narrative.**

Tâches de l'évaluateur :

APE_INT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

APE_INT.1.2E **L'évaluateur doit confirmer que l'introduction du PP est intelligible et possède une cohérence interne.**

APE_INT.1.3E **L'évaluateur doit confirmer que l'introduction du PP est cohérente avec les autres parties du PP.**

4.4 Objectifs de sécurité (APE_OBJ)

Objectifs

151 L'énoncé des objectifs de sécurité constitue une présentation concise de la réponse prévue au problème de sécurité. L'évaluation des objectifs de sécurité est exigée pour démontrer que les objectifs déclarés traitent le problème de sécurité de façon adéquate. Les objectifs de sécurité sont classés en objectifs de sécurité pour la TOE et en objectifs de sécurité pour l'environnement. Il doit être montré que les objectifs de sécurité pour la TOE et pour l'environnement sont reliés aux menaces identifiées devant être contrées ou aux politiques et hypothèses devant être satisfaites par chacun d'entre eux.

APE_OBJ.1 Profil de protection, objectifs de sécurité, exigences d'évaluation

Dépendances :

APE_ENV.1 Profil de protection, environnement de sécurité, exigences d'évaluation

Tâches du développeur :

APE_OBJ.1.1D **Le développeur du PP doit fournir un énoncé des objectifs de sécurité en tant que partie du PP.**

APE_OBJ.1.2D **Le développeur du PP doit fournir l'argumentaire relatif aux objectifs de sécurité.**

Contenu et présentation des éléments de preuve :

APE_OBJ.1.1C **L'énoncé des objectifs de sécurité doit définir les objectifs de sécurité pour la TOE et pour son environnement.**

APE_OBJ.1.2C **Les objectifs de sécurité pour la TOE doivent être présentés clairement et reliés aux aspects des menaces identifiées devant être contrées par la TOE ou aux politiques de sécurité organisationnelles devant être satisfaites par la TOE.**

APE_OBJ.1.3C **Les objectifs de sécurité pour l'environnement doivent être présentés clairement et reliés aux aspects des menaces identifiées non complètement contrées par la TOE ou aux politiques de sécurité organisationnelles ou aux hypothèses auxquelles la TOE ne satisfait pas complètement.**

APE_OBJ.1.4C **L'argumentaire relatif aux objectifs de sécurité doit démontrer que les objectifs de sécurité déclarés conviennent pour contrer les menaces identifiées vis-à-vis de la sécurité.**

APE_OBJ.1.5C **L'argumentaire relatif aux objectifs de sécurité doit démontrer que les objectifs de sécurité déclarés conviennent pour couvrir toutes les politiques de sécurité organisationnelles et hypothèses identifiées.**

Tâches de l'évaluateur :

APE_OBJ.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

APE_OBJ.1.2E **L'évaluateur doit confirmer que l'énoncé des objectifs de sécurité est complet, intelligible et possède une cohérence interne.**

4.5 Exigences de sécurité des TI (APE_REQ)

Objectifs

152 Les exigences de sécurité des TI sélectionnées pour une TOE et présentées ou citées dans un PP doivent être évaluées pour confirmer qu'elles possèdent une cohérence interne et conduisent au développement d'une TOE qui satisfera à ses objectifs de sécurité.

153 Les objectifs de sécurité exprimés dans un PP peuvent ne pas tous être satisfaits par une TOE en adéquation avec ce dernier, car il y a des TOE qui peuvent dépendre de certaines exigences de sécurité des TI devant être satisfaites par l'environnement TI. Lorsque c'est le cas, les exigences de sécurité pour l'environnement TI doivent être exprimées clairement et évaluées dans leur contexte vis-à-vis des exigences de la TOE.

154 Cette famille présente les exigences d'évaluation qui permettent à l'évaluateur de déterminer qu'un PP convient pour servir de spécification des exigences pour une TOE évaluable. Les critères supplémentaires nécessaires à l'évaluation des exigences explicitement énoncées sont couverts par la famille APE_SRE.

Notes d'application

155 Le terme "exigences de sécurité des TI" fait référence aux "exigences de sécurité de la TOE" et aux "exigences de sécurité pour l'environnement TI" qui sont incluses en option.

156 Le terme "exigences de sécurité de la TOE" fait référence aux "exigences fonctionnelles de sécurité de la TOE" ou aux "exigences d'assurance de sécurité de la TOE".

157 Dans le composant APE_REQ.1, le mot "approprié" est utilisé pour indiquer que certains éléments autorisent l'utilisation d'options dans certains cas. L'applicabilité de telle ou telle option dépend du contexte défini dans le PP. Les informations détaillées relatives à tous ces aspects sont contenues dans l'annexe B de la partie 1 des CC.

APE_REQ.1 Profil de protection, exigences de sécurité des TI, exigences d'évaluation

Dépendances :

APE_OBJ.1 Profil de protection, objectifs de sécurité, exigences d'évaluation

Tâches du développeur :

APE_REQ.1.1D **Le développeur du PP doit fournir un énoncé des exigences de sécurité des TI en tant que partie du PP.**

APE_REQ.1.2D **Le développeur du PP doit fournir l'argumentaire relatif aux exigences de sécurité.**

Contenu et présentation des éléments de preuve :

APE_REQ.1.1C **L'énoncé des exigences fonctionnelles de sécurité de la TOE doit identifier les exigences fonctionnelles de sécurité de la TOE provenant des composants d'exigences fonctionnelles de la partie 2 des CC.**

APE_REQ.1.2C **L'énoncé des exigences d'assurance de sécurité de la TOE doit identifier les exigences d'assurance de sécurité de la TOE provenant des composants d'exigences d'assurance de la partie 3 des CC.**

APE_REQ.1.3C **L'énoncé des exigences d'assurance de sécurité de la TOE devrait inclure un niveau d'assurance de l'évaluation (EAL) tel que défini dans la partie 3 des CC.**

APE_REQ.1.4C **Les éléments de preuve doivent justifier que l'énoncé des exigences d'assurance de sécurité de la TOE est approprié.**

APE_REQ.1.5C **Le PP doit, quand cela est approprié, identifier toutes les exigences de sécurité pour l'environnement TI.**

APE_REQ.1.6C **Toutes les opérations complètement renseignées sur des exigences de sécurité des TI incluses dans le PP doivent être identifiées.**

APE_REQ.1.7C **Toutes les opérations non complètement renseignées sur des exigences de sécurité des TI incluses dans le PP doivent être identifiées.**

APE_REQ.1.8C **Les dépendances entre exigences de sécurité des TI incluses dans le PP devraient être satisfaites.**

APE_REQ.1.9C **Les éléments de preuve doivent justifier les raisons pour lesquelles il est approprié de ne pas satisfaire une ou plusieurs dépendances.**

APE_REQ.1.10C **Le PP doit inclure un énoncé du niveau de résistance minimum des fonctions pour les exigences de fonctionnelles sécurité de la TOE, choisi parmi les niveaux SOF-élémentaire, SOF-moyen ou SOF-élevé, quand cela est approprié.**

APE_REQ.1.11C **Le PP doit identifier toutes les exigences fonctionnelles de sécurité spécifiques de la TOE pour lesquelles l'utilisation d'une résistance des fonctions explicite est appropriée, ainsi que la métrique spécifique.**

APE_REQ.1.12C **L'argumentaire relatif aux exigences de sécurité doit démontrer que le niveau de résistance minimum des fonctions pour le PP, ainsi que toute annonce d'une résistance des fonctions explicite, est cohérent avec les objectifs de sécurité pour la TOE.**

APE_REQ.1.13C **L’argumentaire relatif aux exigences de sécurité doit démontrer que les exigences de sécurité des TI conviennent pour satisfaire aux objectifs de sécurité.**

APE_REQ.1.14C **L’argumentaire relatif aux exigences de sécurité doit démontrer que les exigences de sécurité des TI se soutiennent mutuellement et forment un ensemble possédant une cohérence interne.**

Tâches de l’évaluateur :

APE_REQ.1.1E **L’évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

APE_REQ.1.2E **L’évaluateur doit confirmer que l’énoncé des exigences de sécurité des TI est complet, intelligible et possède une cohérence interne.**

4.6 Exigences de sécurité des TI explicitement énoncées (APE_SRE)

Objectifs

158 S'il se trouve qu'après une rigoureuse analyse, aucun des composants d'exigences de la partie 2 ou de la partie 3 des CC n'est immédiatement applicable à toutes ou parties des exigences de sécurité des TI, l'auteur du PP peut spécifier d'autres exigences qui ne font pas référence aux CC. L'utilisation de telles exigences doit être justifiée.

159 Cette famille présente les exigences d'évaluation qui permettent à l'évaluateur de déterminer que les exigences explicitement énoncées sont exprimées clairement et sans ambiguïté. L'évaluation des exigences tirées des CC en conjonction avec des exigences de sécurité explicitement énoncées et valides est traitée par la famille APE_REQ.

160 Les exigences de sécurité des TI explicitement énoncées pour une TOE présentée ou citée dans un PP doivent être évaluées pour démontrer qu'elles sont exprimées clairement et sans ambiguïté.

Notes d'application

161 La formulation des exigences explicitement énoncées dans une structure comparable à celle des composants et des éléments existants des CC implique de choisir une désignation, une manière de les exprimer et un niveau de détail similaires.

162 L'utilisation des exigences des CC comme modèle signifie que les exigences peuvent être clairement identifiées, qu'elles se suffisent à elles-mêmes, et que l'application de chaque exigence est possible et aboutira à un résultat d'évaluation significatif basé sur l'annonce de l'adéquation de la TOE à cette exigence particulière.

163 Le terme "exigences de sécurité des TI" fait référence aux "exigences de sécurité de la TOE" et aux "exigences de sécurité pour l'environnement TI" qui sont incluses en option.

164 Le terme "exigences de sécurité de la TOE" fait référence aux "exigences fonctionnelles de sécurité de la TOE" ou aux "exigences d'assurance de sécurité de la TOE".

APE_SRE.1 Profil de protection, exigences de sécurité des TI explicitement énoncées, exigences d'évaluation

Dépendances :

APE_REQ.1 Profil de protection, exigences de sécurité des TI, exigences d'évaluation

Tâches du développeur :

APE_SRE.1.1D **Le développeur du PP doit fournir un énoncé des exigences de sécurité des TI en tant que partie du PP.**

APE_SRE.1.2D **Le développeur du PP doit fournir l'argumentaire relatif aux exigences de sécurité.**

Contenu et présentation des éléments de preuve :

APE_SRE.1.1C **Toutes les exigences de sécurité de la TOE qui sont explicitement énoncées sans faire référence aux CC doivent être identifiées.**

APE_SRE.1.2C **Toutes les exigences de sécurité pour l'environnement TI qui sont explicitement énoncées sans faire référence aux CC doivent être identifiées.**

APE_SRE.1.3C **Les éléments de preuve doivent justifier les raisons pour lesquelles les exigences de sécurité ont dû être explicitement énoncées.**

APE_SRE.1.4C **Les exigences de sécurité des TI explicitement énoncées doivent utiliser les composants, familles et classes d'exigences des CC comme modèle de présentation.**

APE_SRE.1.5C **Les exigences de sécurité des TI explicitement énoncées doivent être mesurables et énoncer des exigences d'évaluation objectives de telle façon qu'il soit possible de déterminer et de démontrer systématiquement la conformité ou la non conformité d'une TOE.**

APE_SRE.1.6C **Les exigences de sécurité des TI explicitement énoncées doivent être exprimées clairement et sans ambiguïté.**

APE_SRE.1.7C **L'argumentaire relatif aux exigences de sécurité doit démontrer que les exigences d'assurance sont applicables et appropriées pour soutenir toute exigence fonctionnelle de sécurité de la TOE explicitement énoncée.**

Tâches de l'évaluateur :

APE_SRE.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

APE_SRE.1.2E **L'évaluateur doit déterminer que toutes des dépendances relatives aux exigences de sécurité des TI explicitement énoncées ont été identifiées.**

5 Classe ASE : Evaluation d'une cible de sécurité

165 Le but de l'évaluation d'une ST est de démontrer que la ST est complète, cohérente, techniquement correcte et par conséquent convient pour servir de base à l'évaluation de la TOE correspondante.

166 La figure 5.1 présente les familles contenues dans cette classe.

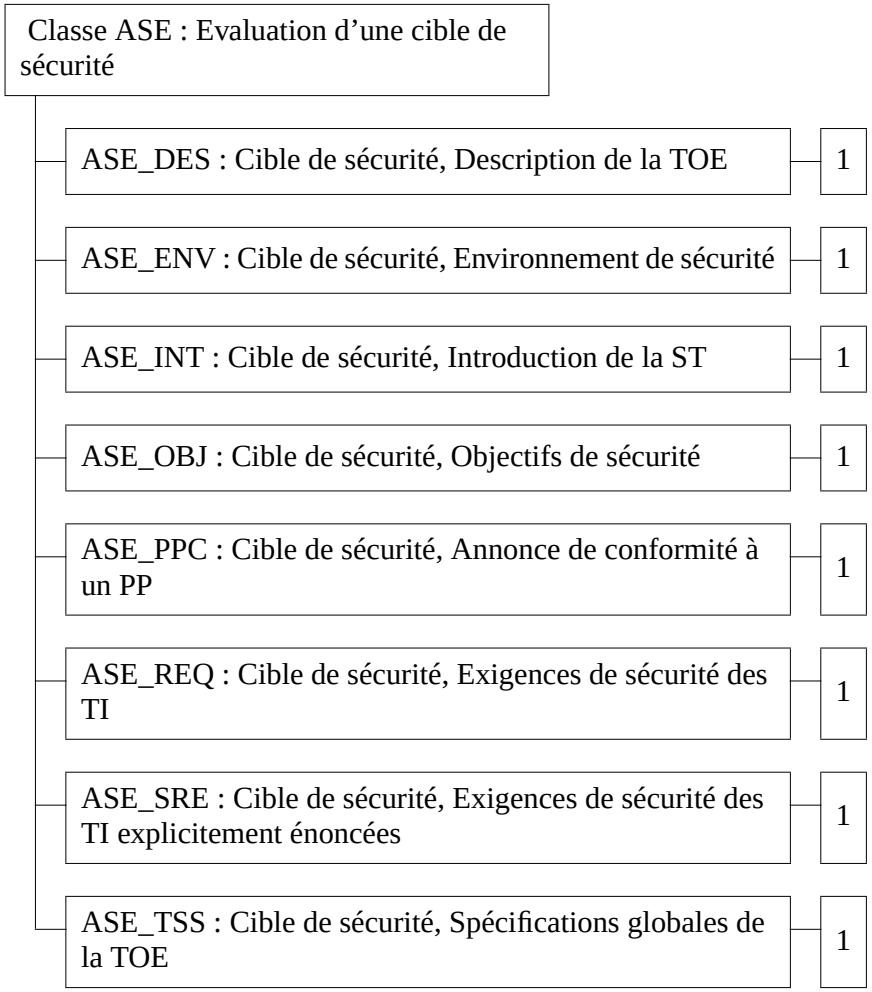

Figure 5.1 - Décomposition de la classe "Évaluation d'une cible de sécurité"

5.1 Description de la TOE (ASE_DES)

Objectifs

167 La description de la TOE constitue une aide à la compréhension des exigences de sécurité de la TOE. L'évaluation de la description de la TOE est exigée pour montrer qu'elle est intelligible, qu'elle possède une cohérence interne et qu'elle est cohérente avec toutes les autres parties de la ST.

ASE_DES.1 Cible de sécurité, description de la TOE, exigences d'évaluation

Dépendances :

ASE_ENV.1 Cible de sécurité, environnement de sécurité, exigences d'évaluation

ASE_INT.1 Cible de sécurité, introduction de la ST, exigences d'évaluation

ASE_OBJ.1 Cible de sécurité, objectifs de sécurité, exigences d'évaluation

ASE_PPC.1 Cible de sécurité, annonces de conformité à un PP, exigences d'évaluation

ASE_REQ.1 Cible de sécurité, exigences de sécurité des TI, exigences d'évaluation

ASE_TSS.1 Cible de sécurité, spécifications globales de la TOE, exigences d'évaluation

Tâches du développeur :

ASE_DES.1.1D **Le développeur doit fournir une description de la TOE en tant que partie de la ST.**

Contenu et présentation des éléments de preuve :

ASE_DES.1.1C **La description de la TOE doit décrire au minimum le type du produit ou du système, ainsi que le champ d'application et les limites de la TOE en termes généraux, à la fois de façon physique et logique.**

Tâches de l'évaluateur :

ASE_DES.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_DES.1.2E **L'évaluateur doit confirmer que la description de la TOE est intelligible et possède une cohérence interne.**

ASE_DES.1.3E **L'évaluateur doit confirmer que la description de la TOE est cohérente avec les autres parties de la ST.**

5.2 Environnement de sécurité (ASE_ENV)

Objectifs

168 Afin de déterminer si les exigences de sécurité des TI dans la ST sont suffisantes, il est important que le problème de sécurité à résoudre soit parfaitement compris par toutes les parties participant à l'évaluation.

ASE_ENV.1 Cible de sécurité, environnement de sécurité, exigences d'évaluation

Dépendances :

Pas de dépendances.

Tâches du développeur :

ASE_ENV.1.1D **Le développeur doit fournir un énoncé de l'environnement de sécurité de la TOE en tant que partie de la ST.**

Contenu et présentation des éléments de preuve :

ASE_ENV.1.1C **L'énoncé de l'environnement de sécurité de la TOE doit identifier et expliquer toute hypothèse relative à l'utilisation prévue de la TOE et à l'environnement d'utilisation de la TOE.**

ASE_ENV.1.2C **L'énoncé de l'environnement de sécurité de la TOE doit identifier et expliquer toutes les menaces connues ou présumées contre les biens pour lesquels une protection sera exigée, soit par la TOE soit par son environnement.**

ASE_ENV.1.3C **L'énoncé de l'environnement de sécurité de la TOE doit identifier et expliquer toutes les politiques de sécurité organisationnelles auxquelles la TOE doit satisfaire.**

Tâches de l'évaluateur :

ASE_ENV.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_ENV.1.2E **L'évaluateur doit confirmer que l'énoncé de l'environnement de sécurité de la TOE est intelligible et possède une cohérence interne.**

5.3 Introduction de la ST (ASE_INT)

Objectifs

169 L'introduction de la ST contient des éléments d'identification et d'indexation. L'évaluation de l'introduction de la ST est exigée pour démontrer que la ST est correctement identifiée et qu'elle est cohérente avec toutes les autres parties de la ST.

ASE_INT.1 Cible de sécurité, introduction de la ST, exigences d'évaluation

Dépendances :

ASE_DES.1 Cible de sécurité, description de la TOE, exigences d'évaluation

ASE_ENV.1 Cible de sécurité, environnement de sécurité, exigences d'évaluation

ASE_OBJ.1 Cible de sécurité, objectifs de sécurité, exigences d'évaluation

ASE_PPC.1 Cible de sécurité, annonces de conformité à un PP, exigences d'évaluation

ASE_REQ.1 Cible de sécurité, exigences de sécurité des TI, exigences d'évaluation

ASE_TSS.1 Cible de sécurité, spécifications globales de la TOE, exigences d'évaluation

Tâches du développeur :

ASE_INT.1.1D **Le développeur doit fournir une introduction à la ST en tant que partie de la ST.**

Contenu et présentation des éléments de preuve :

ASE_INT.1.1C **L'introduction de la ST doit contenir une identification de la ST qui donne les informations de désignation et de description nécessaires pour contrôler et identifier la ST et la TOE à laquelle elle fait référence.**

ASE_INT.1.2C **L'introduction de la ST doit contenir une vue d'ensemble de la ST qui résume la ST sous forme narrative.**

ASE_INT.1.3C **L'introduction de la ST doit contenir une annonce de conformité aux CC qui indique toute annonce de conformité de la TOE aux CC pouvant être évaluée.**

Tâches de l'évaluateur :

ASE_INT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_INT.1.2E **L’évaluateur doit confirmer que l’introduction de la ST est intelligible et possède une cohérence interne.**

ASE_INT.1.3E **L’évaluateur doit confirmer que l’introduction de la ST est cohérente avec les autres parties de la ST.**

5.4 Objectifs de sécurité (ASE_OBJ)

Objectifs

170 Les objectifs de sécurité constituent une présentation concise de la réponse prévue au problème de sécurité. L'évaluation des objectifs de sécurité est exigée pour démontrer que les objectifs déclarés traitent le problème de sécurité de façon adéquate. Les objectifs de sécurité sont classés en objectifs de sécurité pour la TOE et en objectifs de sécurité pour l'environnement. Il doit être montré que les objectifs de sécurité pour la TOE et pour l'environnement sont reliés aux menaces identifiées devant être contrées ou aux politiques et hypothèses devant être satisfaites par chacun d'entre eux.

ASE_OBJ.1 Cible de sécurité, objectifs de sécurité, exigences d'évaluation

Dépendances :

ASE_ENV.1 Cible de sécurité, environnement de sécurité, exigences d'évaluation

Tâches du développeur :

ASE_OBJ.1.1D **Le développeur doit fournir un énoncé des objectifs de sécurité en tant que partie de la ST.**

ASE_OBJ.1.2D **Le développeur doit fournir l'argumentaire relatif aux objectifs de sécurité.**

Contenu et présentation des éléments de preuve :

ASE_OBJ.1.1C **L'énoncé des objectifs de sécurité doit définir les objectifs de sécurité pour la TOE et pour son environnement.**

ASE_OBJ.1.2C **Les objectifs de sécurité pour la TOE doivent être présentés clairement et reliés aux aspects des menaces identifiées devant être contrées par la TOE ou aux politiques de sécurité organisationnelles devant être satisfaites par la TOE.**

ASE_OBJ.1.3C **Les objectifs de sécurité pour l'environnement doivent être présentés clairement et reliés aux aspects des menaces identifiées non complètement contrées par la TOE ou aux politiques de sécurité organisationnelles ou hypothèses auxquelles la TOE ne satisfait pas complètement.**

ASE_OBJ.1.4C **L'argumentaire relatif aux objectifs de sécurité doit démontrer que les objectifs de sécurité déclarés conviennent pour contrer les menaces identifiées vis-à-vis de la sécurité.**

ASE_OBJ.1.5C **L'argumentaire relatif aux objectifs de sécurité doit démontrer que les objectifs de sécurité déclarés conviennent pour couvrir toutes les politiques de sécurité organisationnelles et les hypothèses identifiées.**

Tâches de l'évaluateur :

ASE_OBJ.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_OBJ.1.2E **L'évaluateur doit confirmer que l'énoncé des objectifs de sécurité est complet, intelligible et possède une cohérence interne.**

5.5 Annonces de conformité à un PP (ASE_PPC)

Objectifs

171 Le but de l'évaluation des annonces de conformité à un PP de la cible de sécurité est de déterminer si la ST constitue une instantiation correcte du PP.

Notes d'application

172 La famille s'applique seulement dans le cas d'une annonce de conformité à un PP. Dans tous les autres cas, aucune tâche du développeur ou de l'évaluateur n'est nécessaire.

173 Bien qu'une activité supplémentaire soit nécessaire dans l'évaluation quand une annonce de conformité à un PP est faite, l'effort pour évaluer la ST est généralement plus faible que dans les cas où aucun PP n'est utilisé, parce qu'il est possible de réutiliser les résultats d'évaluation du PP pour l'évaluation de la ST.

ASE_PPC.1 Cible de sécurité, annonces de conformité à un PP, exigences d'évaluation

Dépendances :

ASE_OBJ.1 Cible de sécurité, objectifs de sécurité, exigences d'évaluation

ASE_REQ.1 Cible de sécurité, exigences de sécurité des TI, exigences d'évaluation

Tâches du développeur :

ASE_PPC.1.1D **Le développeur doit fournir toutes les annonces de conformité à un PP en tant que partie de la ST.**

ASE_PPC.1.2D **Le développeur doit fournir l'argumentaire des annonces de conformité à un PP pour chaque annonce de conformité à un PP qui est faite.**

Contenu et présentation des éléments de preuve :

ASE_PPC.1.1C **Chaque annonce de conformité à un PP doit identifier le PP vis-à-vis duquel la conformité est annoncée et inclure les précisions nécessaires pour cette annonce.**

ASE_PPC.1.2C **Chaque annonce de conformité à un PP doit identifier les énoncés des exigences de sécurité des TI qui satisfont aux opérations autorisées du PP ou bien préciser davantage les exigences du PP.**

ASE_PPC.1.3C **Chaque annonce de conformité à un PP doit identifier les énoncés des objectifs de sécurité et des exigences de sécurité des TI contenus dans la ST qui viennent en supplément de ceux contenus dans le PP.**

Tâches de l'évaluateur :

ASE_PPC.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_PPC.1.2E **L'évaluateur doit confirmer que les annonces de conformité à un PP constituent une instantiation correcte du PP.**

5.6 Exigences de sécurité des TI (ASE_REQ)

Objectifs

174 Les exigences de sécurité des TI sélectionnées pour une TOE et présentées ou citées dans une ST doivent être évaluées pour confirmer qu'elles possèdent une cohérence interne et conduisent au développement d'une TOE qui satisfera à ses objectifs de sécurité.

175 Cette famille présente les exigences d'évaluation qui permettent à l'évaluateur de déterminer qu'une ST convient pour servir de spécification des exigences pour la TOE correspondante. Les critères supplémentaires nécessaires à l'évaluation des exigences explicitement énoncées sont couverts par la famille ASE_SRE.

Notes d'application

176 Le terme "exigences de sécurité des TI" fait référence aux "exigences de sécurité de la TOE" et aux "exigences de sécurité pour l'environnement TI" qui sont incluses en option.

177 Le terme "exigences de sécurité de la TOE" fait référence aux "exigences fonctionnelles de sécurité de la TOE" ou aux "exigences d'assurance de sécurité de la TOE".

178 Dans le composant ASE_REQ.1, le mot "approprié" est utilisé pour indiquer que certains éléments autorisent l'utilisation d'options dans certains cas. L'applicabilité de telle ou telle option dépend du contexte défini dans la ST. Les informations détaillées relatives à tous ces aspects sont contenues dans l'annexe C de la partie 1 des CC.

ASE_REQ.1 Cible de sécurité, exigences de sécurité des TI, exigences d'évaluation

Dépendances :

ASE_OBJ.1 Cible de sécurité, objectifs de sécurité, exigences d'évaluation

Tâches du développeur :

ASE_REQ.1.1D **Le développeur doit fournir un énoncé des exigences de sécurité des TI en tant que partie de la ST.**

ASE_REQ.1.2D **Le développeur doit fournir l'argumentaire relatif aux exigences de sécurité.**

Contenu et présentation des éléments de preuve :

ASE_REQ.1.1C **L'énoncé des exigences fonctionnelles de sécurité de la TOE doit identifier les exigences fonctionnelles de sécurité de la TOE provenant des composants d'exigences fonctionnelles de la partie 2 des CC.**

ASE_REQ.1.2C **L'énoncé des exigences d'assurance de sécurité de la TOE doit identifier les exigences d'assurance de sécurité de la TOE provenant des composants d'exigences d'assurance de la partie 3 des CC.**

ASE_REQ.1.3C **L'énoncé des exigences d'assurance de sécurité de la TOE devrait inclure un niveau d'assurance de l'évaluation (EAL) tel que défini dans la partie 3 des CC.**

ASE_REQ.1.4C **Les éléments de preuve doivent justifier que l'énoncé des exigences d'assurance de sécurité de la TOE est approprié.**

ASE_REQ.1.5C **La ST doit, quand cela est approprié, identifier toutes les exigences de sécurité pour l'environnement TI.**

ASE_REQ.1.6C **Toutes les opérations sur des exigences de sécurité des TI incluses dans la ST doivent être identifiées et renseignées.**

ASE_REQ.1.7C **Les dépendances entre exigences de sécurité des TI incluses dans la ST devraient être satisfaites.**

ASE_REQ.1.8C **Les éléments de preuve doivent justifier les raisons pour lesquelles il est approprié de ne pas satisfaire une ou plusieurs dépendances.**

ASE_REQ.1.9C **La ST doit inclure un énoncé du niveau de résistance minimum des fonctions pour les exigences fonctionnelles de sécurité de la TOE, choisi parmi les niveaux SOF-élémentaire, SOF-moyen ou SOF-élevé, quand cela est approprié.**

ASE_REQ.1.10C **La ST doit identifier toutes les exigences fonctionnelles de sécurité spécifiques de la TOE pour lesquelles l'utilisation d'une résistance des fonctions explicite est appropriée, ainsi que la métrique spécifique.**

ASE_REQ.1.11C **L'argumentaire relatif aux exigences de sécurité doit démontrer que le niveau de résistance minimum des fonctions pour la ST, ainsi que toute annonce d'une résistance des fonctions explicite, est cohérente avec les objectifs de sécurité pour la TOE.**

ASE_REQ.1.12C **L'argumentaire relatif aux exigences de sécurité doit démontrer que les exigences de sécurité des TI conviennent pour satisfaire aux objectifs de sécurité.**

ASE_REQ.1.13C **L'argumentaire relatif aux exigences de sécurité doit démontrer que les exigences de sécurité des TI se soutiennent mutuellement et forment un ensemble possédant une cohérence interne.**

Tâches de l'évaluateur :

ASE_REQ.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_REQ.1.2E **L’évaluateur doit confirmer que l’énoncé des exigences de sécurité des TI est complet, intelligible et possède une cohérence interne.**

5.7 Exigences de sécurité des TI explicitement énoncées (ASE_SRE)

Objectifs

179 S'il se trouve qu'après une rigoureuse analyse, aucun des composants d'exigences de la partie 2 ou de la partie 3 des CC n'est immédiatement applicable à toutes ou parties des exigences de sécurité des TI, l'auteur de la ST peut spécifier d'autres exigences qui ne font pas référence aux CC. L'utilisation de telles exigences doit être justifiée.

180 Cette famille présente les exigences d'évaluation qui permettent à l'évaluateur de déterminer que les exigences explicitement énoncées sont exprimées clairement et sans ambiguïté. L'évaluation des exigences tirées des CC en conjonction avec des exigences de sécurité explicitement énoncées et valides est traitée par la famille ASE_REQ.

181 Les exigences de sécurité des TI explicitement énoncées pour une TOE présentée ou citée dans une ST doivent être évaluées pour démontrer qu'elles sont exprimées clairement et sans ambiguïté.

Notes d'application

182 La formulation des exigences explicitement énoncées dans une structure comparable à celle des composants et des éléments existants des CC implique de choisir une désignation, une manière de les exprimer et un niveau de détail similaires.

183 L'utilisation des exigences des CC comme modèle signifie que les exigences peuvent être clairement identifiées, qu'elles se suffisent à elles-mêmes et que l'application de chaque exigence est possible et aboutira à un résultat d'évaluation significatif basé sur l'annonce de l'adéquation de la TOE à cette exigence particulière.

184 Le terme "exigences de sécurité des TI" fait référence aux "exigences de sécurité de la TOE" et aux "exigences de sécurité pour l'environnement TI" qui sont incluses en option.

185 Le terme "exigences de sécurité de la TOE" fait référence aux "exigences fonctionnelles de sécurité de la TOE" ou aux "exigences d'assurance de sécurité de la TOE".

ASE_SRE.1 Cible de sécurité, exigences de sécurité des TI explicitement énoncées, exigences d'évaluation

Dépendances :

ASE_REQ.1 Cible de sécurité, exigences de sécurité des TI, exigences d'évaluation

Tâches du développeur :

ASE_SRE.1.1D **Le développeur doit fournir un énoncé des exigences de sécurité des TI en tant que partie de la ST.**

ASE_SRE.1.2D **Le développeur doit fournir l'argumentaire relatif aux exigences de sécurité.**

Contenu et présentation des éléments de preuve :

ASE_SRE.1.1C **Toutes les exigences de sécurité de la TOE qui sont explicitement énoncées sans faire référence aux CC doivent être identifiées.**

ASE_SRE.1.2C **Toutes les exigences de sécurité pour l'environnement TI qui sont explicitement énoncées sans faire référence aux CC doivent être identifiées.**

ASE_SRE.1.3C **Les éléments de preuve doivent justifier les raisons pour lesquelles les exigences de sécurité ont dû être explicitement énoncées.**

ASE_SRE.1.4C **Les exigences de sécurité des TI explicitement énoncées doivent utiliser les composants, familles et classes d'exigences des CC comme modèle de présentation.**

ASE_SRE.1.5C **Les exigences de sécurité des TI explicitement énoncées doivent être mesurables et énoncer des exigences d'évaluation objectives de telle façon qu'il soit possible de déterminer et de démontrer systématiquement la conformité ou la non conformité d'une TOE.**

ASE_SRE.1.6C **Les exigences de sécurité des TI explicitement énoncées doivent être exprimées clairement et sans ambiguïté.**

ASE_SRE.1.7C **L'argumentaire relatif aux exigences de sécurité doit démontrer que les exigences d'assurance sont applicables et appropriées pour supporter toute exigence fonctionnelle de sécurité de la TOE explicitement énoncée.**

Tâches de l'évaluateur :

ASE_SRE.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_SRE.1.2E **L'évaluateur doit déterminer que toutes des dépendances relatives aux exigences de sécurité des TI explicitement énoncées ont été identifiées.**

5.8 Spécifications globales de la TOE (ASE_TSS)

Objectifs

186 Les spécifications globales de la TOE donnent une définition de haut niveau des fonctions de sécurité qui sont censées satisfaire aux exigences fonctionnelles, et des mesures d'assurance prises pour satisfaire aux exigences d'assurance.

Notes d'application

187 Les relations entre les fonctions de sécurité des TI et les exigences fonctionnelles de sécurité de la TOE peuvent se présenter sous la forme de relations de "n à n" entités. Cependant, chaque fonction de sécurité doit contribuer à la satisfaction d'au moins une exigence de sécurité pour être en mesure de définir clairement la TSF. Les fonctions de sécurité qui ne remplissent pas cette condition ne devraient normalement pas être nécessaires. Il faut noter cependant que l'exigence suivant laquelle une fonction de sécurité doit contribuer à la satisfaction d'au moins une exigence de sécurité est exprimée d'une façon tout à fait générale, de sorte que toutes les fonctions de sécurité qui ont été considérées comme utiles pour la TOE doivent pouvoir être justifiées.

188 L'énoncé des mesures d'assurance est particulièrement pertinent dans tous les cas où des exigences d'assurance qui ont été prises en dehors des CC sont incluses dans la ST. Si les exigences d'assurance de sécurité de la TOE dans la ST sont basées exclusivement sur les niveaux d'assurance de l'évaluation des CC ou sur d'autres composants d'assurance de la partie 3 des CC, alors les mesures d'assurance pourraient être présentées sous la forme d'une référence aux documents qui montrent que les exigences d'assurance sont satisfaites.

189 Dans le composant ASE_TSS.1, le mot "approprié" est utilisé pour indiquer que certains éléments autorisent l'utilisation d'options dans certains cas. L'applicabilité de telle ou telle option dépend du contexte défini dans la ST. Les informations détaillées relatives à tous ces aspects sont contenues dans l'annexe C de la partie 1 des CC.

ASE_TSS.1 Cible de sécurité, spécifications globales de la TOE, exigences d'évaluation

Dépendances :

ASE_REQ.1 Cible de sécurité, exigences de sécurité des TI, exigences d'évaluation

Tâches du développeur :

ASE_TSS.1.1D **Le développeur doit fournir des spécifications globales de la TOE en tant que partie de la ST.**

ASE_TSS.1.2D **Le développeur doit fournir un argumentaire relatif aux spécifications globales de la TOE.**

Contenu et présentation des éléments de preuve :

ASE_TSS.1.1C **Les spécifications globales de la TOE doivent décrire les fonctions de sécurité des TI et les mesures d'assurance de la TOE.**

ASE_TSS.1.2C **Les spécifications globales de la TOE doivent relier les fonctions de sécurité des TI aux exigences fonctionnelles de sécurité de la TOE de telle façon qu'on puisse voir quelles fonctions de sécurité des TI satisfont à quelles exigences de fonctionnelles sécurité de la TOE, et que chaque fonction de sécurité des TI contribue à la satisfaction d'au moins une exigence fonctionnelle de sécurité de la TOE.**

ASE_TSS.1.3C **Les fonctions de sécurité des TI doivent être définies dans un style informel avec un niveau de détail suffisant pour qu'on puisse comprendre leur but.**

ASE_TSS.1.4C **Toutes les références aux mécanismes de sécurité inclus dans la ST doivent être reliées aux fonctions de sécurité pertinentes de telle façon qu'on puisse voir quels mécanismes de sécurité sont utilisés dans l'implémentation de chaque fonction.**

ASE_TSS.1.5C **L'argumentaire relatif aux spécifications globales de la TOE doit démontrer que les fonctions de sécurité des TI conviennent pour satisfaire aux exigences fonctionnelles de sécurité de la TOE.**

ASE_TSS.1.6C **L'argumentaire relatif aux spécifications globales de la TOE doit démontrer que les fonctions de sécurité des TI spécifiées coopèrent pour satisfaire aux exigences fonctionnelles de sécurité de la TOE.**

ASE_TSS.1.7C **Les spécifications globales de la TOE doivent relier les mesures d'assurance aux exigences d'assurance de telle façon qu'on puisse voir quelles mesures contribuent à la satisfaction de quelles exigences.**

ASE_TSS.1.8C **L'argumentaire relatif aux spécifications globales de la TOE doit démontrer que les mesures d'assurance satisfont à toutes les exigences d'assurance de la TOE.**

ASE_TSS.1.9C **Les spécifications globales de la TOE doivent identifier toutes les fonctions de sécurité des TI qui sont réalisées au moyen d'un mécanisme faisant appel au calcul des probabilités ou des permutations, quand cela est approprié.**

ASE_TSS.1.10C **Les spécifications globales de la TOE doivent, pour chaque fonction de sécurité des TI pour laquelle cela est approprié, spécifier la résistance des fonctions, soit à l'aide d'une métrique spécifique, soit sous forme d'un niveau SOF-élémentaire, SOF-moyen ou SOF-élevé.**

Tâches de l'évaluateur :

ASE_TSS.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ASE_TSS.1.2E **L’évaluateur doit confirmer que les spécifications globales de la TOE sont complètes, intelligibles et possèdent une cohérence interne.**

6 Niveaux d'assurance de l'évaluation

190 Les niveaux d'assurance de l'évaluation (EAL) fournissent une échelle croissante qui permet d'obtenir un équilibre entre le niveau d'assurance obtenu et le coût et la faisabilité nécessaires pour parvenir à ce degré d'assurance. L'approche CC identifie les concepts séparés de l'assurance dans une TOE à la fin de l'évaluation et de maintenance de cette assurance pendant l'exploitation opérationnelle de la TOE.

191 Il est important de noter que les familles et les composants figurant dans la partie 3 des CC ne sont pas tous inclus dans les EAL. Cela ne veut pas dire que ces familles et composants ne fournissent pas une assurance significative et souhaitable. En fait, il est prévu de les prendre en compte pour augmenter un EAL dans les PP et les ST pour lesquels ils seront utiles.

6.1 Généralités sur les niveaux d'assurance de l'évaluation (EAL)

192 Le tableau 6.1 constitue un résumé des EAL. Les colonnes représentent un ensemble hiérarchiquement ordonné d'EAL, tandis que les lignes représentent les familles d'assurance. Chaque numéro indiqué dans la matrice qui en résulte identifie un composant d'assurance spécifique lorsque cela est applicable.

193 Comme cela est présenté dans la section suivante, sept niveaux d'assurance de l'évaluation, hiérarchiquement ordonnés, sont définis dans les CC pour la cotation de l'assurance de la TOE. Ils sont hiérarchiquement ordonnés en ce sens que chaque EAL procure plus d'assurance que tous les EAL qui lui sont inférieurs. L'augmentation de l'assurance d'un EAL à l'autre est réalisé par la *substitution* d'un composant d'assurance par un composant hiérarchique provenant de la même famille d'assurance (i.e. augmentation de la rigueur, du champ d'application ou du degré d'approfondissement) et par l'*addition* de composants d'assurance provenant d'autres familles d'assurance (i.e. ajoutant de nouvelles exigences).

194 Ces EAL consistent en une combinaison pertinente de composants d'assurance, comme cela est décrit dans le chapitre 2 de la présente partie 3. Plus précisément, chaque EAL n'inclut pas plus d'un composant de chaque famille d'assurance et toutes les dépendances d'assurance de chaque composant sont prises en compte.

195 Bien que les EAL soient définis dans les CC, il est possible de représenter d'autres combinaisons d'assurance. De façon spécifique, la notion d'"augmentation" pour un EAL autorise l'addition de composants d'assurance (provenant de familles d'assurance qui ne sont pas déjà incluses dans l'EAL) ou la substitution de composants d'assurance (par un autre composant d'assurance hiérarchique provenant de la même famille d'assurance). Parmi les structures d'assurance définies dans les CC, seuls les EAL peuvent être augmentés. La notion d'un "EAL diminué d'un de ses composants d'assurance" n'est pas reconnue par la norme comme une annonce valide. L'augmentation entraîne l'obligation pour celui qui en

use de justifier l'utilité et la valeur ajoutée du composant d'assurance ajouté à l'EAL. Un EAL peut aussi être étendu à l'aide d'exigences d'assurance explicitement spécifiées.

6.2 Détails relatifs aux niveaux d'assurance de l'évaluation

196 Les sections suivantes définissent les EAL, mettant en évidence par l'utilisation de caractères gras les différences entre les exigences spécifiques et les commentaires relatifs à ces exigences.

Classe d'assurance	Famille d'assurance	Composants d'assurance par niveau d'assurance de l'évaluation						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Gestion de configuration	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Livraison et exploitation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Développement	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	**3**
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guides	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Support au cycle de vie	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Estimation des vulnérabilités	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

Tableau 6.1 - Résumé des niveaux d'assurance de l'évaluation

6.2.1 Niveau d'assurance de l'évaluation 1 (EAL1) - testé fonctionnellement

Objectifs

197 Le niveau EAL1 est applicable quand une certaine confiance dans un fonctionnement correct est exigée, mais que les menaces contre la sécurité ne sont pas considérées comme sérieuses. Ce niveau présentera un intérêt quand une assurance, obtenue de façon indépendante, est nécessaire pour confirmer qu'un soin approprié aura été apporté pour protéger les informations personnelles ou similaires.

198 Le niveau EAL1 permet d'obtenir une évaluation de la TOE, telle qu'elle est disponible à l'utilisateur, comprenant des tests indépendants par rapport à une spécification, ainsi qu'un examen des guides fournis. Une évaluation EAL1 doit pouvoir être réalisée avec succès sans l'assistance du développeur de la TOE et avec un coût minimal.

199 Une évaluation effectuée à ce niveau devrait procurer des éléments de preuve que la TOE fonctionne d'une manière conforme à celle décrite dans sa documentation et qu'elle fournit une protection utile contre les menaces identifiées.

Composants d'assurance

200 **Le niveau EAL1 (voir le tableau 6.2) procure un niveau d'assurance élémentaire consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles et d'interfaces ainsi que des guides, afin de comprendre le comportement de sécurité.**

201 **L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE.**

202 **Le présent EAL procure un accroissement significatif de l'assurance par rapport à un système ou un produit TI non évalué.**

Classe d'assurance	Composants d'assurance
Gestion de configuration	**ACM_CAP.1 Numéros de version**
Livraison et exploitation	**ADO_IGS.1 Procédures d'installation, de génération et de démarrage**
Développement	**ADV_FSP.1 Spécifications fonctionnelles informelles**
	ADV_RCR.1 Démonstration de correspondance informelle
Guides	**AGD_ADM.1 Guide de l'administrateur**
	AGD_USR.1 Guide de l'utilisateur
Tests	**ATE_IND.1 Tests indépendants - conformité**

Tableau 6.2 - Le niveau EAL1

6.2.2 Niveau d'assurance de l'évaluation 2 (EAL2) - testé structurellement

Objectifs

203 Le niveau EAL2 nécessite la coopération du développeur pour la fourniture des informations de conception et des résultats de tests, mais ne devrait pas exiger plus d'efforts de la part du développeur que ceux compatibles avec une bonne pratique commerciale. Ainsi il ne devrait pas entraîner un investissement substantiellement accru en termes de coûts et de délais.

204 Le niveau EAL2 s'applique par conséquent dans les circonstances où les développeurs et les utilisateurs exigent un niveau élémentaire à modéré de sécurité, déterminé de façon indépendante en l'absence de disponibilité immédiate du dossier complet de développement. Une telle situation peut survenir dans le cas de la sécurisation de systèmes hérités d'une situation précédente ou quand l'accès au développeur peut se trouver limité.

Composants d'assurance

205 Le niveau **EAL2** (voir le tableau **6.3**) procure une assurance consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles et d'interfaces, des guides et **de la conception de haut niveau** de la TOE, afin de comprendre le comportement de sécurité.

206 L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE, **par les éléments de preuve des tests du développeur basés sur les spécifications fonctionnelles, par la confirmation sélective et indépendante des résultats des tests du développeur, par l'analyse de la résistance des fonctions, ainsi que par des éléments de preuve d'une recherche des vulnérabilités évidentes (e.g. celles du domaine public) effectuée par le développeur**.

207 **Le niveau EAL2 procure également l'assurance au moyen d'une liste de configuration de la TOE et d'éléments de preuve relatifs à des procédures de livraison sûres.**

208 **Le présent EAL représente un accroissement significatif de l'assurance par rapport au niveau EAL1 en exigeant des tests du développeur, une analyse de vulnérabilités ainsi que des tests indépendants basés sur des spécifications plus détaillées de la TOE.**

Classe d'assurance	Composants d'assurance
Gestion de configuration	**ACM_CAP.2 Éléments de configuration**
Livraison et exploitation	**ADO_DEL.1 Procédures de livraison**
	ADO_IGS.1 Procédures d'installation, de génération et de démarrage
Développement	ADV_FSP.1 Spécifications fonctionnelles informelles
	ADV_HLD.1 Conception de haut niveau descriptive
	ADV_RCR.1 Démonstration de correspondance informelle
Guides	AGD_ADM.1 Guide de l'administrateur
	AGD_USR.1 Guide de l'utilisateur
Tests	**ATE_COV.1 Éléments de preuve de la couverture**
	ATE_FUN.1 Tests fonctionnels
	ATE_IND.2 Tests indépendants - échantillonnage
Estimation des vulnérabilités	**AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE**
	AVA_VLA.1 Analyse de vulnérabilités du développeur

Tableau 6.3 - Le niveau EAL2

6.2.3 Niveau d'assurance de l'évaluation 3 (EAL3) - testé et vérifié méthodiquement

Objectifs

209 Le niveau EAL3 permet à un développeur consciencieux d'obtenir l'assurance maximum issue d'une méthode de développement prenant en compte la sécurité en phase de conception sans bouleversement significatif des pratiques solides de développement déjà en vigueur.

210 Le niveau EAL3 s'applique dans les circonstances où les développeurs ou les utilisateurs exigent un niveau de sécurité modéré déterminé de façon indépendante ainsi qu'un examen approfondi de la TOE et de son développement ne nécessitant pas une retro-ingénierie importante.

Composants d'assurance

211 Le niveau **EAL3** (voir le tableau **6.4**) procure une assurance consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles et d'interfaces, des guides et de la conception de haut niveau de la TOE, afin de comprendre le comportement de sécurité**.**

212 L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE, par les éléments de preuve des tests du développeur basés sur les spécifications fonctionnelles **et la conception de haut niveau**, par la confirmation sélective et indépendante des résultats des tests du développeur, par l'analyse de la résistance des fonctions, ainsi que par des éléments de preuve d'une recherche des vulnérabilités évidentes (e.g. celles du domaine public) effectuée par le développeur.

213 Le niveau **EAL3** procure également l'assurance au moyen de la **mise en œuvre de contrôles de l'environnement de développement, d'une gestion de configuration de la TOE** et d'éléments de preuve relatifs à des procédures de livraison sûres.

214 **Le présent EAL représente un accroissement significatif de l'assurance par rapport au niveau EAL2 en exigeant une couverture plus complète pour les tests relatifs aux fonctions et mécanismes de sécurité ou des procédures de sécurité, qui fournissent une certaine confiance** dans le fait que la TOE ne sera pas altérée au cours du développement.

Classe d'assurance	Composants d'assurance
Gestion de configuration	**ACM_CAP.3 Contrôles des autorisations**
	ACM_SCP.1 Couverture de la TOE par la CM
Livraison et exploitation	ADO_DEL.1 Procédures de livraison
	ADO_IGS.1 Procédures d'installation, de génération et de démarrage
Développement	ADV_FSP.1 Spécifications fonctionnelles informelles
	ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité
	ADV_RCR.1 Démonstration de correspondance informelle
Guides	AGD_ADM.1 Guide de l'administrateur
	AGD_USR.1 Guide de l'utilisateur
Support au cycle de vie	**ALC_DVS.1 Identification des mesures de sécurité**
Tests	**ATE_COV.2 Analyse de la couverture**
	ATE_DPT.1 Tests : conception de haut niveau
	ATE_FUN.1 Tests fonctionnels
	ATE_IND.2 Tests indépendants - échantillonnage
Estimation des vulnérabilités	**AVA_MSU.1 Examen des guides**
	AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE
	AVA_VLA.1 Analyse de vulnérabilités du développeur

Tableau 6.4 - Le niveau EAL3

6.2.4 Niveau d'assurance de l'évaluation 4 (EAL4) - conçu, testé et revu méthodiquement

Objectifs

215 Le niveau EAL4 permet à un développeur d'obtenir l'assurance maximum issue d'une méthode de développement prenant en compte la sécurité, basée sur de bonnes pratiques de développement commercial qui, bien qu'étant rigoureuses, ne nécessitent pas de connaissances, d'expertises ou autres ressources spécialisées. Le niveau EAL4 constitue le niveau le plus élevé pour lequel le réajustement d'une gamme de produits existant devrait rester économiquement faisable.

216 Le niveau EAL4 est par conséquent applicable dans les circonstances où les développeurs ou les utilisateurs ont besoin d'un niveau de sécurité modéré à élevé déterminé de façon indépendante pour des TOE conventionnelles courantes, et où ils sont préparés à assumer les coûts supplémentaires dus à des techniques d'ingénierie spécifiques à la sécurité.

Composants d'assurance

217 Le niveau **EAL4** (voir le tableau **6.5**) procure une assurance consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles, des spécifications **complètes** des interfaces, des guides, de la conception de haut niveau, **de la conception de bas niveau** de la TOE et **d'un sous-ensemble de l'implémentation**, afin de comprendre le comportement de sécurité. **Cette assurance est renforcée par un modèle informel de la politique de sécurité de la TOE.**

218 L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE, par les éléments de preuve des tests du développeur basés sur les spécifications fonctionnelles et la conception de haut niveau, par la confirmation sélective et indépendante des résultats des tests du développeur, par l'analyse de la résistance des fonctions, par des éléments de preuve d'une recherche des vulnérabilités effectuée par le développeur**, ainsi que par une analyse de vulnérabilités indépendante démontrant la résistance à la pénétration d'attaquants possédant un potentiel d'attaque élémentaire**.

219 Le niveau **EAL4** procure également l'assurance au moyen de la mise en œuvre de contrôles de l'environnement de développement, d'une gestion de configuration **complémentaire** de la TOE **comprenant l'automatisation** et d'éléments de preuve relatifs à des procédures de livraison sûres.

220 **Le présent EAL représente un accroissement significatif de l'assurance par rapport au niveau EAL3 en exigeant plus d'éléments descriptifs issus de la conception, un sous-ensemble de l'implémentation, et des mécanismes ou des procédures perfectionnés qui procurent la confiance dans le fait que la TOE ne sera pas altérée au cours du développement ou de la livraison.**

Classe d'assurance	Composants d'assurance
Gestion de configuration	**ACM_AUT.1 Automatisation partielle de la CM**
	ACM_CAP.4 Aide à la génération et procédures de réception
	ACM_SCP.2 Couverture du suivi des problèmes par la CM
Livraison et exploitation	**ADO_DEL.2 Détection de modifications**
	ADO_IGS.1 Procédures d'installation, de génération et de démarrage
Développement	**ADV_FSP.2 Définition exhaustive des interfaces externes**
	ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité
	ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF
	ADV_LLD.1 Conception de bas niveau descriptive
	ADV_RCR.1 Démonstration de correspondance informelle
	ADV_SPM.1 Modèle informel de politique de sécurité de la TOE
Guides	AGD_ADM.1 Guide de l'administrateur
	AGD_USR.1 Guide de l'utilisateur
Support au cycle de vie	**ALC_DVS.1 Identification des mesures de sécurité**
	ALC_LCD.1 Modèle de cycle de vie défini par le développeur
	ALC_TAT.1 Outils de développement bien définis
Tests	ATE_COV.2 Analyse de la couverture
	ATE_DPT.1 Tests : conception de haut niveau
	ATE_FUN.1 Tests fonctionnels
	ATE_IND.2 Tests indépendants - échantillonnage
Estimation des vulnérabilités	**AVA_MSU.2 Validation de l'analyse**
	AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE
	AVA_VLA.2 Analyse de vulnérabilités indépendante

Tableau 6.5 - Le niveau EAL4

6.2.5 Niveau d'assurance de l'évaluation 5 (EAL5) - conçu à l'aide de méthodes semi-formelles et testé

Objectifs

221 Le niveau EAL5 permet à un développeur d'obtenir l'assurance maximum issue d'une méthode de développement prenant en compte la sécurité basée sur des pratiques de développement commercial rigoureuses renforcées par l'application modérée de techniques spécialisées d'ingénierie de la sécurité. Une telle TOE sera probablement conçue et développée avec l'intention de parvenir au niveau d'assurance EAL5. Il est probable que les coûts supplémentaires imputables aux exigences du niveau EAL5, par rapport à un développement rigoureux effectué sans l'application de techniques spécialisées, ne seront pas très importants.

222 Le niveau EAL5 est par conséquent applicable dans les cas où les développeurs ou les utilisateurs ont besoin d'un niveau élevé de sécurité, déterminé de façon indépendante, pour un développement prévu et où ils exigent une approche rigoureuse du développement qui ne provoque pas des coûts déraisonnables dus à des techniques spécialisées d'ingénierie de la sécurité.

Composants d'assurance

223 Le niveau **EAL5** (voir le tableau **6.6**) procure une assurance consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles, des spécifications complètes des interfaces, des guides, de la conception de haut niveau, de la conception de bas niveau de la TOE et de **l'ensemble** de l'implémentation, afin de comprendre le comportement de sécurité**.** Cette assurance est renforcée par un modèle **formel** de la politique de sécurité de la TOE, **une présentation semi-formelle des spécifications fonctionnelles et de la conception de haut niveau, ainsi que par une démonstration semi-formelle de la correspondance entre elles. Une conception modulaire de la TOE est également exigée**.

224 L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE, par les éléments de preuve des tests du développeur basés sur les spécifications fonctionnelles, la conception de haut niveau **et la conception de bas niveau**, par la confirmation sélective et indépendante des résultats des tests du développeur, par l'analyse de la résistance des fonctions, par des éléments de preuve d'une recherche des vulnérabilités effectuée par le développeur, ainsi que par une analyse de vulnérabilités indépendante démontrant la résistance à la pénétration d'attaquants possédant un potentiel d'attaque **moyen**. **L'analyse inclut également la validation de l'analyse de canaux cachés réalisée par le développeur.**

225 Le niveau **EAL5** procure également l'assurance au moyen de la mise en œuvre de contrôles de l'environnement de développement, d'une gestion de configuration **étendue** de la TOE comprenant l'automatisation et d'éléments de preuve relatifs à des procédures de livraison sûres.

226 **Le présent EAL représente un accroissement significatif de l'assurance par rapport au niveau EAL4 en exigeant des descriptions semi-formelles de la**

conception, l'implémentation complète, une architecture plus structurée (et donc se prêtant mieux à l'analyse), une analyse des canaux cachés, et des mécanismes ou des procédures perfectionnés qui procurent la confiance dans le fait que la TOE ne sera pas altérée au cours du développement.

Classe d'assurance	Composants d'assurance
Gestion de configuration	ACM_AUT.1 Automatisation partielle de la CM
	ACM_CAP.4 Aide à la génération et procédures de réception
	ACM_SCP.3 Couverture des outils de développement par la CM
Livraison et exploitation	ADO_DEL.2 Détection de modifications
	ADO_IGS.1 Procédures d'installation, de génération et de démarrage
Développement	**ADV_FSP.3 Spécifications fonctionnelles semi-formelles**
	ADV_HLD.3 Conception de haut niveau semi-formelle
	ADV_IMP.2 Implémentation de la TSF
	ADV_INT.1 Modularité
	ADV_LLD.1 Conception de bas niveau descriptive
	ADV_RCR.2 Démonstration de correspondance semi-formelle
	ADV_SPM.3 Modèle formel de politique de sécurité de la TOE
Guides	AGD_ADM.1 Guide de l'administrateur
	AGD_USR.1 Guide de l'utilisateur
Support au cycle de vie	ALC_DVS.1 Identification des mesures de sécurité
	ALC_LCD.2 Modèle de cycle de vie normalisé
	ALC_TAT.2 Conformité aux normes d'implémentation
Tests	ATE_COV.2 Analyse de la couverture
	ATE_DPT.2 Tests : conception de bas niveau
	ATE_FUN.1 Tests fonctionnels
	ATE_IND.2 Tests indépendants - échantillonnage
Estimation des vulnérabilités	**AVA_CCA.1 Analyse des canaux cachés**
	AVA_MSU.2 Validation de l'analyse
	AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE
	AVA_VLA.3 Résistance moyenne

Tableau 6.6 - Le niveau EAL5

6.2.6 Niveau d'assurance de l'évaluation 6 (EAL6) - conception vérifiée à l'aide de méthodes semi-formelles et testé

Objectifs

227 Le niveau EAL6 permet aux développeurs d'obtenir une assurance élevée par l'application de techniques d'ingénierie de la sécurité dans un environnement de développement rigoureux afin de produire une TOE de grande qualité destinée à protéger des biens de grande valeur contre des risques significatifs.

228 Le niveau EAL6 est par conséquent applicable au développement de TOE de sécurité devant être utilisées dans des situations où existent des risques élevés et où la valeur des biens à protéger justifie les coûts supplémentaires.

Composants d'assurance

229 Le niveau **EAL6** (voir le tableau **6.7**) procure une assurance consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles, des spécifications complètes des interfaces, des guides, de la conception de haut niveau, de la conception de bas niveau de la TOE et **d'une présentation structurée** de l'implémentation, afin de comprendre le comportement de sécurité**.** Cette assurance est renforcée par un modèle formel de la politique de sécurité de la TOE, une présentation semi-formelle des spécifications fonctionnelles, de la conception de haut niveau **et de la conception de bas niveau,** ainsi que par une démonstration de la correspondance entre elles effectuée de façon semi-formelle. Une conception modulaire **et en couches** de la TOE est également exigée.

230 L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE, par les éléments de preuve des tests du développeur basés sur les spécifications fonctionnelles, la conception de haut niveau et la conception de bas niveau, par la confirmation sélective et indépendante des résultats des tests du développeur, par l'analyse de la résistance des fonctions, par des éléments de preuve d'une recherche des vulnérabilités effectuée par le développeur, ainsi que par une analyse de vulnérabilités indépendante démontrant la résistance à la pénétration d'attaquants possédant un potentiel d'attaque **élevé**. L'analyse inclut également la validation de l'analyse **systématique** de canaux cachés réalisée par le développeur.

231 Le niveau **EAL6** procure également l'assurance au moyen de la mise en œuvre d'un **processus de développement structuré**, de contrôles de l'environnement de développement, d'une gestion de configuration étendue de la TOE comprenant l'automatisation **complète**, et d'éléments de preuve relatifs à des procédures de livraison sûres.

232 **Le présent EAL représente un accroissement significatif de l'assurance par rapport au niveau EAL5 en exigeant une analyse plus étendue, une représentation structurée de l'implémentation, une architecture plus structurée (e.g. décomposition en couches), une analyse de vulnérabilités indépendante plus étendue, une identification systématique des canaux cachés,**

et une gestion de configuration et des contrôles de l'environnement de développement perfectionnés.

Classe d'assurance	Composants d'assurance
Gestion de configuration	**ACM_AUT.2 Automatisation complète de la CM**
	ACM_CAP.5 Support avancé
	ACM_SCP.3 Couverture des outils de développement par la CM
Livraison et exploitation	ADO_DEL.2 Détection de modifications
	ADO_IGS.1 Procédures d'installation, de génération et de démarrage
Développement	**ADV_FSP.3 Spécifications fonctionnelles semi-formelles**
	ADV_HLD.4 Explication semi-formelle de la conception de haut niveau
	ADV_IMP.3 Implémentation structurée de la TSF
	ADV_INT.2 Réduction de la complexité
	ADV_LLD.2 Conception de bas niveau semi-formelle
	ADV_RCR.2 Démonstration de correspondance semi-formelle
	ADV_SPM.3 Modèle formel de politique de sécurité de la TOE
Guides	AGD_ADM.1 Guide de l'administrateur
	AGD_USR.1 Guide de l'utilisateur
Support au cycle de vie	**ALC_DVS.2 Caractère suffisant des mesures de sécurité**
	ALC_LCD.2 Modèle de cycle de vie normalisé
	ALC_TAT.3 Conformité aux normes d'implémentation - toutes parties de la TOE
Tests	**ATE_COV.3 Analyse rigoureuse de la couverture**
	ATE_DPT.2 Tests : conception de bas niveau
	ATE_FUN.2 Tests fonctionnels ordonnés
	ATE_IND.2 Tests indépendants - échantillonnage
Estimation des vulnérabilités	**AVA_CCA.2 Analyse systématique des canaux cachés**
	AVA_MSU.3 Analyse et test des états non sûrs
	AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE
	AVA_VLA.4 Résistance élevée

Tableau 6.7 - Le niveau EAL6

6.2.7 Niveau d'assurance de l'évaluation 7 (EAL7) - conception vérifiée à l'aide de méthodes formelles et testé

Objectifs

233 Le niveau EAL7 est applicable au développement des TOE de sécurité devant être utilisées dans des situations où existent des risques extrêmement élevés ou dans lesquelles la grande valeur des biens justifie des coûts plus importants. L'utilisation pratique du niveau EAL7 est actuellement limitée à des TOE comprenant des fonctionnalités de sécurité extrêmement concentrées qui sont susceptibles de faire l'objet d'une analyse formelle approfondie.

Composants d'assurance

234 Le niveau **EAL7** (voir le tableau **6.8**) procure une assurance consistant en une analyse des fonctions de sécurité, à partir des spécifications fonctionnelles, des spécifications complètes des interfaces, des guides, de la conception de haut niveau, de la conception de bas niveau de la TOE et d'une présentation structurée de l'implémentation, afin de comprendre le comportement de sécurité**.** Cette assurance est renforcée par un modèle formel de la politique de sécurité de la TOE, **une présentation formelle des spécifications fonctionnelles et de la conception de haut niveau**, une présentation semi-formelle de la conception de bas niveau, ainsi que par une démonstration de la correspondance entre elles, effectuée de façon **formelle et** semi-formelle**, quand cela est approprié.** Une conception modulaire en couches et **simple** de la TOE est également exigée.

235 L'analyse est confortée par des tests indépendants des fonctions de sécurité de la TOE, par les éléments de preuve des tests du développeur basés sur les spécifications fonctionnelles, la conception de haut niveau, la conception de bas niveau **et la représentation de l'implémentation**, par la confirmation **complète** et indépendante des résultats des tests du développeur, par l'analyse de la résistance des fonctions, par des éléments de preuve d'une recherche des vulnérabilités effectuée par le développeur, ainsi que par une analyse de vulnérabilités indépendante démontrant la résistance à la pénétration d'attaquants possédant un potentiel d'attaque élevé. L'analyse inclut également la validation de l'analyse systématique de canaux cachés réalisée par le développeur.

236 Le niveau **EAL7** procure également l'assurance au moyen de l'utilisation d'un processus de développement structuré, de contrôles sur l'environnement de développement, d'une gestion de configuration étendue de la TOE permettant une automatisation complète et des éléments de preuve de l'utilisation de procédures de livraison sûres.

237 **Le présent EAL représente un accroissement significatif de l'assurance par rapport au niveau EAL6 en exigeant une analyse plus détaillée utilisant des représentations formelles, des correspondances formelles ainsi que des tests détaillés.**

Classe d'assurance	Composants d'assurance
Gestion de configuration	ACM_AUT.2 Automatisation complète de la CM
	ACM_CAP.5 Support avancé
	ACM_SCP.3 Couverture des outils de développement par la CM
Livraison et exploitation	**ADO_DEL.3 Prévention des modifications**
	ADO_IGS.1 Procédures d'installation, de génération et de démarrage
Développement	**ADV_FSP.4 Spécifications fonctionnelles formelles**
	ADV_HLD.5 Conception de haut niveau formelle
	ADV_IMP.3 Implémentation structurée de la TSF
	ADV_INT.3 Minimisation de la complexité
	ADV_LLD.2 Conception de bas niveau semi-formelle
	ADV_RCR.3 Démonstration de correspondance formelle
	ADV_SPM.3 Modèle formel de politique de sécurité de la TOE
Guides	AGD_ADM.1 Guide de l'administrateur
	AGD_USR.1 Guide de l'utilisateur
Support au cycle de vie	ALC_DVS.2 Caractère suffisant des mesures de sécurité
	ALC_LCD.3 Modèle de cycle de vie mesurable
	ALC_TAT.3 Conformité aux normes d'implémentation - toutes parties de la TOE
Tests	ATE_COV.3 Analyse rigoureuse de la couverture
	ATE_DPT.3 Tests : représentation de l'implémentation
	ATE_FUN.2 Tests fonctionnels ordonnés
	ATE_IND.3 Tests indépendants - totalité
Estimation des vulnérabilités	AVA_CCA.2 Analyse systématique des canaux cachés
	AVA_MSU.3 Analyse et test des états non sûrs
	AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE
	AVA_VLA.4 Résistance élevée

Tableau 6.8 - Le niveau EAL7

7 Classes, familles et composants d’assurance

238 Les sept chapitres suivants donnent les exigences détaillées, présentées par ordre alphabétique, de chacun des composants d’assurance, regroupés par classe et par famille.

8 Classe ACM : Gestion de configuration

239 La gestion de configuration (CM) est une méthode ou un moyen pour établir que les exigences et les spécifications fonctionnelles sont mises en œuvre dans l'implémentation de la TOE. La classe CM satisfait à ces objectifs en exigeant de la discipline et du contrôle dans les processus de raffinement et de modification de la TOE et des informations associées. Les systèmes CM sont mis en place pour garantir l'intégrité des parties de la TOE qu'ils contrôlent, en fournissant une méthode de suivi de tous les changements et en garantissant que tous les changements sont autorisés.

240 La figure 8.1 présente les familles de cette classe et la hiérarchie des composants au sein des familles.

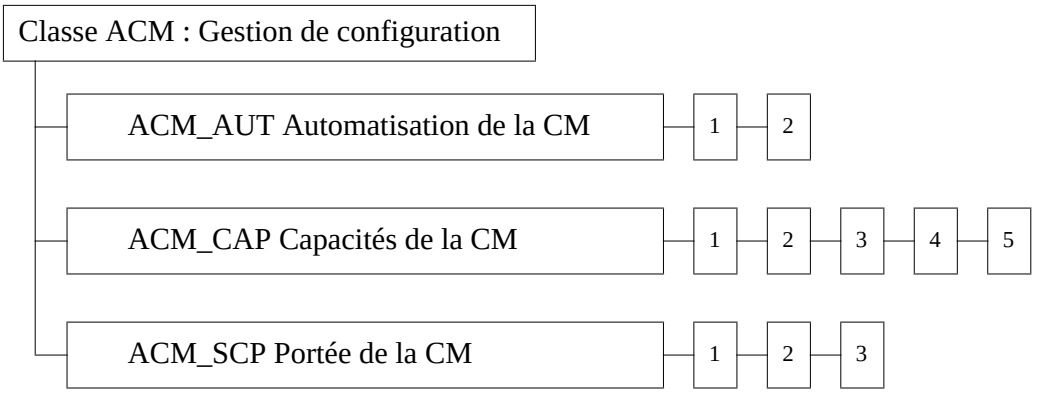

Figure 8.1 - Décomposition de la classe "Gestion de configuration"

8.1 Automatisation de la CM (ACM_AUT)

Objectifs

241 L'introduction d'outils automatisés de CM a pour objectif d'accroître l'efficacité du système de CM. Alors que les systèmes de CM automatisés aussi bien que manuels peuvent être court-circuités, ignorés ou se révéler insuffisants pour empêcher une modification non autorisée, les systèmes automatisés sont moins susceptibles de faire l'objet d'erreurs ou de négligences humaines.

Classement des composants

242 Les composants de cette famille sont classés en fonction de l'ensemble des éléments de configuration qui sont contrôlés par des moyens automatisés.

Notes d'application

243 ACM_AUT.1.1C introduit une exigence relative à la représentation de l'implémentation de la TOE. La représentation de l'implémentation de la TOE est constituée de tous les matériels, logiciels et micro-programmes qui constituent physiquement la TOE. Dans le cas où la TOE n'est constituée que de logiciels, la représentation de l'implémentation peut n'être formée que du code source et du code objet.

244 ACM_AUT.1.2C introduit une exigence selon laquelle le système de CM offre un moyen automatisé de génération de la TOE. Ceci implique que le système de CM offre un moyen automatisé d'assistance pour déterminer que les bons éléments de configuration sont utilisés pour générer la TOE.

245 ACM_AUT.2.5C introduit une exigence selon laquelle le système de CM fournit un moyen automatisé pour établir les changements affectant la TOE par rapport à sa version précédente. Dans le cas où il n'existe pas de version précédente de la TOE, le développeur doit quand même offrir un moyen automatisé pour établir les changements qui auront lieu entre la TOE, et une future version de la TOE.

ACM_AUT.1 Automatisation partielle de la CM

Objectifs

246 Dans des environnements de développement où la représentation de l'implémentation est complexe ou est réalisée par plusieurs développeurs, il est difficile de contrôler les changements sans l'aide d'outils automatisés. En particulier, ces outils automatisés doivent être capables de prendre en compte les nombreux changements qui ont lieu pendant le développement et de garantir que ces changements sont autorisés. L'objectif du présent composant est de garantir que la représentation de l'implémentation est contrôlée par des moyens automatisés.

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ACM_AUT.1.1D **Le développeur doit utiliser un système de CM.**

ACM_AUT.1.2D **Le développeur doit fournir un plan de CM.**

Contenu et présentation des éléments de preuve :

ACM_AUT.1.1C **Le système de CM doit fournir un moyen automatisé ne permettant que les changements autorisés sur la représentation de l'implémentation de la TOE.**

ACM_AUT.1.2C **Le système de CM doit fournir un moyen automatisé de génération de la TOE.**

ACM_AUT.1.3C **Le plan de CM doit décrire les outils automatisés utilisés dans le système de CM.**

ACM_AUT.1.4C **Le plan de CM doit décrire comment les outils automatisés sont utilisés dans le système de CM.**

Tâches de l'évaluateur :

ACM_AUT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ACM_AUT.2 Automatisation complète de la CM

Objectifs

247 Dans des environnements de développement où les éléments de configuration sont complexes ou sont réalisés par plusieurs développeurs, il est difficile de contrôler les changements sans l'aide d'outils automatisés. En particulier, ces outils automatisés doivent être capables de prendre en compte les nombreux changements qui ont lieu pendant le développement et de garantir que ces changements sont autorisés. L'objectif du présent composant est de garantir que tous les éléments de configuration sont contrôlés par des moyens automatisés.

248 La fourniture d'un moyen automatisé pour établir les changements entre différentes versions de la TOE et pour identifier les éléments de configuration qui sont affectés par les modifications d'autres éléments de configuration, aide à déterminer l'impact des changements entre des versions successives de la TOE. Ceci permet alors d'obtenir des informations intéressantes pour déterminer si les changements effectués sur la TOE conduisent à ce que tous les éléments de configuration soient cohérents entre eux.

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ACM_AUT.2.1D Le développeur doit utiliser un système de CM.

ACM_AUT.2.2D Le développeur doit fournir un plan de CM.

Contenu et présentation des éléments de preuve :

ACM_AUT.2.1C Le système CM doit fournir un moyen automatisé ne permettant que les changements autorisés sur la représentation de l'implémentation de la TOE **et sur tous les autres éléments de configuration**.

ACM_AUT.2.2C Le système de CM doit fournir un moyen automatisé de génération de la TOE.

ACM_AUT.2.3C Le plan de CM doit décrire les outils automatisés utilisés dans le système de CM.

ACM_AUT.2.4C Le plan de CM doit décrire comment les outils automatisés sont utilisés dans le système de CM.

ACM_AUT.2.5C **Le système de CM doit fournir un moyen automatisé pour établir les changements entre la TOE et sa version précédente.**

ACM_AUT.2.6C **Le système de CM doit fournir un moyen automatisé pour identifier tous les autres éléments de configuration qui sont affectés par la modification d'un élément de configuration donné.**

Tâches de l'évaluateur :

ACM_AUT.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

8.2 Capacités de la CM (ACM_CAP)

Objectifs

249 Les capacités du système de CM traitent de la probabilité que des modifications accidentelles ou non autorisées d'éléments de configuration puissent se produire. Le système de CM devrait garantir l'intégrité de la TOE depuis les étapes de conception initiales jusqu'aux actions de maintenance ultérieures.

250 Les objectifs de cette famille comprennent les aspects suivants :

a) garantir que la TOE est correcte et complète avant son envoi à l'utilisateur ;

b) garantir qu'aucun élément de configuration n'est omis pendant l'évaluation ;

c) empêcher les modifications, ajouts ou suppression non autorisés d'éléments de configuration de la TOE.

Classement des composants

251 Les composants de cette famille sont classés suivant les capacités du système de CM, la portée de la documentation de CM fournie par le développeur et l'aptitude du développeur à justifier que le système de CM satisfait à ses exigences de sécurité.

Notes d'application

252 ACM_CAP.2 introduit plusieurs éléments qui font référence aux éléments de configuration. La famille ACM_SCP contient des exigences pour que le système de CM suit les éléments de configuration.

253 ACM_CAP.2.3C introduit une exigence pour qu'une liste de configuration soit fournie. La liste de configuration contient tous les éléments de configuration qui sont maintenus par le système de CM.

254 ACM_CAP.2.6C introduit une exigence pour que le système de CM identifie de façon unique tous les éléments de configuration. Ceci implique également que suite à modification, un nouvel identifiant unique est attribué aux éléments de configuration.

255 ACM_CAP.3.8C introduit l'exigence selon laquelle les éléments de preuve doivent démontrer que le système de CM fonctionne conformément au plan de CM. On pourrait citer comme exemples de tels éléments de preuve des copies d'écran ou des traces d'audit issus du fonctionnement du système de CM ou encore une démonstration détaillée du système de CM effectuée par le développeur. L'évaluateur est chargé de déterminer si ces éléments de preuve sont suffisants pour montrer que le système de CM fonctionne conformément au plan de CM.

256 ACM_CAP.3.9C introduit l'exigence que des éléments de preuve soient fournis afin de montrer que tous les éléments de configuration sont maintenus par le système de CM. Comme un élément de configuration fait référence à un élément qui figure dans la liste configuration, cette exigence stipule que tous les éléments de la liste configuration sont maintenus dans le système de CM.

257 ACM_CAP.4.11C introduit l'exigence que le système de CM contribue à la génération de la TOE. Ceci implique que le système de CM fournisse des informations ou un moyen électronique pour aider à déterminer que les bons éléments de configuration sont utilisés pour générer la TOE.

ACM_CAP.1 Numéros de version

Objectifs

258 Une référence unique est exigée pour garantir qu'il n'y a pas d'ambiguïté sur l'exemplaire de la TOE qui fait l'objet de l'évaluation. L'identification de la TOE par sa référence garantit que les utilisateurs de la TOE peuvent être à même de savoir quel exemplaire de la TOE ils utilisent.

Dépendances :

Pas de dépendances.

Tâches du développeur :

ACM_CAP.1.1D **Le développeur doit fournir une référence pour la TOE.**

Contenu et présentation des éléments de preuve :

ACM_CAP.1.1C **La référence de la TOE doit être unique pour chaque version de la TOE.**

ACM_CAP.1.2C **La TOE doit être identifiée par sa référence.**

Tâches de l'évaluateur :

ACM_CAP.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ACM_CAP.2 Éléments de configuration

Objectifs

259 Une référence unique est exigée pour garantir qu'il n'y a pas d'ambiguïté sur l'exemplaire de la TOE qui fait l'objet de l'évaluation. L'identification de la TOE par sa référence garantit que les utilisateurs de la TOE peuvent être à même de savoir quel exemplaire de la TOE ils utilisent.

260 L'identification unique des éléments de configuration conduit à une meilleure compréhension de la composition de la TOE, ce qui aide alors à déterminer quels éléments font l'objet d'exigences d'évaluation pour la TOE.

Dépendances :

Pas de dépendances.

Tâches du développeur :

ACM_CAP.2.1D Le développeur doit fournir une référence pour la TOE.

ACM_CAP.2.2D **Le développeur doit utiliser un système de CM.**

ACM_CAP.2.3D **Le développeur doit fournir une documentation de CM.**

Contenu et présentation des éléments de preuve :

ACM_CAP.2.1C La référence de la TOE doit être unique pour chaque version de la TOE.

ACM_CAP.2.2C La TOE doit être identifiée par sa référence.

ACM_CAP.2.3C **La documentation de CM doit inclure une liste de configuration.**

ACM_CAP.2.4C **La liste de configuration doit décrire les éléments de configuration qui constituent la TOE.**

ACM_CAP.2.5C **La documentation CM doit décrire la méthode utilisée pour identifier de façon unique les éléments de configuration.**

ACM_CAP.2.6C **Le système de CM doit identifier de façon unique tous les éléments de configuration.**

Tâches de l'évaluateur :

ACM_CAP.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ACM_CAP.3 Contrôles des autorisations

Objectifs

261 Une référence unique est exigée pour garantir qu'il n'y a pas d'ambiguïté sur l'exemplaire de la TOE qui fait l'objet de l'évaluation. L'identification de la TOE par sa référence garantit que les utilisateurs de la TOE peuvent être à même de savoir quel exemplaire de la TOE ils utilisent.

262 L'identification unique des éléments de configuration conduit à une meilleure compréhension de la composition de la TOE, ce qui aide alors à déterminer quels éléments font l'objet d'exigences d'évaluation pour la TOE.

263 Offrir des contrôles pour garantir que des modifications non autorisées ne sont pas effectuées sur la TOE et garantir des fonctionnalités et une utilisation correctes du système de CM, aident à maintenir l'intégrité de la TOE.

Dépendances :

ACM_SCP.1 Couverture de la TOE par la CM

ALC_DVS.1 Identification des mesures de sécurité

Tâches du développeur :

ACM_CAP.3.1D Le développeur doit fournir une référence pour la TOE.

ACM_CAP.3.2D Le développeur doit utiliser un système de CM.

ACM_CAP.3.3D Le développeur doit fournir une documentation de CM.

Contenu et présentation des éléments de preuve :

ACM_CAP.3.1C La référence de la TOE doit être unique pour chaque version de la TOE.

ACM_CAP.3.2C La TOE doit être identifiée par sa référence.

ACM_CAP.3.3C La documentation de CM doit inclure une liste de configuration **et un plan de CM**.

ACM_CAP.3.4C La liste de configuration doit décrire les éléments de configuration qui constituent la TOE.

ACM_CAP.3.5C La documentation CM doit décrire la méthode utilisée pour identifier de façon unique les éléments de configuration.

ACM_CAP.3.6C Le système de CM doit identifier de façon unique tous les éléments de configuration.

ACM_CAP.3.7C **Le plan de CM doit décrire comment le système de CM est utilisé.**

ACM_CAP.3.8C **Les éléments de preuve doivent démontrer que le système de CM fonctionne conformément au plan de CM.**

ACM_CAP.3.9C **La documentation de CM doit fournir des éléments de preuve montrant que tous les éléments de configuration ont été et sont maintenus efficacement par le système de CM.**

ACM_CAP.3.10C **Le système de CM doit proposer des mesures permettant que les changements autorisés sur les éléments de configuration.**

Tâches de l'évaluateur :

ACM_CAP.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ACM_CAP.4 Aide à la génération et procédures de réception

Objectifs

264 Une référence unique est exigée pour garantir qu'il n'y a pas d'ambiguïté sur l'exemplaire de la TOE qui fait l'objet de l'évaluation. L'identification de la TOE par sa référence garantit que les utilisateurs de la TOE peuvent être à même de savoir quel exemplaire de la TOE ils utilisent.

265 L'identification unique des éléments de configuration conduit à une meilleure compréhension de la composition de la TOE, ce qui aide alors à déterminer quels éléments font l'objet d'exigences d'évaluation pour la TOE.

266 Offrir des contrôles pour garantir que des modifications non autorisées ne sont pas effectuées sur la TOE et garantir des fonctionnalités et une utilisation correctes du système de CM aident à maintenir l'intégrité de la TOE.

267 Le but des procédures de réception est de confirmer que toute création ou modification d'éléments de configuration est autorisée.

Dépendances :

ACM_SCP.1 Couverture de la TOE par la CM

ALC_DVS.1 Identification des mesures de sécurité

Tâches du développeur :

ACM_CAP.4.1D Le développeur doit fournir une référence pour la TOE.

ACM_CAP.4.2D Le développeur doit utiliser un système de CM.

ACM_CAP.4.3D Le développeur doit fournir une documentation de CM.

Contenu et présentation des éléments de preuve :

ACM_CAP.4.1C La référence de la TOE doit être unique pour chaque version de la TOE.

ACM_CAP.4.2C La TOE doit être identifiée par sa référence.

ACM_CAP.4.3C La documentation de CM doit inclure une liste de configuration, un plan de CM **et un plan de réception.**

ACM_CAP.4.4C La liste de configuration doit décrire les éléments de configuration qui constituent la TOE.

ACM_CAP.4.5C La documentation CM doit décrire la méthode utilisée pour identifier de façon unique les éléments de configuration.

ACM_CAP.4.6C Le système de CM doit identifier de façon unique tous les éléments de configuration.

ACM_CAP.4.7C Le plan de CM doit décrire comment le système de CM est utilisé.

ACM_CAP.4.8C Les éléments de preuve doivent démontrer que le système de CM fonctionne conformément au plan de CM.

ACM_CAP.4.9C **La documentation de CM doit fournir des éléments de preuve montrant que tous les éléments de configuration ont été et sont maintenus efficacement par le système de CM.**

ACM_CAP.4.10C **Le système de CM doit proposer des mesures permettant que les changements autorisés sur les éléments de configuration.**

ACM_CAP.4.11C **Le système de CM doit aider à la génération de la TOE.**

ACM_CAP.4.12C **Le plan de réception doit décrire les procédures utilisées pour accepter des éléments de configuration modifiés ou récemment créés en tant que partie de la TOE.**

Tâches de l'évaluateur :

ACM_CAP.4.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ACM_CAP.5 Support avancé

Objectifs

268 Une référence unique est exigée pour garantir qu'il n'y a pas d'ambiguïté sur l'exemplaire de la TOE qui fait l'objet de l'évaluation. L'identification de la TOE par sa référence garantit que les utilisateurs de la TOE peuvent être à même de savoir quel exemplaire de la TOE ils utilisent.

269 L'identification unique des éléments de configuration conduit à une meilleure compréhension de la composition de la TOE, ce qui aide alors à déterminer quels éléments font l'objet d'exigences d'évaluation pour la TOE.

270 Offrir des contrôles pour garantir que des modifications non autorisées ne sont pas effectuées sur la TOE et garantir des fonctionnalités et une utilisation correctes du système de CM aident à maintenir l'intégrité de la TOE.

271 Le but des procédures de réception est de confirmer que toute création ou modification d'éléments de configuration est autorisée.

272 Les procédures d'intégration aident à garantir que la génération de la TOE à partir d'un ensemble d'éléments de configuration est effectuée correctement et d'une façon autorisée.

273 Le fait d'exiger que le système de CM soit capable d'identifier la copie originale des éléments utilisés pour générer la TOE aide à garantir que l'intégrité de ces

éléments est préservée par les protections techniques, physiques ou procédurales appropriées.

Dépendances :

ACM_SCP.1 Couverture de la TOE par la CM

ALC_DVS.2 Caractère suffisant des mesures de sécurité

Tâches du développeur :

ACM_CAP.5.1D Le développeur doit fournir une référence pour la TOE.

ACM_CAP.5.2D Le développeur doit utiliser un système de CM.

ACM_CAP.5.3D Le développeur doit fournir une documentation de CM.

Contenu et présentation des éléments de preuve :

ACM_CAP.5.1C La référence de la TOE doit être unique pour chaque version de la TOE.

ACM_CAP.5.2C La TOE doit être identifiée par sa référence.

ACM_CAP.5.3C La documentation doit inclure une liste de configuration, un plan de CM, un plan de réception **et des procédures d'intégration.**

ACM_CAP.5.4C La liste de configuration doit décrire les éléments de configuration qui constituent la TOE.

ACM_CAP.5.5C La documentation CM doit décrire la méthode utilisée pour identifier de façon unique les éléments de configuration.

ACM_CAP.5.6C Le système de CM doit identifier de façon unique tous les éléments de configuration.

ACM_CAP.5.7C Le plan de CM doit décrire comment le système de CM est utilisé.

ACM_CAP.5.8C Les éléments de preuve doivent démontrer que le système de CM fonctionne conformément au plan de CM.

ACM_CAP.5.9C La documentation de CM doit fournir des éléments de preuve montrant que tous les éléments de configuration ont été et sont maintenus efficacement par le système de CM.

ACM_CAP.5.10C Le système de CM doit proposer des mesures permettant que les changements autorisés sur les éléments de configuration.

ACM_CAP.5.11C Le système de CM doit aider à la génération de la TOE.

ACM_CAP.5.12C Le plan de réception doit décrire les procédures utilisées pour accepter des éléments de configuration modifiés ou récemment créés en tant que partie de la TOE.

ACM_CAP.5.13C **Les procédures d'intégration doivent décrire comment le système de CM est appliqué dans le processus de fabrication de la TOE.**

ACM_CAP.5.14C **Le système de CM doit exiger que la personne responsable de l'acceptation d'un élément de configuration dans la CM soit différente de la personne qui a développé cet élément.**

ACM_CAP.5.15C **Le système de CM doit identifier clairement les éléments de configuration qui constituent la TSF.**

ACM_CAP.5.16C **Le système de CM doit aider à l'audit de toutes les modifications de la TOE en indiquant au minimum l'auteur, la date et l'heure dans les traces d'audit.**

ACM_CAP.5.17C **Le système de CM doit être capable d'identifier la copie originale de tous les éléments utilisés pour générer le TOE.**

ACM_CAP.5.18C **La documentation de CM doit démontrer que l'utilisation du système de CM, associée aux mesures de sécurité relatives au développement, permet les seuls changements autorisés sur la TOE.**

ACM_CAP.5.19C **La documentation de CM doit démontrer que l'utilisation des procédures d'intégration garantit que la génération de la TOE est effectuée correctement et de façon autorisée.**

ACM_CAP.5.20C **La documentation de CM doit démontrer que le système de CM est suffisant pour garantir que la personne responsable de l'acceptation d'un élément de configuration dans la CM est différente de la personne qui a développé cet élément.**

ACM_CAP.5.21C **La documentation de CM doit justifier que les procédures de réception comprennent une revue adéquate et appropriée des changements sur tous les éléments de configuration.**

Tâches de l'évaluateur :

ACM_CAP.5.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

8.3 Portée de la CM (ACM_SCP)

Objectifs

274 L'objectif de la présente famille est de garantir que tous les éléments de configuration nécessaires à la TOE sont gérés par le système de CM. Ceci aide à garantir que l'intégrité de ces éléments de configuration est protégée grâce aux capacités du système de CM.

275 Les objectifs de cette famille comprennent les aspects suivants :

a) garantir que la représentation de l'implémentation de la TOE est contrôlée ;

b) garantir que toute la documentation nécessaire, y compris les comptes-rendus de problèmes, est contrôlée pendant le développement et l'exploitation ;

c) garantir que les options de configuration (e.g. options de compilation) sont contrôlées ;

d) garantir que les outils de développement sont contrôlés.

Classement des composants

276 Les composants de cette famille sont classés sur la base de ce qui est contrôlé par le système de CM parmi les éléments suivants : la représentation de l'implémentation de la TOE, la documentation de conception, la documentation de test, la documentation de l'utilisateur, la documentation de l'administrateur, la documentation de CM, les anomalies de sécurité et les outils de développement.

Notes d'application

277 ACM_SCP.1.1C introduit l'exigence selon laquelle la représentation de l'implémentation de la TOE doit être contrôlée par le système de CM. La représentation de l'implémentation de la TOE est constituée de tous les matériels, logiciels et micro-programmes qui constituent physiquement la TOE. Dans le cas où la TOE n'est constituée que de logiciels, la représentation de l'implémentation peut n'être formée que de l'ensemble du code source et du code objet.

278 ACM_SCP.1.1C introduit également l'exigence selon laquelle la documentation de CM doit être contrôlée par le système de CM. Ceci inclut le plan de CM, aussi bien que les informations sur la version courante de tous les outils constituant le système de CM.

279 ACM_SCP.2.1C introduit l'exigence selon laquelle les anomalies de sécurité doivent être suivies par le système de CM. Ceci nécessite que les informations relatives aux anomalies de sécurité précédentes et à leur résolution soient maintenues, aussi bien que les détails relatifs aux anomalies de sécurité courantes.

280 ACM_SCP.3.1C introduit l'exigence selon laquelle les outils de développement et autres informations associées doivent être contrôlés par le système de CM. Les langages de programmation et les compilateurs constituent des exemples d'outils de développement. Les informations relatives à la génération de la TOE (tels que les options de compilation, les options d'installation ou de génération et les options de construction) sont des exemples d'informations associées aux outils de développement.

ACM_SCP.1 Couverture de la TOE par la CM

Objectifs

281 Un système de CM ne peut contrôler que les changements des éléments qui sont placés sous son contrôle. Placer sous le contrôle de la CM la représentation de l'implémentation de la TOE, la documentation de conception, de test, de l'utilisateur et de l'administrateur ainsi que la documentation de CM offre l'assurance qu'elles ont été modifiées d'une façon contrôlée avec les autorisations adéquates.

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ACM_SCP.1.1D **Le développeur doit fournir la documentation de CM.**

Contenu et présentation des éléments de preuve :

ACM_SCP.1.1C **La documentation de CM doit montrer que le système de CM contrôle au minimum les éléments suivants : la représentation de l'implémentation de la TOE, la documentation de conception, la documentation de test, la documentation de l'utilisateur, la documentation de l'administrateur et la documentation de CM.**

ACM_SCP.1.2C **La documentation de CM doit décrire comment les éléments de configuration sont contrôlés par le système de CM.**

Tâches de l'évaluateur :

ACM_SCP.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ACM_SCP.2 Couverture du suivi des problèmes par la CM

Objectifs

282 Un système de CM ne peut contrôler que les changements des éléments qui sont placés sous son contrôle. Placer sous le contrôle de la CM la représentation de l'implémentation de la TOE, la documentation de conception, de test, de

l'utilisateur et de l'administrateur ainsi que la documentation de CM offre l'assurance qu'elles ont été modifiées d'une façon contrôlée avec les autorisations adéquates.

283 La possibilité de suivre les anomalies de sécurité grâce à la CM garantit que les rapports relatifs aux anomalies de sécurité ne sont ni perdus ni oubliés et permet au développeur de suivre les anomalies de sécurité jusqu'à leur correction.

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ACM_SCP.2.1D Le développeur doit fournir la documentation de CM.

Contenu et présentation des éléments de preuve :

ACM_SCP.2.1C La documentation de CM doit montrer que le système CM contrôle au minimum les éléments suivants : la représentation de l'implémentation de la TOE, la documentation de conception, la documentation de test, la documentation de l'utilisateur, la documentation de l'administrateur, la documentation de CM **et les anomalies de sécurité.**

ACM_SCP.2.2C La documentation de CM doit décrire comment les éléments de configuration sont contrôlés par le système de CM.

Tâches de l'évaluateur :

ACM_SCP.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ACM_SCP.3 Couverture des outils de développement par la CM

Objectifs

284 Un système de CM ne peut contrôler que les changements des éléments qui sont placés sous son contrôle. Placer sous le contrôle de la CM la représentation de l'implémentation de la TOE, la documentation de conception, de test, de l'utilisateur et de l'administrateur ainsi que la documentation de CM offre l'assurance qu'elles ont été modifiées d'une façon contrôlée avec les autorisations adéquates.

285 La possibilité de suivre les anomalies de sécurité grâce à la CM garantit que les rapports relatifs aux anomalies de sécurité ne sont ni perdus ni oubliés et permet au développeur de suivre les anomalies de sécurité jusqu'à leur correction.

286 Les outils de développement jouent un rôle important en garantissant la production d'une version de la TOE de qualité. C'est pourquoi il est important de contrôler les modifications apportées à ces outils.

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ACM_SCP.3.1D Le développeur doit fournir la documentation de CM.

Contenu et présentation des éléments de preuve :

ACM_SCP.3.1C La documentation de CM doit montrer que le système CM contrôle au minimum les éléments suivants : la représentation de l'implémentation de la TOE, la documentation de conception, la documentation de test, la documentation de l'utilisateur, la documentation de l'administrateur, la documentation CM, les anomalies de sécurité, **ainsi que les outils de développement et les informations associées.**

ACM_SCP.3.2C La documentation de CM doit décrire comment les éléments de configuration sont contrôlés par le système de CM.

Tâches de l'évaluateur :

ACM_SCP.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

9 Classe ADO : Livraison et exploitation

287 La classe “Livraison et exploitation” définit les exigences pour une livraison, une installation, une génération et un démarrage corrects de la TOE.

288 La figure 9.1 montre les familles de cette classe et la hiérarchie des composants au sein des familles.

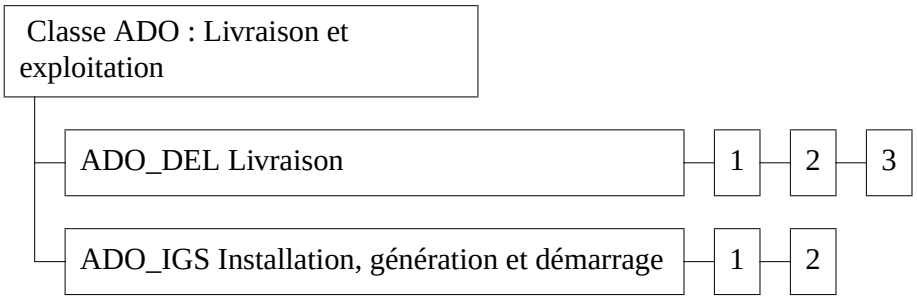

Figure 9.1 -Décomposition de la classe “Livraison et exploitation”

9.1 Livraison (ADO_DEL)

Objectifs

289 Les exigences pour la livraison demandent un contrôle du système ainsi que des réseaux et procédures de distribution qui procurent l'assurance que le destinataire reçoit la TOE que l'expéditeur a voulu envoyer, sans aucune modification. Pour que la livraison soit valide, ce qui est reçu doit correspondre précisément à l'exemplaire original de la TOE, évitant ainsi toute altération de la version reçue ou toute substitution par une version falsifiée.

Classement des composants

290 Les composants de cette famille sont classés sur la base d'exigences croissantes, à l'intention du développeur, pour détecter et empêcher les modifications de la TOE pendant la livraison.

ADO_DEL.1 Procédures de livraison

Dépendances :

Pas de dépendances.

Tâches du développeur :

ADO_DEL.1.1D **Le développeur doit documenter les procédures de livraison à l'utilisateur de la TOE ou de parties de celle-ci.**

ADO_DEL.1.2D **Le développeur doit utiliser les procédures de livraison.**

Contenu et présentation des éléments de preuve :

ADO_DEL.1.1C **La documentation de livraison doit décrire toutes les procédures qui sont nécessaires pour maintenir la sécurité lors de la distribution de versions de la TOE vers le site d'un utilisateur.**

Tâches de l'évaluateur :

ADO_DEL.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences pour le contenu et la présentation des éléments de preuve.**

ADO_DEL.2 Détection de modifications

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ADO_DEL.2.1D Le développeur doit documenter les procédures de livraison à l'utilisateur de la TOE ou de parties de celle-ci.

ADO_DEL.2.2D Le développeur doit utiliser les procédures de livraison.

Contenu et présentation des éléments de preuve :

ADO_DEL.2.1C La documentation de livraison doit décrire toutes les procédures qui sont nécessaires pour maintenir la sécurité lors de la distribution de versions de la TOE vers le site d'un utilisateur.

ADO_DEL.2.2C **La documentation de livraison doit décrire comment les différentes procédures et mesures techniques permettent de détecter les modifications, ou toute différence entre l'exemplaire original du développeur et la version reçue sur le site de l'utilisateur.**

ADO_DEL.2.3C **La documentation de livraison doit décrire comment les différentes procédures permettent de détecter les tentatives de se faire passer pour le développeur (mascarade), même dans les cas où le développeur n'a rien envoyé vers le site de l'utilisateur.**

Tâches de l'évaluateur :

ADO_DEL.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences pour le contenu et la présentation des éléments de preuve.

ADO_DEL.3 Prévention des modifications

Dépendances :

ACM_CAP.3 Contrôles des autorisations

Tâches du développeur :

ADO_DEL.3.1D Le développeur doit documenter les procédures de livraison à l'utilisateur de la TOE ou de parties de celle-ci.

ADO_DEL.3.2D Le développeur doit utiliser les procédures de livraison.

Contenu et présentation des éléments de preuve :

ADO_DEL.3.1C La documentation de livraison doit décrire toutes les procédures qui sont nécessaires pour maintenir la sécurité lors de la distribution de versions de la TOE vers le site d'un utilisateur.

ADO_DEL.3.2C La documentation de livraison doit décrire comment les différentes procédures et mesures techniques permettent **d'empêcher** les modifications, ou toute différence

entre l'exemplaire original du développeur et la version reçue sur le site de l'utilisateur.

ADO_DEL.3.3C **La documentation de livraison doit décrire comment les différentes procédures permettent de détecter les tentatives de se faire passer pour le développeur (mascarade), même dans les cas où le développeur n'a rien envoyé vers le site de l'utilisateur.**

Tâches de l'évaluateur :

ADO_DEL.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences pour le contenu et la présentation des éléments de preuve.

9.2 Installation, génération et démarrage (ADO_IGS)

Objectifs

291 Les procédures d'installation, de génération et de démarrage sont utiles pour garantir que la TOE a été installée, générée et démarrée d'une façon sûre, comme prévu par le développeur. Les exigences relatives à l'installation, la génération et le démarrage font appel à une transition sûre entre la représentation de l'implémentation de la TOE sous contrôle de configuration et son exploitation initiale dans l'environnement de l'utilisateur.

Classement des composants

292 Les composants de cette famille sont classés selon que les options de génération de la TOE sont ou non enregistrées.

Notes d'application

293 Il est reconnu que l'application de ces exigences varie en fonction de facteurs tels que le fait pour la TOE d'être un produit ou un système, le fait qu'elle soit livrée en état de fonctionnement ou le fait qu'elle doive être reconstituée sur le site de l'utilisateur, etc. Pour une TOE donnée, il y aura normalement une séparation des responsabilités relatives à l'installation, la génération et le démarrage entre le développeur de la TOE et le propriétaire de la TOE, mais il y a des exemples où toutes ces activités se déroulent sur un seul site. Par exemple, pour une carte à puce, l'installation, la génération et le démarrage peuvent avoir été réalisés sur le site du développeur de la TOE. Inversement, la TOE pourrait être livrée en tant que système TI sous la forme d'un logiciel pour lequel l'installation, la génération et le démarrage sont réalisés sur le site du propriétaire de la TOE.

294 On pourrait aussi envisager le cas où la TOE est déjà installée lorsque l'évaluation démarre. Dans ce cas, il peut se révéler inapproprié de demander et d'analyser des procédures d'installation.

295 De plus, les exigences relatives à la génération sont applicables seulement pour des TOE qui offrent la possibilité de générer des parties d'une TOE opérationnelle à partir de la représentation de son implémentation.

296 Les procédures d'installation, de génération et de démarrage peuvent figurer dans des documents séparés ou peuvent être regroupées avec d'autres guides d'administration. Les exigences de cette famille d'assurance sont présentées séparément de celles de la famille AGD_ADM, du fait de l'utilisation peu fréquente, voire unique, des procédures d'installation, de génération et de démarrage.

ADO_IGS.1 Procédures d'installation, de génération et de démarrage

Dépendances :

AGD_ADM.1 Guide de l'administrateur

Tâches du développeur :

ADO_IGS.1.1D **Le développeur doit documenter les procédures nécessaires à une installation, une génération et un démarrage sûrs de la TOE.**

Contenu et présentation des éléments de preuve :

ADO_IGS.1.1C **La documentation doit décrire les étapes nécessaires à une installation, une génération et un démarrage sûrs de la TOE.**

Tâches de l'évaluateur :

ADO_IGS.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences pour le contenu et la présentation des éléments de preuve.**

ADO_IGS.1.2E **L'évaluateur doit déterminer que les procédures d'installation, de génération et de démarrage conduisent à une configuration sûre.**

ADO_IGS.2 Fichier de génération

Dépendances :

AGD_ADM.1 Guide de l'administrateur

Tâches du développeur :

ADO_IGS.2.1D Le développeur doit documenter les procédures nécessaires à une installation, une génération et un démarrage sûrs de la TOE.

Contenu et présentation des éléments de preuve :

ADO_IGS.2.1C La documentation doit décrire les étapes nécessaires à une installation, une génération et un démarrage sûrs de la TOE.

ADO_IGS.2.2C **La documentation doit décrire les procédures permettant la création d'une trace contenant les options de génération utilisées pour générer la TOE, de telle sorte qu'il soit possible de déterminer exactement comment et quand la TOE a été générée.**

Tâches de l'évaluateur :

ADO_IGS.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences pour le contenu et la présentation des éléments de preuve.

ADO_IGS.2.2E L'évaluateur doit déterminer que les procédures d'installation, de génération et de démarrage conduisent à une configuration sûre.

10 Classe ADV : Développement

297 La classe “Développement” comprend quatre familles d’exigences pour représenter la TSF à différents niveaux d’abstraction, depuis l’interface fonctionnelle jusqu’à la représentation de l’implémentation. La classe ADV inclut également une famille d’exigences pour la mise en correspondance des différentes représentations de la TSF, allant jusqu’à exiger une démonstration de correspondance depuis la représentation la moins abstraite jusqu’aux spécifications globales de la TOE fournies dans la ST, en passant par toutes les représentations intermédiaires. Il y a de plus une famille d’exigences relative à un modèle de la TSP et aux liens de correspondance entre la TSP, le modèle de TSP et les spécifications fonctionnelles. Enfin, il y a une famille d’exigences relatives à la structure interne de la TSF, qui couvre des aspects tels que la modularité, la décomposition en couches et la minimisation de la complexité.

298 La figure 10.1 montre les familles de cette classe et la hiérarchie des composants au sein des familles.

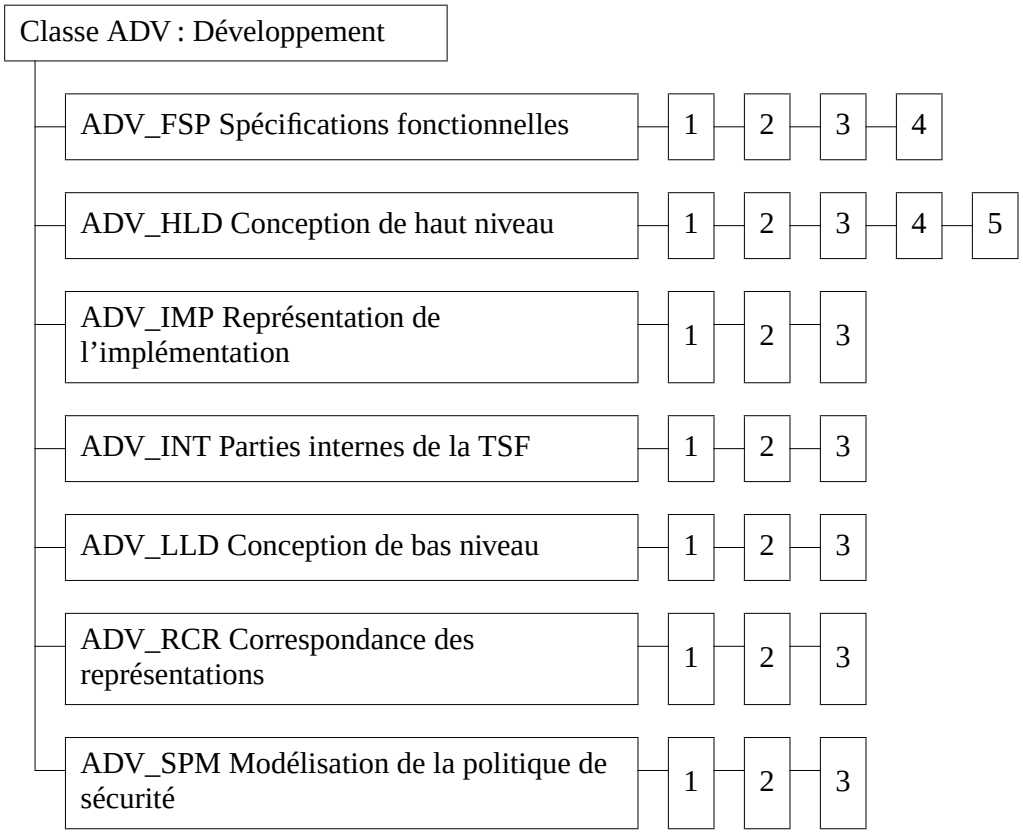

Figure 10.1 - Décomposition de la classe “Développement”

299 Le paradigme évident relatif à ces familles est celui d’une spécification fonctionnelle de la TSF, décomposant la TSF en sous-systèmes, et les sous-systèmes en modules, montrant l’implémentation des modules et démontrant la

correspondance entre toutes ces décompositions qui sont fournies comme éléments de preuve. Cependant, les exigences pour les différentes représentations de la TSF sont réparties en différentes familles pour permettre à l'auteur du PP ou de la ST de spécifier quel sous-ensemble des représentations de la TSF est exigé.

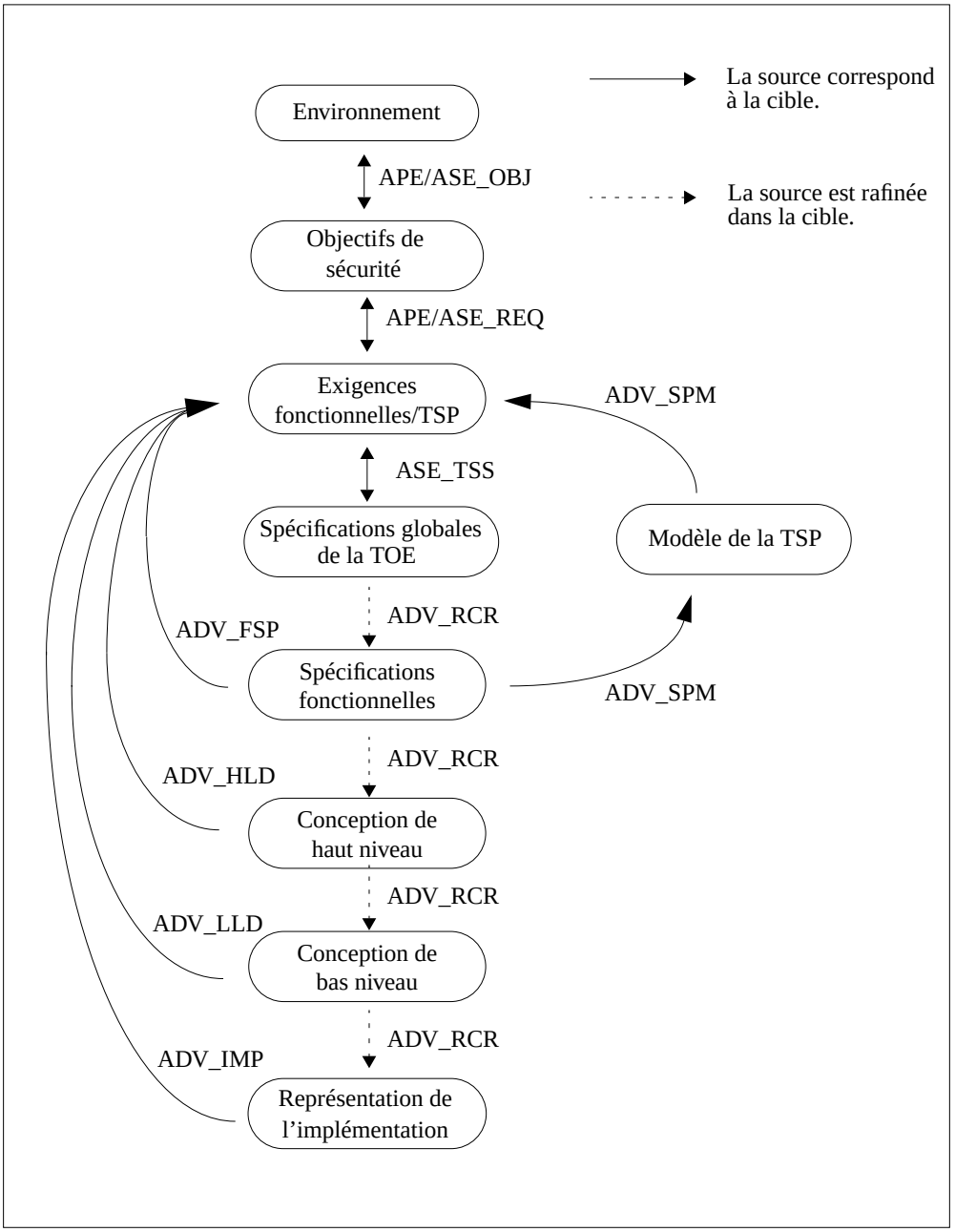

Figure 10.2 - Relations entre les représentations et les exigences de la TOE

300 La figure 10.2 indique les relations entre les différentes représentations de la TSF et les objectifs et les exigences qu'elles sont censées remplir. Comme l'indique la

figure, les classes APE et ASE définissent les exigences relatives à la correspondance entre les exigences fonctionnelles et les objectifs de sécurité, ainsi qu'entre les objectifs de sécurité et l'environnement prévu pour la TOE. La classe ASE définit également les exigences relatives à la correspondance entre, d'une part les objectifs de sécurité et les exigences fonctionnelles, et d'autre part les exigences fonctionnelles et les spécifications globales de la TOE.

301 Les exigences relatives à toutes les autres correspondances montrées dans la figure 10.2 sont définies dans la classe ADV. La famille ADV_SPM définit les exigences relatives à la correspondance entre la TSP et le modèle de la TSP, ainsi qu'entre le modèle de la TSP et les spécifications fonctionnelles. La famille ADV_RCR définit les exigences relatives à la correspondance entre toutes les représentations de la TSF disponibles, depuis les spécifications globales de la TOE jusqu'à la représentation de l'implémentation. Enfin, chaque famille d'assurance spécifique à une représentation de la TSF (i.e. ADV_FSP, ADV_HLD, ADV_LLD et ADV_IMP) définit les exigences relatives aux relations entre cette représentation de la TSF et les exigences fonctionnelles, ce qui aide à garantir que l'ensemble des exigences fonctionnelles de sécurité de la TOE a été couvert. L'analyse de traçabilité doit toujours être menée en partant du plus haut niveau de représentation de la TSF et en descendant à travers chaque niveau de représentation de la TSF fourni. Les CC rendent cette exigence de traçabilité via des dépendances sur la famille ADV_RCR. La famille ADV_INT n'est pas représentée dans cette figure puisqu'elle concerne la structure interne de la TSF et n'est reliée qu'indirectement au processus de raffinement des représentations de la TSF.

Notes d'application

302 La politique de sécurité de la TOE (TSP) est l'ensemble des règles qui définissent comment les ressources sont gérées, protégées et distribuées au sein d'une TOE, exprimée par les exigences fonctionnelles de sécurité de la TOE. Le développeur n'est pas obligé de fournir explicitement une TSP puisque celle-ci est exprimée par les exigences fonctionnelles de sécurité de la TOE, par une combinaison de politiques d'une fonction de sécurité (SFP) et par les autres éléments d'exigences individuels.

303 Les fonctions de sécurité de la TOE (TSF) sont toutes les parties de la TOE sur lesquelles on s'appuie pour l'application de la TSP. La TSF inclut aussi bien les fonctions directement dédiées à l'application de la TSP que celles qui, bien que n'étant pas directement dédiées à l'application de la TSP, contribuent à la mise en œuvre de la TSP d'une façon plus indirecte.

304 Bien que les exigences de la famille ASE_TSS et de plusieurs familles de cette classe demandent différentes représentations de la TSF, il n'est pas absolument nécessaire que chaque représentation de la TSF fasse l'objet d'un document séparé. En fait, il peut arriver qu'un document unique satisfasse toutes les exigences de documentation pour plus d'une représentation de la TSF, puisque ce sont les informations relatives à chacune de ces représentations qui sont exigées plutôt que la structure du document qui en résulte. Dans les cas où de multiples représentations de la TSF sont rassemblées dans un seul document, le développeur devrait indiquer quels documents satisfont quelles exigences.

305 Trois types de style de spécification sont requis par cette classe : informel, semi-formel et formel. Les spécifications fonctionnelles, la conception de haut niveau, la conception de bas niveau et les modèles de TSP seront écrits dans un ou plusieurs de ces styles de spécification. L'utilisation d'un niveau croissant de formalisme réduit l'ambiguïté de ces spécifications.

306 Une spécification informelle est écrite en langage naturel. L'expression "langage naturel" est utilisée ici dans le sens d'une communication dans tout langage parlé courant (e.g. allemand, anglais, français, hollandais). Une spécification informelle n'est l'objet d'aucune restriction spéciale ou relative à la notation autres que celles exigées en tant que conventions ordinaires pour ce langage (e.g. grammaire et syntaxe). Bien qu'aucune restriction de notation ne s'applique, la spécification informelle doit cependant définir la signification de termes qui sont utilisés dans un contexte autre que celui accepté dans le cadre d'un usage normal.

307 Une spécification semi-formelle est écrite dans un langage à syntaxe restreinte et est typiquement complétée par des explications (en langage informel). Le langage à syntaxe réduite peut être un langage naturel avec des structures de phrase restreintes et des mots clés ayant une signification particulière, ou bien il peut être constitué de diagrammes (e.g. diagrammes de flux de données, diagrammes de transitions d'états, diagrammes entités-relations, diagrammes de structures de données, diagramme de structures de processus ou de programme). Que le langage soit basé sur des diagrammes ou sur le langage naturel, un ensemble de conventions doit être fourni pour définir les restrictions relatives à la syntaxe.

308 Une spécification formelle est écrite avec des notations basées sur des concepts mathématiques bien établis et est typiquement complétée par des explications (en langage informel). Ces concepts mathématiques sont utilisés pour définir la syntaxe et la sémantique des notations, ainsi que les règles de preuve qui fondent le raisonnement logique. Les règles syntaxiques et sémantiques relatives à une notation formelle devraient définir comment reconnaître les constructions de façon non ambiguë et déterminer leur sens. L'impossibilité de formuler des contradictions doit pouvoir être prouvée, et toutes les règles relatives aux notations doivent être définies ou référencées.

309 Une assurance significative peut être obtenue en garantissant la traçabilité de la TSF avec chacune de ses représentations et en garantissant que le modèle de TSP correspond aux spécifications fonctionnelles. La famille ADV_RCR contient des exigences pour la mise en correspondance des différentes représentations de la TSF, et la famille ADV_SPM contient des exigences pour la mise en correspondance du modèle de la TSP avec les spécifications fonctionnelles. Une correspondance peut prendre la forme d'une démonstration informelle, d'une démonstration semi-formelle ou d'une preuve formelle.

310 Lorsqu'une démonstration informelle de correspondance est requise, cela signifie que seule une correspondance élémentaire est requise. Les méthodes de mise en correspondance comprennent par exemple l'utilisation d'un tableau bidimensionnel dont les entrées indiquent les correspondances ou encore l'utilisation de notations appropriées pour les diagrammes de conception. Des pointeurs et des renvois à d'autres documents peuvent également être utilisés.

311 Une démonstration semi-formelle de correspondance nécessite une approche structurée pour l'analyse de correspondance. Cette approche devrait réduire les ambiguïtés qui pourraient exister dans une correspondance informelle, en limitant les interprétations possibles des termes décrivant les correspondances. Des pointeurs et des renvois à d'autres documents peuvent être utilisés.

312 Une preuve formelle de correspondance nécessite que des concepts mathématiques biens établis soient utilisés pour définir la syntaxe et la sémantique des notations formelles ainsi que les règles de preuves qui fondent le raisonnement logique. Les propriétés de sécurité doivent pouvoir être exprimées dans le langage de spécification formel et il doit pouvoir être montré que ces propriétés de sécurité sont satisfaites par la spécification formelle. Des pointeurs et des renvois à d'autres documents peuvent également être utilisés.

313 Les éléments ADV_RCR.*.1C exigent que le développeur fournisse la preuve, pour chaque paire adjacente de représentations de la TSF, que toutes les fonctionnalités de sécurité pertinentes au niveau de représentation de la TSF le plus abstrait sont raffinées dans le niveau de représentation de la TSF le moins abstrait. Les éléments ADV_FSP.*.2E, ADV_HLD.*.2E, ADV_LLD.*.2E et ADV_IMP.*.2E exigent de la part de l'évaluateur qu'il détermine que la TSF représentée par cette famille d'exigences est une instantiation précise et complète des exigences fonctionnelles de sécurité de la TOE. Pour déterminer qu'une représentation de la TSF est une instantiation précise et complète des exigences fonctionnelles de sécurité de la TOE, il est prévu que l'évaluateur utilise les éléments de preuve fournis par le développeur dans ADV_RCR.*.1C comme données d'entrée pour sa détermination. Le processus par étapes qui consiste à établir une correspondance entre les exigences fonctionnelles de sécurité de la TOE et chaque représentation successive de la TSF en ordre décroissant, procurera pour finir une plus grande assurance dans le fait que la représentation de la TSF la moins abstraite correspond aux exigences fonctionnelles de sécurité de la TOE, ce qui est le but ultime de cette classe. Si l'évaluateur ne fait pas de démonstration de correspondance en remontant des niveaux de représentation intermédiaires de la TSF vers les exigences fonctionnelles de sécurité de la TOE, alors essayer de démontrer la correspondance entre le plus bas niveau d'abstraction de représentation de la TSF et les exigences fonctionnelles de sécurité de la TOE peut représenter un pas trop important pour être réalisé correctement. Enfin, selon l'ensemble des représentations de la TSF exigées, il est tout à fait possible que la conception de bas niveau, la conception de haut niveau ou même les spécifications fonctionnelles constituent le plus bas niveau d'abstraction de la représentation de la TSF qui soit fourni.

10.1 Spécifications fonctionnelles (ADV_FSP)

Objectifs

314 Les spécifications fonctionnelles représentent une description de haut niveau de l'interface visible par l'utilisateur et du comportement de la TSF. Il s'agit d'une instantiation des exigences fonctionnelles de sécurité de la TOE. Les spécifications fonctionnelles doivent montrer que toutes les exigences fonctionnelles de sécurité sont traitées.

Classement des composants

315 Les composants de cette famille sont classés sur la base du degré de formalisme exigé pour les spécifications fonctionnelles et du niveau de détail fourni pour les interfaces externes de la TSF.

Notes d'application

316 Les éléments ADV_FSP.*.2E de cette famille définissent l'exigence qui impose à l'évaluateur de déterminer que les spécifications fonctionnelles sont une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE. Ceci fournit une correspondance directe entre les exigences fonctionnelles de sécurité de la TOE et les spécifications fonctionnelles, en complément des correspondances deux à deux exigées par la famille ADV_RCR. On attend de l'évaluateur qu'il utilise les éléments de preuve fournis dans ADV_RCR comme données d'entrée pour cette détermination, et l'exigence relative à la complétude est censée être liée au niveau d'abstraction des spécifications fonctionnelles.

317 Pour ADV_FSP.1.3C, il est prévu que suffisamment d'informations soient fournies dans les spécifications fonctionnelles pour comprendre comment les exigences fonctionnelles de sécurité de la TOE ont été prises en compte, et pour permettre la spécification de tests reflétant les exigences fonctionnelles de sécurité de la TOE énoncées dans la ST. Ces tests ne couvriront pas nécessairement toutes les valeurs de réponse possibles et tous les messages d'erreur qui pourraient être générés à l'interface, mais les informations fournies devraient pouvoir indiquer clairement les conséquences d'une utilisation réussie d'une interface ainsi que les cas les plus courants de défaillances.

318 ADV_FSP.2.3C introduit l'exigence d'une présentation complète de l'interface fonctionnelle. Ainsi seront fournis les détails nécessaires à un test approfondi de la TOE ainsi qu'à l'estimation des vulnérabilités.

319 Concernant le niveau de formalisme des spécifications fonctionnelles, les styles informel, semi-formel et formel sont considérés comme étant hiérarchiques par nature. Ainsi, ADV_FSP.1.1C et ADV_FSP.2.1C peuvent aussi être satisfaits par des spécifications fonctionnelles semi-formelles ou formelles, sous réserve qu'elles soient complétées lorsque cela est nécessaire par des explications en langage informel. De plus, ADV_FSP.3.1C peut également être satisfait par une spécification fonctionnelle formelle.

ADV_FSP.1 Spécifications fonctionnelles informelles

Dépendances :

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_FSP.1.1D **Le développeur doit fournir des spécifications fonctionnelles.**

Contenu et présentation des éléments de preuve :

ADV_FSP.1.1C **Les spécifications fonctionnelles doivent décrire la TSF et ses interfaces externes dans un style informel.**

ADV_FSP.1.2C **Les spécifications fonctionnelles doivent avoir une cohérence interne.**

ADV_FSP.1.3C **Les spécifications fonctionnelles doivent décrire le but et le mode d'emploi de toutes les interfaces externes de la TSF, en fournissant, lorsque cela est approprié, les détails sur les effets, les exceptions et les messages d'erreur.**

ADV_FSP.1.4C **Les spécifications fonctionnelles doivent représenter complètement la TSF.**

Tâches de l'évaluateur :

ADV_FSP.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_FSP.1.2E **L'évaluateur doit déterminer que les spécifications fonctionnelles constituent une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_FSP.2 Définition exhaustive des interfaces externes

Dépendances :

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_FSP.2.1D Le développeur doit fournir des spécifications fonctionnelles.

Contenu et présentation des éléments de preuve :

ADV_FSP.2.1C Les spécifications fonctionnelles doivent décrire la TSF et ses interfaces externes dans un style informel.

ADV_FSP.2.2C Les spécifications fonctionnelles doivent avoir une cohérence interne.

ADV_FSP.2.3C Les spécifications fonctionnelles doivent décrire le but et le mode d'emploi de toutes les interfaces externes de la TSF, en fournissant, lorsque cela est approprié, les détails **complets** sur **tous** les effets, les exceptions et les messages d'erreur**.**

ADV_FSP.2.4C Les spécifications fonctionnelles doivent représenter complètement la TSF.

ADV_FSP.2.5C **Les spécifications fonctionnelles doivent comprendre un argumentaire justifiant que la TSF est complètement représentée.**

Tâches de l'évaluateur :

ADV_FSP.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_FSP.2.2E **L'évaluateur doit déterminer que les spécifications fonctionnelles constituent une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_FSP.3 Spécifications fonctionnelles semi-formelles

Dépendances :

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_FSP.3.1D Le développeur doit fournir des spécifications fonctionnelles.

Contenu et présentation des éléments de preuve :

ADV_FSP.3.1C Les spécifications fonctionnelles doivent décrire la TSF et ses interfaces externes dans un style **semi-formel, complété si nécessaire par des explications en langage informel.**

ADV_FSP.3.2C Les spécifications fonctionnelles doivent avoir une cohérence interne.

ADV_FSP.3.3C **Les spécifications fonctionnelles doivent décrire le but et le mode d'emploi de toutes les interfaces externes de la TSF, en fournissant, lorsque cela est approprié, les détails complets sur tous les effets, les exceptions et les messages d'erreur.**

ADV_FSP.3.4C Les spécifications fonctionnelles doivent représenter complètement la TSF.

ADV_FSP.3.5C Les spécifications fonctionnelles doivent comprendre un argumentaire justifiant que la TSF est complètement représentée.

Tâches de l'évaluateur :

ADV_FSP.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_FSP.3.2E **L'évaluateur doit déterminer que les spécifications fonctionnelles constituent une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_FSP.4 Spécifications fonctionnelles formelles

Dépendances :

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_FSP.4.1D Le développeur doit fournir des spécifications fonctionnelles.

Contenu et présentation des éléments de preuve :

ADV_FSP.4.1C Les spécifications fonctionnelles doivent décrire la TSF et ses interfaces externes dans un style **formel**, complété si nécessaire par des explications en langage informel.

ADV_FSP.4.2C Les spécifications fonctionnelles doivent avoir une cohérence interne.

ADV_FSP.4.3C Les spécifications fonctionnelles doivent décrire le but et le mode d'emploi de toutes les interfaces externes de la TSF, en fournissant, lorsque cela est approprié, les détails complets sur tous les effets, les exceptions et les messages d'erreur.

ADV_FSP.4.4C Les spécifications fonctionnelles doivent représenter complètement la TSF.

ADV_FSP.4.5C Les spécifications fonctionnelles doivent comprendre un argumentaire justifiant que la TSF est complètement représentée.

Tâches de l'évaluateur :

ADV_FSP.4.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_FSP.4.2E L'évaluateur doit déterminer que les spécifications fonctionnelles constituent une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.

10.2 Conception de haut niveau (ADV_HLD)

Objectifs

320 La conception de haut niveau d'une TOE apporte une description de la TSF en termes d'éléments structurels principaux (i.e. sous-systèmes) et relie ces éléments aux fonctions qu'ils remplissent. Les exigences relatives à la conception de haut niveau sont censées apporter l'assurance que la TOE fournit une architecture appropriée pour implémenter les exigences fonctionnelles de sécurité de la TOE.

321 La conception de haut niveau est un raffinement des spécifications fonctionnelles en sous-systèmes. Pour chaque sous-système de la TSF, la conception de haut niveau décrit sa finalité et sa fonction, et identifie les fonctions de sécurité contenues dans le sous-système. Les relations entre tous les sous-systèmes sont également définies dans la conception de haut niveau. Ces relations seront représentées comme interfaces externes pour les flux de données, flux de contrôle, etc., lorsque cela est approprié.

Classement des composants

322 Les composants de cette famille sont classés sur la base du degré de formalisme qui est exigé pour la description de la conception de haut niveau et du degré de détail fourni pour les spécifications de l'interface.

Notes d'application

323 Le développeur est censé décrire l'architecture de la TSF en termes de sous-systèmes. Le terme "sous-système" est utilisé ici pour exprimer l'idée d'une décomposition de la TSF en un nombre relativement restreint de parties. Bien que le développeur ne soit pas obligé d'avoir effectivement des "sous-systèmes", on attend de lui qu'il propose un niveau de décomposition similaire. Par exemple, une architecture peut être décomposée de plusieurs façons similaires en utilisant des "couches", des "domaines" ou des "serveurs".

324 Le terme "fonctionnalité de sécurité" est utilisé pour représenter l'ensemble des opérations qu'un sous-système réalise comme contribution aux fonctions de sécurité mises en œuvre par la TOE. Cette distinction est faite car les structures constituant l'architecture, tels que les sous-systèmes et les modules, ne sont pas nécessairement liées à des fonctions de sécurité spécifiques. Bien qu'un sous-système donné puisse correspondre directement à une fonction de sécurité, ou même à plusieurs fonctions de sécurité, il est aussi possible que plusieurs sous-systèmes doivent être combinés pour implémenter une seule fonction de sécurité.

325 Le terme "sous-système dédié à la mise en œuvre de la TSP" se réfère à un sous-système qui contribue à la mise en œuvre de la TSP, soit directement, soit indirectement.

326 Les éléments ADV_HLD.*.2E de cette famille définissent l'exigence qui impose à l'évaluateur de déterminer que la conception de haut niveau est une instantiation

correcte et complète des exigences fonctionnelles de sécurité de la TOE. Cela procure une correspondance directe entre les exigences fonctionnelles de sécurité de la TOE et la conception de haut niveau, en complément des correspondances deux à deux exigées par la famille ADV_RCR. On attend de l'évaluateur qu'il utilise les éléments de preuve fournis dans ADV_RCR comme données d'entrée pour sa détermination, et l'exigence de complétude est censée être liée le niveau d'abstraction de la conception de haut niveau.

327 ADV_HLD.3.8C introduit l'exigence d'une présentation complète des interfaces des sous-systèmes. Ainsi seront fournis les détails nécessaires à un test approfondi de la TOE (en utilisant les composants de ATE_DPT) ainsi qu'à l'estimation des vulnérabilités

328 Concernant le niveau de formalisme de la conception de haut niveau, les styles informel, semi-formel et formel sont considérés comme étant hiérarchiques par nature. Ainsi, ADV_HLD.1.1C et ADV_HLD.2.1C peuvent aussi être satisfaits par une conception de haut niveau semi-formelle ou formelle et ADV_HLD.3.1C et ADV_HLD.4.1C peuvent également être satisfaits par une conception de haut niveau formelle.

ADV_HLD.1 Conception de haut niveau descriptive

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_HLD.1.1D **Le développeur doit fournir la conception de haut niveau de la TSF.**

Contenu et présentation des éléments de preuve :

ADV_HLD.1.1C **La présentation de la conception de haut niveau doit être en style informel.**

ADV_HLD.1.2C **La conception de haut niveau doit avoir une cohérence interne.**

ADV_HLD.1.3C **La conception de haut niveau doit décrire la structure de la TSF en termes de sous-systèmes.**

ADV_HLD.1.4C **La conception de haut niveau doit décrire les fonctionnalités de sécurité fournies par chaque sous-système de la TSF.**

ADV_HLD.1.5C **La conception de haut niveau doit identifier tout matériel, micro-programme ou logiciel sous-jacent exigé par la TSF, et présenter les fonctions fournies par le mécanisme de protection de soutien implémenté dans ce matériel, micro-programme ou logiciel.**

ADV_HLD.1.6C **La conception de haut niveau doit identifier toutes les interfaces des sous-systèmes de la TSF.**

ADV_HLD.1.7C **La conception de haut niveau doit identifier les interfaces des sous-systèmes de la TSF qui sont visibles de l'extérieur.**

Tâches de l'évaluateur :

ADV_HLD.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_HLD.1.2E **L'évaluateur doit déterminer que la conception de haut niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_HLD.2.1D Le développeur doit fournir la conception de haut niveau de la TSF.

Contenu et présentation des éléments de preuve :

ADV_HLD.2.1C La présentation de la conception de haut niveau doit être en style informel.

ADV_HLD.2.2C La conception de haut niveau doit avoir une cohérence interne.

ADV_HLD.2.3C La conception de haut niveau doit décrire la structure de la TSF en termes de sous-systèmes.

ADV_HLD.2.4C La conception de haut niveau doit décrire les fonctionnalités de sécurité fournies par chaque sous-système de la TSF.

ADV_HLD.2.5C La conception de haut niveau doit identifier tout matériel, micro-programme ou logiciel sous-jacent exigé par la TSF, et présenter les fonctions fournies par le mécanisme de protection de soutien implémenté dans ce matériel, micro-programme ou logiciel.

ADV_HLD.2.6C **La conception de haut niveau doit identifier toutes les interfaces des sous-systèmes de la TSF.**

ADV_HLD.2.7C La conception de haut niveau doit identifier les interfaces des sous-systèmes de la TSF qui sont visibles de l'extérieur.

ADV_HLD.2.8C **La conception de haut niveau doit décrire le but et le mode d'emploi de toutes les interfaces des sous-systèmes de la TSF, en fournissant, lorsque cela est approprié, les détails sur les effets, les exceptions et les messages d'erreur.**

ADV_HLD.2.9C **La conception de haut niveau doit décrire la séparation de la TOE entre les sous-systèmes dédiés à la mise en œuvre de la TSP et les autres sous-systèmes.**

Tâches de l'évaluateur :

ADV_HLD.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_HLD.2.2E **L'évaluateur doit déterminer que la conception de haut niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_HLD.3 Conception de haut niveau semi-formelle

Dépendances :

ADV_FSP.3 Spécifications fonctionnelles semi-formelles

ADV_RCR.2 Démonstration de correspondance semi-formelle

Tâches du développeur :

ADV_HLD.3.1D Le développeur doit fournir la conception de haut niveau de la TSF.

Contenu et présentation des éléments de preuve :

ADV_HLD.3.1C La présentation de la conception de haut niveau doit être en style **semi-formel**.

ADV_HLD.3.2C La conception de haut niveau doit avoir une cohérence interne.

ADV_HLD.3.3C La conception de haut niveau doit décrire la structure de la TSF en termes de sous-systèmes.

ADV_HLD.3.4C La conception de haut niveau doit décrire les fonctionnalités de sécurité fournies par chaque sous-système de la TSF.

ADV_HLD.3.5C La conception de haut niveau doit identifier tout matériel, micro-programme ou logiciel sous-jacent exigé par la TSF, et présenter les fonctions fournies par le mécanisme de protection de soutien implémenté dans ce matériel, micro-programme ou logiciel.

ADV_HLD.3.6C La conception de haut niveau doit identifier toutes les interfaces des sous-systèmes de la TSF.

ADV_HLD.3.7C La conception de haut niveau doit identifier les interfaces des sous-systèmes de la TSF qui sont visibles de l'extérieur.

ADV_HLD.3.8C La conception de haut niveau doit décrire le but et le mode d'emploi de toutes les interfaces des sous-systèmes de la TSF, en fournissant, lorsque cela est approprié, les détails **complets** sur **tous** les effets, les exceptions et les messages d'erreur.

ADV_HLD.3.9C **La conception de haut niveau doit décrire la séparation de la TOE entre les sous-systèmes dédiés à la mise en œuvre de la TSP et les autres sous-systèmes.**

Tâches de l'évaluateur :

ADV_HLD.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_HLD.3.2E **L'évaluateur doit déterminer que la conception de haut niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_HLD.4 Explication semi-formelle de la conception de haut niveau

Dépendances :

ADV_FSP.3 Spécifications fonctionnelles semi-formelles

ADV_RCR.2 Démonstration de correspondance semi-formelle

Tâches du développeur :

ADV_HLD.4.1D Le développeur doit fournir la conception de haut niveau de la TSF.

Contenu et présentation des éléments de preuve :

ADV_HLD.4.1C **La présentation de la conception de haut niveau doit être en style semi-formel.**

ADV_HLD.4.2C La conception de haut niveau doit avoir une cohérence interne.

ADV_HLD.4.3C La conception de haut niveau doit décrire la structure de la TSF en termes de sous-systèmes.

ADV_HLD.4.4C La conception de haut niveau doit décrire les fonctionnalités de sécurité fournies par chaque sous-système de la TSF.

ADV_HLD.4.5C La conception de haut niveau doit identifier tout matériel, micro-programme ou logiciel sous-jacent exigé par la TSF, et présenter les fonctions fournies par le mécanisme de protection de soutien implémenté dans ce matériel, micro-programme ou logiciel.

ADV_HLD.4.6C La conception de haut niveau doit identifier toutes les interfaces des sous-systèmes de la TSF.

ADV_HLD.4.7C La conception de haut niveau doit identifier les interfaces des sous-systèmes de la TSF qui sont visibles de l'extérieur.

ADV_HLD.4.8C La conception de haut niveau doit décrire le but et le mode d'emploi de toutes les interfaces des sous-systèmes de la TSF, en fournissant, lorsque cela est approprié, les détails complets sur tous les effets, les exceptions et les messages d'erreur.

ADV_HLD.4.9C La conception de haut niveau doit décrire la séparation de la TOE entre les sous-systèmes dédiés à la mise en œuvre de la TSP et les autres sous-systèmes.

ADV_HLD.4.10C **La conception de haut niveau doit justifier que les moyens identifiés pour réaliser la séparation, y compris tous les mécanismes de protection, sont suffisants pour garantir une séparation claire et effective entre les fonctions qui sont dédiées à l'application de la TSP et celles qui ne le sont pas.**

ADV_HLD.4.11C **La conception de haut niveau doit justifier que les mécanismes de la TSF sont suffisants pour implémenter les fonctions de sécurité identifiées dans la conception de haut niveau.**

Tâches de l'évaluateur :

ADV_HLD.4.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_HLD.4.2E L'évaluateur doit déterminer que la conception de haut niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.

ADV_HLD.5 Conception de haut niveau formelle

Dépendances :

ADV_FSP.4 Spécifications fonctionnelles formelles

ADV_RCR.3 Démonstration de correspondance formelle

Tâches du développeur :

ADV_HLD.5.1D Le développeur doit fournir la conception de haut niveau de la TSF.

Contenu et présentation des éléments de preuve :

ADV_HLD.5.1C La présentation de la conception de haut niveau doit être en style **formel**.

ADV_HLD.5.2C La conception de haut niveau doit avoir une cohérence interne.

ADV_HLD.5.3C La conception de haut niveau doit décrire la structure de la TSF en termes de sous-systèmes.

ADV_HLD.5.4C La conception de haut niveau doit décrire les fonctionnalités de sécurité fournies par chaque sous-système de la TSF.

ADV_HLD.5.5C La conception de haut niveau doit identifier tout matériel, micro-programme ou logiciel sous-jacent exigé par la TSF, et présenter les fonctions fournies par le mécanisme de protection de soutien implémenté dans ce matériel, micro-programme ou logiciel.

ADV_HLD.5.6C La conception de haut niveau doit identifier toutes les interfaces des sous-systèmes de la TSF.

ADV_HLD.5.7C La conception de haut niveau doit identifier les interfaces des sous-systèmes de la TSF qui sont visibles de l'extérieur.

ADV_HLD.5.8C La conception de haut niveau doit décrire le but et le mode d'emploi de toutes les interfaces des sous-systèmes de la TSF, en fournissant, lorsque cela est approprié, les détails complets sur tous les effets, les exceptions et les messages d'erreur.

ADV_HLD.5.9C La conception de haut niveau doit décrire la séparation de la TOE entre les sous-systèmes dédiés à la mise en œuvre de la TSP et les autres sous-systèmes.

ADV_HLD.5.10C La conception de haut niveau doit justifier que les moyens identifiés pour réaliser la séparation, y compris tous les mécanismes de protection, sont suffisants pour garantir une séparation claire et effective entre les fonctions qui sont dédiées à l'application de la TSP et celles qui ne le sont pas.

ADV_HLD.5.11C La conception de haut niveau doit justifier que les mécanismes de la TSF sont suffisants pour implémenter les fonctions de sécurité identifiées dans la conception de haut niveau.

Tâches de l'évaluateur :

ADV_HLD.5.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_HLD.5.2E L'évaluateur doit déterminer que la conception de haut niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.

10.3 Représentation de l'implémentation (ADV_IMP)

Objectifs

329 La description de la représentation de l'implémentation sous la forme de code source, de micro-programmes, de schémas descriptifs des matériels, etc. fournit les détails de fonctionnement interne de la TSF afin de constituer une aide à l'analyse.

Classement des composants

330 Les composants de cette famille sont classés sur la base de la complétude et de la structure de la représentation de l'implémentation fournie.

Notes d'application

331 La représentation de l'implémentation est utilisée pour exprimer la notion de représentation de la TSF de plus bas niveau, soit celle qui est utilisée pour créer la TSF elle-même, sans raffinement de conception supplémentaire. Le code source, qui est compilé par la suite, ou les schémas descriptifs des matériels qui sont utilisés pour construire la partie matérielle de la TOE, sont des exemples de parties de la représentation de l'implémentation.

332 Il est possible que l'évaluateur utilise la représentation de l'implémentation pour aider directement à la réalisation d'autres tâches d'évaluation (e.g. analyse de vulnérabilités, analyse de couverture des tests ou identification de tests complémentaires réalisés par l'évaluateur). On attend des auteurs de PP et de ST qu'ils sélectionnent un composant exigeant que l'implémentation soit suffisamment complète et étendue pour couvrir les besoins de toutes les autres exigences inclues dans le PP ou la ST.

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

Notes d'application

333 ADV_IMP.1.1D exige que le développeur fournisse la représentation de l'implémentation pour un sous-ensemble de la TSF. L'objectif recherché est que l'accès à au moins une partie de la TSF offre à l'évaluateur la possibilité d'examiner la représentation de l'implémentation des parties de la TOE pour lesquelles un tel examen peut améliorer significativement la compréhension des mécanismes employés, ainsi que l'assurance qui s'y attache. La fourniture d'un échantillon de la représentation de l'implémentation permettra également à l'évaluateur d'échantillonner les éléments de preuve de traçabilité pour acquérir l'assurance dans l'approche suivie pour le raffinement, et pour estimer la présentation de la représentation de l'implémentation elle-même.

334 L'élément ADV_IMP.1.2E définit l'exigence indiquant que l'évaluateur doit déterminer que la représentation de la TSF de plus bas niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE. Ceci fournit une correspondance directe entre les exigences fonctionnelles de sécurité de

la TOE et la représentation de la TSF de plus bas niveau, en complément des mises en correspondance deux à deux exigées par la famille ADV_RCR. On attend de l'évaluateur qu'il utilise les éléments de preuve fournis pour ADV_RCR comme données d'entrée pour sa détermination. Le plus bas niveau de représentation de la TSF pour ce composant est l'ensemble formé par la représentation de l'implémentation qui est fournie et par la partie de la conception de bas niveau pour laquelle aucune représentation de l'implémentation correspondante n'est fournie.

Dépendances :

ADV_LLD.1 Conception de bas niveau descriptive

ADV_RCR.1 Démonstration de correspondance informelle

ALC_TAT.1 Outils de développement bien définis

Tâches du développeur :

ADV_IMP.1.1D **Le développeur doit fournir la représentation de l'implémentation d'un sous-ensemble sélectionné de la TSF.**

Contenu et présentation des éléments de preuve :

ADV_IMP.1.1C **La représentation de l'implémentation doit définir la TSF d'une façon non ambiguë avec un niveau de détail suffisant pour qu'elle puisse être générée sans décision de conception supplémentaire.**

ADV_IMP.1.2C **La représentation de l'implémentation doit avoir une cohérence interne.**

Tâches de l'évaluateur :

ADV_IMP.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_IMP.1.2E **L'évaluateur doit déterminer que le plus bas niveau de représentation de la TSF fourni est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_IMP.2 Implémentation de la TSF

Notes d'application

335 L'élément ADV_IMP.2.2E définit une exigence indiquant que l'évaluateur doit déterminer que la représentation de l'implémentation est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE. Ceci fournit une correspondance directe entre les exigences fonctionnelles de sécurité de la TOE et la représentation de l'implémentation, en complément des mises en correspondance deux à deux exigées par la famille ADV_RCR. On attend de l'évaluateur qu'il utilise le éléments de preuve fournis pour ADV_RCR comme données d'entrée pour faire sa détermination.

Dépendances :

ADV_LLD.1 Conception de bas niveau descriptive

ADV_RCR.1 Démonstration de correspondance informelle

ALC_TAT.1 Outils de développement bien définis

Tâches du développeur :

ADV_IMP.2.1D Le développeur doit fournir une représentation de l'implémentation **de l'ensemble de la TSF.**

Contenu et présentation des éléments de preuve :

ADV_IMP.2.1C **La représentation de l'implémentation doit définir la TSF d'une façon non ambiguë avec un niveau de détail suffisant pour qu'elle puisse être générée sans décision de conception supplémentaire.**

ADV_IMP.2.2C La représentation de l'implémentation doit avoir une cohérence interne.

ADV_IMP.2.3C **La représentation de l'implémentation doit décrire les relations entre toutes les parties de l'implémentation.**

Tâches de l'évaluateur :

ADV_IMP.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_IMP.2.2E L'évaluateur doit déterminer que la **représentation de l'implémentation** est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.

ADV_IMP.3 Implémentation structurée de la TSF

Notes d'application

336 L'élément ADV_IMP.3.2E définit une exigence indiquant que l'évaluateur doit déterminer que la représentation de l'implémentation est une instantiation correcte et complète des exigences fonctionnellesde sécurité de la TOE. Ceci fournit une correspondance directe entre les exigences fonctionnelles de sécurité de la TOE et la représentation de l'implémentation, en complément des mises en correspondance deux à deux exigées par la famille ADV_RCR. On attend de l'évaluateur qu'il utilise les éléments de preuve fournis pour ADV_RCR comme données d'entrée pour sa détermination.

Dépendances :

ADV_INT.1 Modularité

ADV_LLD.1 Conception de bas niveau descriptive

ADV_RCR.1 Démonstration de correspondance informelle

ALC_TAT.1 Outils de développement bien définis

Tâches du développeur :

ADV_IMP.3.1D **Le développeur doit fournir une représentation de l'implémentation de l'ensemble de la TSF.**

Contenu et présentation des éléments de preuve :

ADV_IMP.3.1C La représentation de l'implémentation doit définir la TSF d'une façon non ambiguë avec un niveau de détail suffisant pour qu'elle puisse être générée sans décision de conception supplémentaire.

ADV_IMP.3.2C La représentation de l'implémentation doit avoir une cohérence interne.

ADV_IMP.3.3C La représentation de l'implémentation doit décrire les relations entre toutes les parties de l'implémentation.

ADV_IMP.3.4C **La représentation de l'implémentation doit être structurée en parties petites et compréhensibles.**

Tâches de l'évaluateur :

ADV_IMP.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_IMP.3.2E **L'évaluateur doit déterminer que la représentation de l'implémentation est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

10.4 Parties internes de la TSF (ADV_INT)

Objectifs

337 La présente famille concerne la structure interne de la TSF. Des exigences sont présentées sur la modularité, la décomposition en couches (pour séparer les niveaux d'abstraction et minimiser les dépendances circulaires), la minimisation de la complexité des mécanismes de mise en œuvre de la politique, ainsi que la minimisation du nombre de fonctionnalités au sein de la TSF qui ne sont pas dédiées à l'application de la TSP, conduisant ainsi à une TSF suffisamment simple pour qu'elle puisse être analysée.

338 Une conception modulaire réduit les dépendances mutuelles entre les éléments de la TSF, diminuant ainsi le risque qu'un changement ou une anomalie dans un module ait des effets sur l'ensemble de la TOE. Ainsi, une conception modulaire fournit la base pour pouvoir déterminer la portée des interactions avec d'autres éléments de la TSF, et pour obtenir une assurance accrue que des effets inattendus ne se produiront pas ; elle fournit également la base pour pouvoir élaborer et évaluer des séries de tests.

339 L'utilisation d'une structure en couches et d'une conception plus simple pour les fonctionnalités de la TSF qui sont dédiées à la mise en œuvre de la TSP, réduit la complexité de la TSF. Cela permet à son tour une meilleure compréhension de la TSF, apportant une assurance plus élevée sur le fait que les exigences fonctionnelles de sécurité de la TOE sont correctement et complètement instanciées dans l'implémentation.

340 La minimisation du nombre de fonctionnalités de la TSF qui ne sont pas dédiées à la mise en œuvre de la TSP réduit la possibilité d'anomalies dans la TSF. Combiné avec la modularité et la décomposition en couches, elle permet à l'évaluateur de se focaliser uniquement sur les fonctionnalités qui sont nécessaires à la mise en œuvre de la TSP.

341 La minimisation de la complexité de la conception contribue à l'assurance dans le fait que le code est compris : moins le code de la TSF est complexe, plus grande est la probabilité pour que la conception de la TSF soit compréhensible. La minimisation de la complexité de la conception est une caractéristique clé d'un mécanisme de validation de référence.

Classement des composants

342 Les composants de cette famille sont classés suivant le degré de structuration et de minimisation qui sont exigés.

Notes d'application

343 Le terme "parties de la TSF" est utilisé pour représenter des parties de la TSF à différents niveaux de granularité basés sur les représentations de la TSF disponibles. Les spécifications fonctionnelles permettent une identification en

termes d'interfaces, la conception de haut niveau permet une identification en termes de sous-systèmes, la conception de bas niveau permet une identification en termes de modules et la représentation de l'implémentation permet une identification en termes d'unités d'implémentation.

344 Les éléments ADV_INT.2.5C et ADV_INT.3.5C concernent la minimisation des interactions mutuelles entre les couches. Cependant, il est toujours possible d'avoir des interactions mutuelles entre les couches, mais dans une telle éventualité, le développeur devra démontrer que ces interactions mutuelles sont nécessaires et ne peuvent raisonnablement pas être évitées.

345 ADV_INT.2.6C introduit le concept de moniteur de référence, en exigeant la minimisation de la complexité des parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information identifiées dans la TSP. ADV_INT.3.6C développe ensuite le concept de moniteur de référence en exigeant la minimisation de la complexité de la totalité de la TSF.

346 Plusieurs des éléments faisant partie des composants de cette famille font référence à la description de l'architecture. La description de l'architecture est faite à un niveau d'abstraction similaire à celui de la conception de bas niveau, en cela qu'il est relatif aux modules de la TSF. Alors que la conception de bas niveau décrit la conception des modules de la TSF, la finalité de la description de l'architecture est de fournir les éléments de preuve de la modularité, de la décomposition en couches et de la minimisation de la complexité de la TSF, lorsque cela est applicable. La conception de bas niveau et la représentation de l'implémentation doivent être conformes à la description de l'architecture, afin de fournir l'assurance que ces représentations de la TSF offrent la modularité, la décomposition en couches et la minimisation de la complexité requises.

ADV_INT.1 Modularité

Dépendances :

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

Tâches du développeur :

ADV_INT.1.1D **Le développeur doit concevoir et structurer la TSF d'une façon modulaire, qui évite les interactions inutiles entre les modules de la conception.**

ADV_INT.1.2D **Le développeur doit fournir une description de l'architecture.**

Contenu et présentation des éléments de preuve :

ADV_INT.1.1C **La description de l'architecture doit identifier les modules de la TSF.**

ADV_INT.1.2C **La description de l'architecture doit décrire le but, les interfaces, les paramètres et les effets de chaque module de la TSF.**

ADV_INT.1.3C **La description de l'architecture doit décrire comment la conception de la TSF permet des modules largement indépendants qui évitent les interactions inutiles.**

Tâches de l'évaluateur :

ADV_INT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_INT.1.2E **L'évaluateur doit déterminer que la conception de bas niveau et la représentation de l'implémentation sont conformes à la description de l'architecture.**

ADV_INT.2 Réduction de la complexité

Notes d'application

347 Ce composant introduit le concept de moniteur de référence en exigeant la minimisation de la complexité des parties de la TSF qui mettent en oeuvre les politiques de contrôle d'accès et de contrôle de flux d'information identifiées dans la TSP.

Dépendances :

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

Tâches du développeur :

ADV_INT.2.1D Le développeur doit concevoir et structurer la TSF d'une façon modulaire, qui évite les interactions inutiles entre les modules de la conception.

ADV_INT.2.2D Le développeur doit fournir une description de l'architecture.

ADV_INT.2.3D **Le développeur doit concevoir et structurer la TSF en la décomposant en couches, ce qui permet de minimiser les interactions mutuelles entre les couches.**

ADV_INT.2.4D **Le développeur doit concevoir et structurer la TSF de façon à minimiser la complexité des parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information.**

Contenu et présentation des éléments de preuve :

ADV_INT.2.1C La description de l'architecture doit identifier les modules de la TSF **et doit spécifier les parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information**.

ADV_INT.2.2C La description de l'architecture doit décrire le but, les interfaces, les paramètres et les effets de chaque module de la TSF.

ADV_INT.2.3C La description de l'architecture doit décrire comment la conception de la TSF permet des modules largement indépendants qui évitent les interactions inutiles.

ADV_INT.2.4C **La description de l'architecture doit décrire la décomposition de l'architecture en couches.**

ADV_INT.2.5C **La description de l'architecture doit montrer que les interactions mutuelles ont été minimisées, et justifier celles qui subsistent.**

ADV_INT.2.6C **La description de l'architecture doit décrire comment les parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information ont été structurées pour minimiser la complexité.**

Tâches de l'évaluateur :

ADV_INT.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_INT.2.2E L'évaluateur doit déterminer que la conception de bas niveau et la représentation de l'implémentation sont conformes à la description de l'architecture.

ADV_INT.3 Minimisation de la complexité

Notes d'application

348 Ce composant exige que la propriété du moniteur de référence qui consiste à être "suffisamment simple pour être analysé" soit pleinement respectée. Quand ce composant est combiné avec les exigences fonctionnelles FPT_RVM.1 et FPT_SEP.3, le concept de moniteur de référence est pleinement mis en œuvre.

Dépendances :

ADV_IMP.2 Implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

Tâches du développeur :

ADV_INT.3.1D Le développeur doit concevoir et structurer la TSF d'une façon modulaire, qui évite les interactions inutiles entre les modules de la conception.

ADV_INT.3.2D Le développeur doit fournir une description de l'architecture.

ADV_INT.3.3D Le développeur doit concevoir et structurer la TSF en la décomposant en couches, ce qui permet de minimiser les interactions mutuelles entre les couches.

ADV_INT.3.4D Le développeur doit concevoir et structurer la TSF de façon à minimiser la complexité **de l'ensemble de la TSF**.

ADV_INT.3.5D **Le développeur doit concevoir et structurer les parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information afin qu'elles soient suffisamment simples pour être analysées.**

ADV_INT.3.6D **Le développeur doit garantir que les fonctions dont les objectifs ne sont pas pertinents pour la TSF sont exclues des modules de la TSF.**

Contenu et présentation des éléments de preuve :

ADV_INT.3.1C **La description de l'architecture doit identifier les modules de la TSF et doit spécifier les parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information.**

ADV_INT.3.2C La description de l'architecture doit décrire le but, les interfaces, les paramètres et les effets de chaque module de la TSF.

ADV_INT.3.3C La description de l'architecture doit décrire comment la conception de la TSF permet des modules largement indépendants qui évitent les interactions inutiles.

ADV_INT.3.4C **La description de l'architecture doit décrire la décomposition de l'architecture en couches.**

ADV_INT.3.5C La description de l'architecture doit montrer que les interactions mutuelles ont été minimisées, et justifier celles qui subsistent.

ADV_INT.3.6C La description de l'architecture doit décrire comment **l'ensemble de la TSF a été structuré** pour minimiser la complexité.

ADV_INT.3.7C **La description de l'architecture doit justifier l'inclusion dans la TSF de tout module non dédié à la mise en œuvre de la TSP.**

Tâches de l'évaluateur :

ADV_INT.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_INT.3.2E L'évaluateur doit déterminer que la conception de bas niveau et la représentation de l'implémentation sont conformes à la description de l'architecture.

ADV_INT.3.3E **L'évaluateur doit confirmer que les parties de la TSF qui mettent en œuvre les politiques de contrôle d'accès ou de contrôle de flux d'information sont suffisamment simples pour êtres analysées.**

10.5 Conception de bas niveau (ADV_LLD)

Objectifs

349 La conception de bas niveau d'une TOE fournit une description du fonctionnement interne de la TSF sous forme de modules, ainsi que de leurs relations et de leurs dépendances mutuelles. La conception de bas niveau procure l'assurance que les sous-systèmes de la TSF ont été raffinés de façon correcte et efficace.

350 Pour chaque module de la TSF, la conception de bas niveau décrit son but, sa fonction, ses interfaces, ses dépendances et l'implémentation de toutes les fonctions dédiées à la mise en œuvre de la TSP.

Classement des composants

351 Les composants de cette famille sont classés sur la base du degré de formalisme exigé pour la conception de bas niveau et le niveau de détail exigé pour les spécifications de l'interface.

Notes d'application

352 L'expression "module dédié à la mise en œuvre de la TSP" se réfère à tout module sur lequel repose l'application correcte de la TSP.

353 L'expression "fonctionnalités de sécurité" est utilisée pour représenter l'ensemble des opérations qu'un module effectue en contribution aux fonctions de sécurité implémentées par la TOE. Cette distinction est faite parce que les modules ne se rapportent pas nécessairement à des fonctions de sécurité spécifiques. Alors qu'un module donné peut correspondre directement à une fonction de sécurité, ou même à plusieurs fonctions de sécurité, il est également possible que plusieurs modules doivent être combinés pour réaliser une unique fonction de sécurité.

354 Les éléments ADV_LLD.*.6C exigent que la conception de bas niveau décrive comment chaque fonction dédiée à la mise en œuvre de la TSP est fournie. Le but de cette exigence est que la conception de bas niveau décrive l'implémentation prévue de chaque module du point de vue de la conception.

355 Les éléments ADV_LLD.*.2E de cette famille définissent une exigence indiquant que l'évaluateur doit déterminer que la conception de bas niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE. Ceci fournit une correspondance directe entre les exigences fonctionnelles de sécurité de la TOE et la conception de bas niveau, en complément des mises en correspondance deux à deux exigées par la famille ADV_RCR. On attend de l'évaluateur qu'il utilise les éléments de preuve fournis dans ADV_RCR comme données d'entrée pour sa détermination, et l'exigence de complétude est censée être liée au niveau d'abstraction de la conception de bas niveau.

356 ADV_LLD.2.9C introduit l'exigence d'une présentation complète des interfaces des modules. Cela fournit les détails nécessaires au test approfondi de la TOE (en utilisant les composants de ATE_DPT) et à l'estimation des vulnérabilités.

357 En ce qui concerne le niveau de formalisme de la conception de bas niveau, les styles informel, semi-formel et formel sont considérés comme étant hiérarchiques par nature. Ainsi, ADV_LLD.1.1C peut aussi être satisfait par une conception de bas niveau semi-formelle ou formelle, et ADV_LLD.2.1C peut aussi être satisfait par une conception de bas niveau formelle.

ADV_LLD.1 Conception de bas niveau descriptive

Dépendances :

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

ADV_RCR.1 Démonstration de correspondance informelle

Tâches du développeur :

ADV_LLD.1.1D **Le développeur doit fournir la conception de bas niveau de la TSF.**

Contenu et présentation des éléments de preuve :

ADV_LLD.1.1C **La présentation de la conception de bas niveau doit être en style informel.**

ADV_LLD.1.2C **La conception de bas niveau doit avoir une cohérence interne.**

ADV_LLD.1.3C **La conception de bas niveau doit décrire la TSF en termes de modules.**

ADV_LLD.1.4C **La conception de bas niveau doit décrire le but de chaque module.**

ADV_LLD.1.5C **La conception de bas niveau doit définir les relations mutuelles entre les modules en termes de fonctionnalités de sécurité fournies et de dépendances vis-à-vis des autres modules.**

ADV_LLD.1.6C **La conception de bas niveau doit décrire comment chaque fonction dédiée à la mise en œuvre de la TSP est fournie.**

ADV_LLD.1.7C **La conception de bas niveau doit identifier toutes les interfaces des modules de la TSF.**

ADV_LLD.1.8C **La conception de bas niveau doit identifier les interfaces des modules de la TSF qui sont visibles de l'extérieur.**

ADV_LLD.1.9C **La conception de bas niveau doit décrire le but et le mode d'utilisation de toutes les interfaces des modules de la TSF, en fournissant, lorsque cela est approprié, les détails sur les effets, les exceptions et les messages d'erreur.**

ADV_LLD.1.10C **La conception de bas niveau doit décrire la séparation de la TOE en modules dédiés à la mise en œuvre de la TSP et en autres modules.**

Tâches de l'évaluateur :

ADV_LLD.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_LLD.1.2E **L'évaluateur doit déterminer que la conception de bas niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_LLD.2 Conception de bas niveau semi-formelle

Dépendances :

ADV_HLD.3 Conception de haut niveau semi-formelle

ADV_RCR.2 Démonstration de correspondance semi-formelle

Tâches du développeur :

ADV_LLD.2.1D Le développeur doit fournir la conception de bas niveau de la TSF.

Contenu et présentation des éléments de preuve :

ADV_LLD.2.1C La présentation de la conception de bas niveau doit être en style **semi-formel**.

ADV_LLD.2.2C La conception de bas niveau doit avoir une cohérence interne.

ADV_LLD.2.3C La conception de bas niveau doit décrire la TSF en termes de modules.

ADV_LLD.2.4C La conception de bas niveau doit décrire le but de chaque module.

ADV_LLD.2.5C La conception de bas niveau doit définir les relations mutuelles entre les modules en termes de fonctionnalités de sécurité fournies et de dépendances vis-à-vis des autres modules.

ADV_LLD.2.6C **La conception de bas niveau doit décrire comment chaque fonction dédiée à la mise en œuvre de la TSP est fournie.**

ADV_LLD.2.7C La conception de bas niveau doit identifier toutes les interfaces des modules de la TSF.

ADV_LLD.2.8C La conception de bas niveau doit identifier les interfaces des modules de la TSF qui sont visibles de l'extérieur.

ADV_LLD.2.9C La conception de bas niveau doit décrire le but et le mode d'utilisation de toutes les interfaces des modules de la TSF, en fournissant les détails **complets** sur **tous** les effets, les exceptions et les messages d'erreur.

ADV_LLD.2.10C **La conception de bas niveau doit décrire la séparation de la TOE en modules dédiés à la mise en œuvre de la TSP et en autres modules.**

Tâches de l'évaluateur :

ADV_LLD.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_LLD.2.2E **L'évaluateur doit déterminer que la conception de bas niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.**

ADV_LLD.3 Conception de bas niveau formelle

Dépendances :

ADV_HLD.5 Conception de haut niveau formelle

ADV_RCR.3 Démonstration de correspondance formelle

Tâches du développeur :

ADV_LLD.3.1D Le développeur doit fournir la conception de bas niveau de la TSF.

Contenu et présentation des éléments de preuve :

ADV_LLD.3.1C La présentation de la conception de bas niveau doit être en style **formel**.

ADV_LLD.3.2C La conception de bas niveau doit avoir une cohérence interne.

ADV_LLD.3.3C La conception de bas niveau doit décrire la TSF en termes de modules.

ADV_LLD.3.4C La conception de bas niveau doit décrire le but de chaque module.

ADV_LLD.3.5C La conception de bas niveau doit définir les relations mutuelles entre les modules en termes de fonctionnalités de sécurité fournies et de dépendances vis-à-vis des autres modules.

ADV_LLD.3.6C **La conception de bas niveau doit décrire comment chaque fonction dédiée à la mise en œuvre de la TSP est fournie.**

ADV_LLD.3.7C La conception de bas niveau doit identifier toutes les interfaces des modules de la TSF.

ADV_LLD.3.8C La conception de bas niveau doit identifier les interfaces des modules de la TSF qui sont visibles de l'extérieur.

ADV_LLD.3.9C **La conception de bas niveau doit décrire le but et le mode d'utilisation de toutes les interfaces des modules de la TSF, en fournissant les détails complets sur tous les effets, les exceptions et les messages d'erreur.**

ADV_LLD.3.10C **La conception de bas niveau doit décrire la séparation de la TOE en modules dédiés à la mise en œuvre de la TSP et en autres modules.**

Tâches de l'évaluateur :

ADV_LLD.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_LLD.3.2E L'évaluateur doit déterminer que la conception de bas niveau est une instantiation correcte et complète des exigences fonctionnelles de sécurité de la TOE.

10.6 Correspondance des représentations (ADV_RCR)

Objectifs

358 La correspondance entre les différentes représentations de la TSF (i.e. spécifications globales de la TOE, spécifications fonctionnelles, conception de haut niveau, conception de bas niveau, représentation de l'implémentation) concerne l'instantiation correcte et complète des exigences jusqu'au niveau le moins abstrait de représentation de la TSF fourni. Ce résultat est obtenu grâce au raffinement par étapes et par les résultats cumulés des déterminations de correspondance entre les représentations de niveaux d'abstraction adjacents.

Classement des composants

359 Les composants de cette famille sont classés sur la base du niveau de formalisme exigé pour la correspondance entre les différentes représentations de la TSF.

Notes d'application

360 Le développeur doit démontrer à l'évaluateur que la représentation de la TSF la plus détaillée, ou la moins abstraite, qui est fournie est une instantiation correcte, cohérente et complète des fonctions exprimées par les exigences fonctionnelles dans la ST. Ceci est réalisé en montrant la correspondance entre les représentations adjacentes à un niveau de rigueur adéquat.

361 Cette famille d'exigences n'est pas destinée à couvrir les correspondances relatives au modèle de TSP ou à la TSP. Il s'agit plutôt, comme le montre la figure 10.2, de couvrir la correspondance entre les différentes représentations de la TSF qui sont fournies (i.e. spécifications globales de la TOE, spécifications fonctionnelles, conception de haut niveau, conception de bas niveau, représentation de l'implémentation).

362 Les éléments ADV_RCR.*.1C se réfèrent à "toutes les fonctionnalités de sécurité pertinentes" en définissant l'étendue de ce qui doit être raffiné entre chaque paire de représentations de la TSF adjacentes. Pour les raffinements entre les spécifications globales de la TOE et les spécifications fonctionnelles, cet élément exige uniquement que les fonctions de sécurité de la TOE figurant dans les spécifications globales de la TOE soient raffinées jusqu'aux spécifications fonctionnelles et n'exige pas que les spécifications fonctionnelles contiennent des détails relatifs aux mesures d'assurance (qui sont présentées dans les spécifications globales de la TOE). Lorsque la représentation de l'implémentation n'est fournie que pour un sous-ensemble de la TSF (comme pour ADV_IMP.1), les raffinements exigés entre la conception de bas niveau et la représentation de l'implémentation sont limités aux fonctionnalités de sécurité qui sont présentées dans la représentation de l'implémentation. Dans tous les autres cas, cet élément exige que toutes les parties de la représentation la plus abstraite de la TSF soient raffinées dans la représentation la moins abstraite de la TSF.

363 En ce qui concerne le niveau de formalisme pour la correspondance entre des représentations adjacentes de la TSF, les styles informel, semi-formel et formel sont considérés comme hiérarchiques par nature. Ainsi, ADV_RCR.2.2C et ADV_RCR.3.2C peuvent être satisfaits avec une preuve formelle de correspondance et, en l'absence de toute exigence relative à son niveau de formalisme, une démonstration de correspondance peut indifféremment être faite dans un style informel, semi-formel ou formel.

ADV_RCR.1 Démonstration de correspondance informelle

Dépendances :

Pas de dépendances.

Tâches du développeur :

ADV_RCR.1.1D **Le développeur doit fournir une analyse de correspondance entre toutes les paires de représentations de la TSF adjacentes fournies.**

Contenu et présentation des éléments de preuve :

ADV_RCR.1.1C **Pour chaque paire de représentations de la TSF adjacentes fournies, l'analyse doit démontrer que toutes les fonctionnalités de sécurité pertinentes de la représentation de la TSF la plus abstraite sont correctement et complètement raffinées dans la représentation de la TSF la moins abstraite.**

Tâches de l'évaluateur :

ADV_RCR.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_RCR.2 Démonstration de correspondance semi-formelle

Dépendances :

Pas de dépendances.

Tâches du développeur :

ADV_RCR.2.1D Le développeur doit fournir une analyse de correspondance entre toutes les paires de représentations de la TSF adjacentes fournies.

Contenu et présentation des éléments de preuve :

ADV_RCR.2.1C Pour chaque paire de représentations de la TSF adjacentes fournies, l'analyse doit démontrer que toutes les fonctionnalités de sécurité pertinentes de la représentation de la TSF la plus abstraite sont correctement et complètement raffinées dans la représentation de la TSF la moins abstraite.

ADV_RCR.2.2C **Pour chaque paire de représentations de la TSF adjacentes fournies, lorsque des parties des deux représentations sont spécifiées en style semi-formel au**

minimum, la démonstration de correspondance entre ces parties des représentations doit être faite en style semi-formel.

Tâches de l'évaluateur :

ADV_RCR.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_RCR.3 Démonstration de correspondance formelle

Notes d'application

364 Le développeur doit soit démontrer soit prouver la correspondance, comme cela est décrit dans les exigences ci-après, d'une façon adaptée au niveau de rigueur du style de présentation. Par exemple, la correspondance doit être prouvée lorsque les représentations correspondantes sont spécifiées en style formel.

Dépendances :

Pas de dépendances.

Tâches du développeur :

ADV_RCR.3.1D Le développeur doit fournir une analyse de correspondance entre toutes les paires de représentations de la TSF adjacentes fournies.

ADV_RCR.3.2D **Pour les parties des représentations qui sont spécifiées en style formel, le développeur doit apporter la preuve de cette correspondance.**

Contenu et présentation des éléments de preuve :

ADV_RCR.3.1C Pour chaque paire de représentations de la TSF adjacentes fournies, l'analyse doit **prouver ou** démontrer que toutes les fonctionnalités de sécurité pertinentes de la représentation de la TSF la plus abstraite sont correctement et complètement raffinées dans la représentation de la TSF la moins abstraite.

ADV_RCR.3.2C Pour chaque paire de représentations de la TSF adjacentes fournies, lorsque des parties **d'une** représentation **sont spécifiées en style semi-formel et** des parties **de l'autre** en style semi-formel au minimum, la démonstration de correspondance entre ces parties des représentations doit être faite en style semi-formel.

ADV_RCR.3.3C **Pour chaque paire de représentations de la TSF adjacentes fournies, lorsque des parties des deux représentations sont spécifiées formellement, la preuve de la correspondance entre ces parties des représentations doit être formelle.**

Tâches de l'évaluateur :

ADV_RCR.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_RCR.3.2E **L’évaluateur doit déterminer l’exactitude des preuves de la correspondance en contrôlant de façon sélective l’analyse formelle.**

10.7 Modélisation de la politique de sécurité (ADV_SPM)

Objectifs

365 L’objectif de cette famille est d’apporter une assurance complémentaire que les fonctions de sécurité figurant dans les spécifications fonctionnelles mettent en œuvre les politiques de la TSP. Ceci est réalisé via le développement d’un modèle de politique de sécurité qui est basé sur un sous-ensemble des politiques de la TSP, et en établissant une correspondance entre les spécifications fonctionnelles, le modèle de politique de sécurité et ces politiques de la TSP.

Classement des composants

366 Les composants de cette famille sont classés sur la base du degré de formalisme exigé pour le modèle de TSP, et le degré de formalisme exigé pour la correspondance entre le modèle de TSP et les spécifications fonctionnelles.

Notes d’application

367 Alors qu’une TSP peut inclure n’importe quelles politiques, les modèles de TSP ne représentent traditionnellement que des sous-ensembles de ces politiques, car la modélisation de certaines politiques dépasse pour l’instant l’état de l’art. L’état de l’art courant détermine les politiques qui peuvent être modélisées, et l’auteur du PP ou de la ST devrait identifier les fonctions spécifiques et les politiques correspondantes qui peuvent, et donc qui doivent, être modélisées. Au minimum, les politiques de contrôle d’accès et de contrôle des flux d’information doivent faire l’objet d’une modélisation (si elles font partie de la TSP) puisque l’état de l’art le permet.

368 Pour chacun des composants de cette famille, il est exigé de décrire dans le modèle de TSP, les règles et les caractéristiques des politiques de la TSP applicables, et de garantir que le modèle de TSP satisfait aux politiques de la TSP correspondantes. Les “règles” et les “caractéristiques” d’un modèle de TSP sont destinées à donner de la flexibilité sur le type de modèle qui peut être développé (e.g. modèle de transition d’états, de non-interférence). Par exemple, des règles peuvent être présentées sous la forme de “propriétés” (e.g. propriété de sécurité simple) et les caractéristiques peuvent être présentées sous la forme de définitions telles que “état initial”, “état sûr”, “sujets” et “objets”.

369 En ce qui concerne le niveau de formalisme du modèle de TSP et de la correspondance entre le modèle de TSP et les spécifications fonctionnelles, les styles informel, semi-formel et formel sont considérés comme étant hiérarchiques par nature. Ainsi, ADV_SPM.1.1C peut aussi être satisfait par un modèle de TSP semi-formel ou formel, et ADV_SPM.2.1C peut aussi être satisfait par un modèle de TSP formel. De plus, ADV_SPM.2.5C et ADV_SPM.3.5C peuvent être satisfaits par une preuve de correspondance formelle. Enfin, en l’absence de toute exigence relative au niveau de formalisme, une démonstration de correspondance peut être informelle, semi-formelle ou formelle.

ADV_SPM.1 Modèle informel de politique de sécurité de la TOE

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

Tâches du développeur :

ADV_SPM.1.1D **Le développeur doit fournir un modèle de TSP.**

ADV_SPM.1.2D **Le développeur doit démontrer la correspondance entre les spécifications fonctionnelles et le modèle de TSP.**

Contenu et présentation des éléments de preuve :

ADV_SPM.1.1C **Le modèle de TSP doit être en style informel.**

ADV_SPM.1.2C **Le modèle de TSP doit décrire les règles et les caractéristiques de toutes les politiques de la TSP qui peuvent être modélisées.**

ADV_SPM.1.3C **Le modèle de TSP doit inclure un argumentaire qui démontre qu'il est cohérent et complet par rapport à toutes les politiques de la TSP qui peuvent être modélisées.**

ADV_SPM.1.4C **La démonstration de correspondance entre le modèle de TSP et les spécifications fonctionnelles doit montrer que les fonctions de sécurité figurant dans les spécifications fonctionnelles sont cohérentes et complètes par rapport au modèle de TSP.**

Tâches de l'évaluateur :

ADV_SPM.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ADV_SPM.2 Modèle semi-formel de politique de sécurité de la TOE

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

Tâches du développeur :

ADV_SPM.2.1D Le développeur doit fournir un modèle de TSP.

ADV_SPM.2.2D Le développeur doit démontrer la correspondance entre les spécifications fonctionnelles et le modèle de TSP.

Contenu et présentation des éléments de preuve :

ADV_SPM.2.1C Le modèle de TSP doit être en style **semi-formel**.

ADV_SPM.2.2C Le modèle de TSP doit décrire les règles et les caractéristiques de toutes les politiques de la TSP qui peuvent être modélisées.

ADV_SPM.2.3C Le modèle de TSP doit inclure un argumentaire qui démontre qu'il est cohérent et complet par rapport à toutes les politiques de la TSP qui peuvent être modélisées.

ADV_SPM.2.4C La démonstration de correspondance entre le modèle de TSP et les spécifications fonctionnelles doit montrer que les fonctions de sécurité figurant dans les spécifications fonctionnelles sont cohérentes et complètes par rapport au modèle de TSP.

ADV_SPM.2.5C **Lorsque les spécifications fonctionnelles sont rédigées au minimum en style semi-formel, la démonstration de la correspondance entre le modèle de TSP et les spécifications fonctionnelles doit être semi-formelle.**

Tâches de l'évaluateur :

ADV_SPM.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_SPM.3 Modèle formel de politique de sécurité de la TOE

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

Tâches du développeur :

ADV_SPM.3.1D Le développeur doit fournir un modèle de TSP.

ADV_SPM.3.2D Le développeur doit démontrer **ou prouver, selon le cas,** la correspondance entre les spécifications fonctionnelles et le modèle de TSP.

Contenu et présentation des éléments de preuve :

ADV_SPM.3.1C Le modèle de TSP doit être en style **formel**.

ADV_SPM.3.2C Le modèle de TSP doit décrire les règles et les caractéristiques de toutes les politiques de la TSP qui peuvent être modélisées.

ADV_SPM.3.3C Le modèle de TSP doit inclure un argumentaire qui démontre qu'il est cohérent et complet par rapport à toutes les politiques de la TSP qui peuvent être modélisées.

ADV_SPM.3.4C La démonstration de correspondance entre le modèle de TSP et les spécifications fonctionnelles doit montrer que les fonctions de sécurité figurant dans les spécifications fonctionnelles sont cohérentes et complètes par rapport au modèle de TSP.

ADV_SPM.3.5C Lorsque les spécifications fonctionnelles sont rédigées en style **semi-formel**, la démonstration de correspondance entre le modèle de TSP et les spécifications fonctionnelles doit être semi-formelle.

ADV_SPM.3.6C **Lorsque les spécifications fonctionnelles sont rédigées en style formel, la preuve de correspondance entre le modèle de TSP et les spécifications fonctionnelles doit être formelle.**

Tâches de l'évaluateur :

ADV_SPM.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

11 Classe AGD : Guides

370 La classe “Guides” fournit les exigences pour la documentation destinée à l’utilisateur et à l’administrateur. Pour l’administration et l’utilisation sûres de la TOE, il est nécessaire de décrire tous les aspects pertinents relatifs à l’exploitation sûre de la TOE.

371 La figure 11.1 présente les familles de cette classe et la hiérarchie des composants au sein des familles.

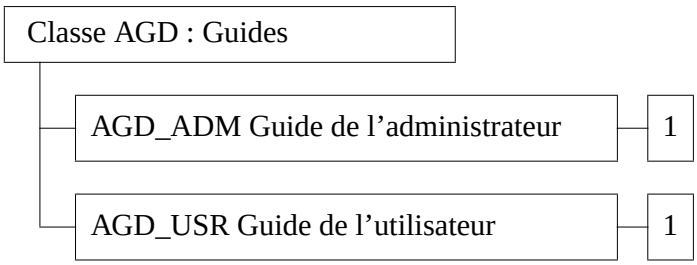

Figure 11.1 - Décomposition de la classe “Guides”

11.1 Guide de l'administrateur (AGD_ADM)

Objectifs

372 Le guide de l'administrateur est l'ensemble des documents prévus pour être utilisés par les personnes chargées d'effectuer de façon correcte la configuration, la maintenance et l'administration de la TOE, afin d'obtenir une sécurité maximum. L'exploitation sûre de la TOE dépendant du fonctionnement correct de la TSF, les personnes chargées d'exécuter les opérations précédentes sont considérées comme étant de confiance par la TSF. Le guide de l'administrateur est prévu pour aider les administrateurs à comprendre les fonctions de sécurité fournies par la TOE, incluant à la fois les fonctions qui obligent l'administrateur à effectuer des tâches critiques pour la sécurité et les fonctions qui fournissent des informations critiques pour la sécurité.

Classement des composants

373 La présente famille ne contient qu'un composant.

Notes d'application

374 Les exigences AGD_ADM.1.3C et AGD_ADM.1.7C prennent en compte le fait que tous les avertissements aux utilisateurs d'une TOE, relatifs à l'environnement de sécurité de la TOE et aux objectifs de sécurité décrits dans le PP ou la ST, sont couverts de manière adéquate dans le guide de l'administrateur.

375 Le concept de valeurs sûres, tel qu'il est employé dans AGD_ADM.1.5C, concerne la situation dans laquelle un administrateur contrôle les paramètres de sécurité. Des conseils relatifs à l'assignation sûre et non sûre de tels paramètres doivent être fournis. Ce concept est lié à l'utilisation du composant FMT_MSA.2 de la partie 2 des CC.

AGD_ADM.1 Guide de l'administrateur

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

Tâches du développeur :

AGD_ADM.1.1D **Le développeur doit fournir un guide de l'administrateur à l'attention du personnel chargé de l'administration du système.**

Contenu et présentation des éléments de preuve :

AGD_ADM.1.1C **Le guide de l'administrateur doit décrire les fonctions et les interfaces d'administration à la disposition de l'administrateur de la TOE.**

AGD_ADM.1.2C **Le guide de l'administrateur doit décrire comment administrer la TOE d'une façon sûre.**

AGD_ADM.1.3C **Le guide de l'administrateur doit contenir des avertissements concernant les fonctions et les privilèges qui devraient être contrôlés dans un environnement d'exploitation sûr.**

AGD_ADM.1.4C **Le guide de l'administrateur doit décrire toutes les hypothèses relatives au comportement de l'utilisateur, qui ont un rapport avec l'exploitation sûre de la TOE.**

AGD_ADM.1.5C **Le guide de l'administrateur doit décrire tous les paramètres de sécurité qui sont sous le contrôle de l'administrateur, en indiquant les valeurs sûres quand cela est approprié.**

AGD_ADM.1.6C **Le guide de l'administrateur doit décrire chaque type d'événement touchant à la sécurité, relatif aux fonctions d'administration qui doivent être réalisées, y compris le changement des caractéristiques de sécurité d'entités qui sont sous le contrôle de la TSF.**

AGD_ADM.1.7C **Le guide de l'administrateur doit être cohérent avec tous les autres documents fournis pour l'évaluation.**

AGD_ADM.1.8C **Le guide de l'administrateur doit décrire toutes les exigences de sécurité pour l'environnement TI, qui concernent l'administrateur.**

Tâches de l'évaluateur :

AGD_ADM.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

11.2 Guide de l'utilisateur (AGD_USR)

Objectifs

376 Le guide de l'utilisateur fait référence à des éléments prévus pour être utilisés par des utilisateurs humains de la TOE ne remplissant pas de fonctions d'administration et par d'autres utilisateurs (e.g. des programmeurs) qui utilisent les interfaces externes de la TOE. Le guide de l'utilisateur décrit les fonctions de sécurité fournies par la TSF et offre des instructions et des directives, comprenant des avertissements, pour l'utilisation sûre de la TSF.

377 Le guide de l'utilisateur donne une base pour former des hypothèses quant à l'utilisation de la TOE et une mesure de la confiance dans le fait que des utilisateurs, des fournisseurs d'applications et d'autres entités faisant appel aux interfaces externes de la TOE, comprendront ce qu'est l'exploitation sûre de la TOE et l'utiliseront comme cela est prévu.

Classement des composants

378 La présente famille ne contient qu'un composant.

Notes d'application

379 Les exigences AGD_USR.1.3.C et AGD_USR.1.5C prennent en compte le fait que tous les avertissements à l'attention des utilisateurs d'une TOE, par rapport à l'environnement de sécurité de la TOE et aux objectifs de sécurité décrits dans le PP ou la ST, sont couverts de manière adéquate dans le guide de l'utilisateur.

380 Dans de nombreux cas, il peut se révéler approprié que le guide soit présenté sous la forme de documents séparés : un document destiné aux utilisateurs humains et un autre pour les programmeurs d'applications ou les concepteurs de matériels qui utilisent les interfaces logicielles ou matérielles de la TOE.

AGD_USR.1 Guide de l'utilisateur

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

Tâches du développeur :

AGD_USR.1.1D **Le développeur doit fournir un guide de l'utilisateur.**

Contenu et présentation des éléments de preuve :

AGD_USR.1.1C **Le guide de l'utilisateur doit décrire les fonctions et les interfaces disponibles aux utilisateurs ne remplissant pas des fonctions d'administrateurs de la TOE.**

AGD_USR.1.2C **Le guide de l'utilisateur doit décrire l'utilisation des fonctions de sécurité fournies par la TOE accessibles aux utilisateurs.**

AGD_USR.1.3C **Le guide de l'utilisateur doit contenir des avertissements concernant les fonctions et les privilèges accessibles aux utilisateurs qui devraient être contrôlés dans un environnement d'exploitation sûr.**

AGD_USR.1.4C **Le guide de l'utilisateur doit présenter clairement toutes les responsabilités qui incombent à l'utilisateur et qui sont nécessaires pour une exploitation sûre de la TOE, y compris celles liées aux hypothèses relatives au comportement de l'utilisateur figurant dans l'énoncé de l'environnement de sécurité de la TOE.**

AGD_USR.1.5C **Le guide de l'utilisateur doit être cohérent avec toute autre documentation fournie pour l'évaluation.**

AGD_USR.1.6C **Le guide de l'utilisateur doit décrire toutes les exigences de sécurité pour l'environnement TI, qui concernent l'utilisateur.**

Tâches de l'évaluateur :

AGD_USR.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve**.

12 Classe ALC : Support au cycle de vie

381 La classe “Support au cycle de vie” constitue l’un des aspects permettant d’établir une discipline et un contrôle dans les processus de raffinement de la TOE pendant son développement et sa maintenance. La confiance dans la correspondance entre les exigences de sécurité de la TOE et la TOE elle-même est plus élevée si l’analyse de sécurité et la production des éléments de preuve sont réalisées de façon continue et font partie intégrante des activités de développement et de maintenance.

382 La figure 12.1 présente les familles de cette classe et la hiérarchie des composants au sein des familles.

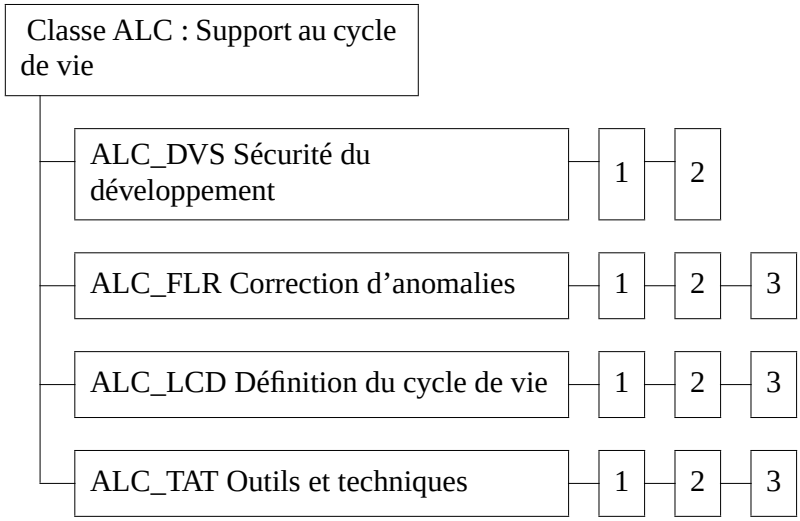

Figure 12.1 - Décomposition de la classe “Support au cycle de vie”

12.1 Sécurité du développement (ALC_DVS)

Objectifs

383 La sécurité du développement traite de mesures de sécurité physiques, organisationnelles, touchant au personnel et autres qui peuvent être utilisées dans l'environnement de développement pour protéger la TOE. Ceci comprend la sécurité physique du site de développement et toutes les procédures utilisées pour sélectionner l'équipe de développement.

Classement des composants

384 Les composants de cette famille sont classés suivant qu'il est exigé ou non de justifier le caractère suffisant des mesures de sécurité.

Notes d'application

385 Cette famille traite des mesures destinées à supprimer ou à réduire les menaces existant sur le site du développeur. Inversement, les menaces devant être contrées sur le site de l'utilisateur de la TOE sont normalement couvertes dans l'énoncé de l'environnement de sécurité d'un PP ou d'une ST.

386 L'évaluateur devrait déterminer s'il existe un besoin de visiter le site du développeur afin de confirmer que les exigences de la présente famille sont satisfaites.

387 Il est reconnu que la confidentialité peut ne pas toujours être un problème pour la protection de la TOE dans son environnement de développement. L'utilisation du mot "nécessaire" implique la sélection des mesures de protection appropriées.

ALC_DVS.1 Identification des mesures de sécurité

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_DVS.1.1D **Le développeur doit produire la documentation relative à la sécurité du développement.**

Contenu et présentation des éléments de preuve :

ALC_DVS.1.1C **La documentation relative à la sécurité du développement doit décrire toutes les mesures de sécurité physiques, organisationnelles, touchant au personnel et autres qui sont nécessaires pour protéger la confidentialité et l'intégrité de la conception et de l'implémentation de la TOE dans son environnement de développement.**

ALC_DVS.1.2C **La documentation relative à la sécurité du développement doit fournir des éléments de preuve indiquant que ces mesures de sécurité sont appliquées au cours du développement et de la maintenance de la TOE.**

Tâches de l'évaluateur :

ALC_DVS.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve**.

ALC_DVS.1.2E **L'évaluateur doit confirmer que les mesures de sécurité sont appliquées.**

ALC_DVS.2 Caractère suffisant des mesures de sécurité

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_DVS.2.1D Le développeur doit produire la documentation relative à la sécurité du développement.

Contenu et présentation des éléments de preuve :

ALC_DVS.2.1C La documentation relative à la sécurité du développement doit décrire toutes les mesures de sécurité physiques, organisationnelles, touchant au personnel et autres qui sont nécessaires pour protéger la confidentialité et l'intégrité de la conception et de l'implémentation de la TOE dans son environnement de développement.

ALC_DVS.2.2C La documentation relative à la sécurité du développement doit fournir des éléments de preuve indiquant que ces mesures de sécurité sont appliquées au cours du développement et de la maintenance de la TOE.

ALC_DVS.2.3C **Les éléments de preuve doivent justifier que les mesures de sécurité fournissent le niveau de protection nécessaire pour maintenir la confidentialité et l'intégrité de la TOE.**

Tâches de l'évaluateur :

ALC_DVS.2.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ALC_DVS.2.2E L'évaluateur doit confirmer que les mesures de sécurité sont appliquées.

12.2 Correction d'anomalies (ALC_FLR)

Objectifs

388 La correction d'anomalies exige que les anomalies de sécurité qui ont été découvertes soient suivies, ainsi que leurs états successifs, et qu'elles soient corrigées par le développeur. Bien que la conformité ultérieure avec les procédures de correction d'anomalies ne puisse pas être déterminée au moment de l'évaluation de la TOE, il est possible d'évaluer les politiques et les procédures qu'un développeur a mises en place pour suivre les anomalies, leurs états successifs et les corriger, et pour diffuser les informations et les corrections relatives aux anomalies.

Classement des composants

389 Les composants de la présente famille sont classés suivant l'étendue du champ d'application des procédures de correction d'anomalies et suivant la rigueur des politiques de correction d'anomalies.

Notes d'application

390 La présente famille procure l'assurance que la TOE sera maintenue et bénéficiera d'un support, exigeant du développeur de la TOE qu'il suive les anomalies, leurs états successifs et les corrige dans la TOE. De plus, des exigences sont incluses pour la diffusion des corrections d'anomalies. Cependant, cette famille n'impose pas d'exigences d'évaluation après l'évaluation en cours.

391 Les procédures de correction d'anomalies devraient décrire les méthodes pour traiter tous les types d'anomalies rencontrés. Certaines anomalies peuvent ne pas être corrigées immédiatement. Il peut y avoir des cas où une anomalie ne peut pas être corrigée, et où d'autres mesures (e.g. organisationnelles) doivent être prises. La documentation fournie devrait couvrir les procédures pour la transmission des corrections aux sites d'exploitation et pour l'émission d'informations sur les anomalies quand les corrections sont prévues avec un certain délai (et pour indiquer ce qu'il faut faire en attendant), ou quand il n'est pas possible de faire des corrections.

ALC_FLR.1 Correction d'anomalies élémentaire

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_FLR.1.1D **Le développeur doit documenter les procédures de correction d'anomalies.**

Contenu et présentation des éléments de preuve :

ALC_FLR.1.1C **La documentation relative aux procédures de correction d'anomalies doit décrire les procédures utilisées pour surveiller toutes les anomalies de sécurité signalées dans chaque version de la TOE.**

ALC_FLR.1.2C **Les procédures de correction d'anomalies doivent exiger qu'une description de la nature et des effets de chaque anomalie de sécurité soit fournie, ainsi que l'état d'avancement de la recherche d'une correction de cette anomalie.**

ALC_FLR.1.3C **Les procédures de correction d'anomalies doivent exiger que des actions correctives soient identifiées pour chacune des anomalies de sécurité.**

ALC_FLR.1.4C **La documentation relative aux procédures de correction d'anomalies doit décrire les méthodes utilisées pour fournir aux utilisateurs de la TOE les informations sur les anomalies, les corrections et les conseils concernant les actions correctives.**

Tâches de l'évaluateur :

ALC_FLR.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ALC_FLR.2 Procédures pour signaler les anomalies

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_FLR.2.1D Le développeur doit documenter les procédures de correction d'anomalies.

ALC_FLR.2.2D **Le développeur doit établir une procédure pour recevoir et traiter les anomalies de sécurité signalées par les utilisateurs et les demandes de corrections de ces anomalies.**

Contenu et présentation des éléments de preuve :

ALC_FLR.2.1C **La documentation relative aux procédures de correction d'anomalies doit décrire les procédures utilisées pour surveiller toutes les anomalies de sécurité signalées dans chaque version de la TOE.**

ALC_FLR.2.2C **Les procédures de correction d'anomalies doivent exiger qu'une description de la nature et des effets de chaque anomalie de sécurité soit fournie, ainsi que l'état d'avancement de la recherche d'une correction de cette anomalie.**

ALC_FLR.2.3C **Les procédures de correction d'anomalies doivent exiger que des actions correctives soient identifiées pour chacune des anomalies de sécurité.**

ALC_FLR.2.4C **La documentation relative aux procédures de correction d'anomalies doit décrire les méthodes utilisées pour fournir aux utilisateurs de la TOE les informations sur les anomalies, les corrections et les conseils concernant les actions correctives.**

ALC_FLR.2.5C **Les procédures pour traiter les anomalies de sécurité signalées doivent garantir que toutes les anomalies signalées seront corrigées et que les corrections seront diffusées aux utilisateurs de la TOE.**

ALC_FLR.2.6C **Les procédures pour traiter les anomalies de sécurité signalées doivent fournir des garanties pour que toutes les corrections apportées à ces anomalies de sécurité n'introduisent pas de nouvelles anomalies.**

Tâches de l'évaluateur :

ALC_FLR.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ALC_FLR.3 Correction d'anomalies systématique

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_FLR.3.1D Le développeur doit documenter les procédures de correction d'anomalies.

ALC_FLR.3.2D **Le développeur doit établir une procédure pour recevoir et traiter les anomalies de sécurité signalées par les utilisateurs et les demandes de corrections de ces anomalies.**

ALC_FLR.3.3D **Le développeur doit désigner un ou plusieurs points de contacts spécifiques pour les rapports et les requêtes des utilisateurs relatives aux problèmes de sécurité qui impliquent la TOE.**

Contenu et présentation des éléments de preuve :

ALC_FLR.3.1C La documentation relative aux procédures de correction d'anomalies doit décrire les procédures utilisées pour surveiller toutes les anomalies de sécurité signalées dans chaque version de la TOE.

ALC_FLR.3.2C Les procédures de correction d'anomalies doivent exiger qu'une description de la nature et des effets de chaque anomalie de sécurité soit fournie, ainsi que l'état d'avancement de la recherche d'une correction de cette anomalie.

ALC_FLR.3.3C Les procédures de correction d'anomalies doivent exiger que des actions correctives soient identifiées pour chacune des anomalies de sécurité.

ALC_FLR.3.4C La documentation relative aux procédures de correction d'anomalies doit décrire les méthodes utilisées pour fournir aux utilisateurs de la TOE les informations sur les anomalies, les corrections et les conseils concernant les actions correctives.

ALC_FLR.3.5C **Les procédures pour traiter les anomalies de sécurité signalées doivent garantir que toutes les anomalies signalées seront corrigées et que les corrections seront diffusées aux utilisateurs de la TOE.**

ALC_FLR.3.6C Les procédures pour traiter les anomalies de sécurité signalées doivent fournir des garanties pour que toutes les corrections apportées à ces anomalies de sécurité n'introduisent pas de nouvelles anomalies.

ALC_FLR.3.7C **Les procédures de correction d'anomalies doivent inclure une procédure exigeant des réactions en temps opportun pour distribuer de façon automatique les rapports d'anomalie de sécurité, et les corrections associées, aux utilisateurs enregistrés qui pourraient être affectés par l'anomalie de sécurité.**

Tâches de l'évaluateur :

ALC_FLR.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

12.3 Définition du cycle de vie (ALC_LCD)

Objectifs

392 Un développement et une maintenance de la TOE mal contrôlés peuvent aboutir à une implémentation erronée d'une TOE (ou bien à une TOE qui ne satisfait pas à toutes ses exigences de sécurité). Cela, à son tour, peut avoir pour résultat des violations de la sécurité. Par conséquent, il est important qu'un modèle pour le développement et la maintenance d'une TOE soit établi aussi tôt que possible dans le cycle de vie de la TOE.

393 L'utilisation d'un modèle pour le développement et la maintenance d'une TOE ne garantit pas que la TOE ne contiendra pas d'anomalies, ni que la TOE satisfera à toutes ses exigences de sécurité fonctionnelles. Il est possible que le modèle choisi soit insuffisant ou inadéquat et qu'en conséquence aucune amélioration de la qualité de la TOE ne soit visible. L'utilisation d'un modèle de cycle de vie qui a reçu l'approbation de certains groupes d'experts (e.g. des experts universitaires, des organismes de normalisation) augmente la probabilité que les modèles de développement et de maintenance contribueront à la qualité globale de la TOE.

Classement des composants

394 Les composants de la présente famille sont classés suivant les exigences croissantes relatives aux caractères normalisé et mesurable du modèle de cycle de vie, ainsi qu'à la conformité à ce modèle.

Notes d'application

395 Un modèle de cycle de vie comprend les procédures, les outils et les techniques utilisés pour développer et maintenir la TOE. Parmi les aspects du processus qui peuvent être couverts par un tel modèle, on peut citer les méthodes de conception, les procédures de revue, les contrôles de gestion de projet, les procédures pour contrôler les changements, les procédures relatives aux méthodes de test et à la réception. Un modèle de cycle de vie efficace traitera ces aspects du processus de développement et de maintenance dans le cadre d'une structure de gestion globale qui attribue les responsabilités et contrôle l'avancement.

396 Bien que la définition du cycle de vie traite de la maintenance de la TOE et donc des aspects qui ne deviennent pertinents qu'après l'achèvement de l'évaluation, l'évaluation du cycle de vie permet d'obtenir plus d'assurance au moyen de l'analyse des informations du cycle de vie pour la TOE fournie au moment de l'évaluation.

397 Un modèle de cycle de vie normalisé est un modèle qui a été approuvé par certains groupes d'experts (e.g. des experts universitaires, des organismes de normalisation).

398 Un modèle de cycle de vie mesurable est un modèle doté de paramètres arithmétiques ou de métriques qui mesurent les propriétés du développement de la TOE (e.g. des métriques de complexité du code source).

399 Un modèle de cycle de vie pourvoit aux contrôles nécessaires tout au long du développement et de la maintenance de la TOE, si le développeur peut fournir les informations montrant que le modèle minimise de façon adéquate le danger de violations de la sécurité dans la TOE. Les informations données dans la ST, relatives à l'environnement prévu de la TOE et aux objectifs de sécurité de la TOE, peuvent être utiles pour définir le modèle dans la partie du cycle de vie qui se situe après la livraison de la TOE.

ALC_LCD.1 Modèle de cycle de vie défini par le développeur

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_LCD.1.1D **Le développeur doit établir un modèle de cycle de vie utilisé dans le développement et la maintenance de la TOE.**

ALC_LCD.1.2D **Le développeur doit fournir une documentation relative à la définition du cycle de vie.**

Contenu et présentation des éléments de preuve :

ALC_LCD.1.1C **La documentation relative à la définition du cycle de vie doit décrire le modèle utilisé pour développer et maintenir la TOE.**

ALC_LCD.1.2C **Le modèle de cycle de vie doit permettre les contrôles nécessaires tout au long du développement et de la maintenance de la TOE.**

Tâches de l'évaluateur :

ALC_LCD.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve**.

ALC_LCD.2 Modèle de cycle de vie normalisé

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_LCD.2.1D Le développeur doit établir un modèle de cycle de vie utilisé dans le développement et la maintenance de la TOE.

ALC_LCD.2.2D Le développeur doit fournir une documentation relative à la définition du cycle de vie.

ALC_LCD.2.3D **Le développeur doit utiliser un modèle de cycle de vie normalisé pour développer et maintenir la TOE.**

Contenu et présentation des éléments de preuve :

ALC_LCD.2.1C La documentation relative à la définition du cycle de vie doit décrire le modèle utilisé pour développer et maintenir la TOE.

ALC_LCD.2.2C Le modèle de cycle de vie doit permettre les contrôles nécessaires tout au long du développement et de la maintenance de la TOE.

ALC_LCD.2.3C **La documentation relative à la définition du cycle de vie doit expliquer pourquoi le modèle a été choisi.**

ALC_LCD.2.4C **La documentation relative à la définition du cycle de vie doit expliquer comment le modèle est utilisé pour développer et maintenir la TOE.**

ALC_LCD.2.5C **La documentation relative à la définition du cycle de vie doit démontrer la conformité avec le modèle de cycle de vie normalisé.**

Tâches de l'évaluateur :

ALC_LCD.2.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ALC_LCD.3 Modèle de cycle de vie mesurable

Dépendances :

Pas de dépendances.

Tâches du développeur :

ALC_LCD.3.1D Le développeur doit établir un modèle de cycle de vie utilisé dans le développement et la maintenance de la TOE.

ALC_LCD.3.2D Le développeur doit fournir une documentation relative à la définition du cycle de vie.

ALC_LCD.3.3D Le développeur doit utiliser un modèle de cycle de vie normalisé **et mesurable** pour développer et maintenir la TOE.

ALC_LCD.3.4D **Le développeur doit mesurer le développement de la TOE en utilisant le modèle de cycle de vie normalisé et mesurable.**

Contenu et présentation des éléments de preuve :

ALC_LCD.3.1C La documentation relative à la définition du cycle de vie doit décrire le modèle utilisé pour développer et maintenir la TOE, **y compris les informations détaillées sur les paramètres arithmétiques ou les métriques utilisées pour mesurer le développement de la TOE par rapport au modèle**.

ALC_LCD.3.2C Le modèle de cycle de vie doit permettre les contrôles nécessaires tout au long du développement et de la maintenance de la TOE.

ALC_LCD.3.3C La documentation relative à la définition du cycle de vie doit expliquer pourquoi le modèle a été choisi.

ALC_LCD.3.4C La documentation relative à la définition du cycle de vie doit expliquer comment le modèle est utilisé pour développer et maintenir la TOE.

ALC_LCD.3.5C La documentation relative à la définition du cycle de vie doit démontrer la conformité avec le modèle de cycle de vie normalisé **et mesurable**.

ALC_LCD.3.6C **La documentation relative au cycle de vie doit fournir les résultats des mesures de développement de la TOE en utilisant le modèle normalisé et mesurable du cycle de vie.**

Tâches de l'évaluateur :

ALC_LCD.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

12.4 Outils et techniques (ALC_TAT)

Objectifs

400 La famille “Outils et techniques” constitue un aspect de la sélection d’outils qui sont utilisés pour développer, analyser et implémenter la TOE. Elle comprend des exigences pour empêcher que ne soient utilisés pour développer la TOE des outils de développement mal définis, incohérents ou incorrects. Cela inclut, entre autres, les langages de programmation, la documentation, les normes d’implémentation et d’autres parties de la TOE telles que les bibliothèques utiles à l’exécution.

Classement des composants

401 Les composants de la présente famille sont classés suivant les exigences croissantes concernant la description et le champ d’application des normes d’implémentation et la documentation des options dépendant de l’implémentation.

Notes d’application

402 Il existe une exigence pour utiliser des outils de développement bien définis. Ce sont des outils qui se sont révélés être applicables sans le recours à une clarification supplémentaire intensive. Par exemple, les langages de programmation et les systèmes d’aide à la conception automatisée (CAD : computer aided design) qui sont basés sur une norme publiée par des organismes de normalisation, sont considérés comme étant bien définis.

403 La famille “Outils et techniques” distingue les normes d’implémentation appliquées par le développeur (ALC_TAT.2.3D) des normes d’implémentation valables pour “toutes les parties de la TOE” (ALC_TAT.3.3D) incluant en plus des logiciels, matériels et micro-programmes de tierces parties.

404 L’exigence contenue dans ALC_TAT.1.2C s’applique particulièrement aux langages de programmation de façon à garantir que toutes les instructions du code source ont une signification non ambiguë.

ALC_TAT.1 Outils de développement bien définis

Dépendances :

ADV_IMP.1 Sous-ensemble de l’implémentation de la TSF

Tâches du développeur :

ALC_TAT.1.1D **Le développeur doit identifier les outils de développement utilisés pour la TOE.**

ALC_TAT.1.2D **Le développeur doit documenter les options dépendant de l’implémentation qui ont été choisies pour les outils de développement.**

Contenu et présentation des éléments de preuve :

ALC_TAT.1.1C **Tous les outils de développement utilisés pour l'implémentation doivent être bien définis.**

ALC_TAT.1.2C **La documentation relative aux outils de développement doit définir sans ambiguïté la signification de toutes les instructions utilisées dans l'implémentation.**

ALC_TAT.1.3C **La documentation relative aux outils de développement doit définir sans ambiguïté la signification de toutes les options dépendant de l'implémentation.**

Tâches de l'évaluateur :

ALC_TAT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve**.

ALC_TAT.2 Conformité aux normes d'implémentation

Dépendances :

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

Tâches du développeur :

ALC_TAT.2.1D Le développeur doit identifier les outils de développement utilisés pour la TOE.

ALC_TAT.2.2D Le développeur doit documenter les options dépendant de l'implémentation qui ont été choisies pour les outils de développement.

ALC_TAT.2.3D **Le développeur doit décrire les normes d'implémentation à appliquer.**

Contenu et présentation des éléments de preuve :

ALC_TAT.2.1C Tous les outils de développement utilisés pour l'implémentation doivent être bien définis.

ALC_TAT.2.2C La documentation relative aux outils de développement doit définir sans ambiguïté la signification de toutes les instructions utilisées dans l'implémentation.

ALC_TAT.2.3C La documentation relative aux outils de développement doit définir sans ambiguïté la signification de toutes les options dépendant de l'implémentation.

Tâches de l'évaluateur :

ALC_TAT.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ALC_TAT.2.2E **L'évaluateur doit confirmer que les normes d'implémentation ont bien été appliquées.**

ALC_TAT.3 Conformité aux normes d'implémentation - toutes parties de la TOE

Dépendances :

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

Tâches du développeur :

ALC_TAT.3.1D Le développeur doit identifier les outils de développement utilisés pour la TOE.

ALC_TAT.3.2D Le développeur doit documenter les options dépendant de l'implémentation qui ont été choisies pour les outils de développement.

ALC_TAT.3.3D Le développeur doit décrire les normes d'implémentation **pour toutes les parties de la TOE**.

Contenu et présentation des éléments de preuve :

ALC_TAT.3.1C Tous les outils de développement utilisés pour l'implémentation doivent être bien définis.

ALC_TAT.3.2C La documentation relative aux outils de développement doit définir sans ambiguïté la signification de toutes les instructions utilisées dans l'implémentation.

ALC_TAT.3.3C La documentation relative aux outils de développement doit définir sans ambiguïté la signification de toutes les options dépendant de l'implémentation.

Tâches de l'évaluateur :

ALC_TAT.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ALC_TAT.3.2E L'évaluateur doit confirmer que les normes d'implémentation ont bien été appliquées.

13 Classe ATE : Tests

405 La classe "Tests" comprend quatre familles : couverture (ATE_COV), profondeur (ATE_DPT), tests indépendants (e.g. les tests fonctionnels exécutés par les évaluateurs) (ATE_IND) et tests fonctionnels (ATE_FUN). Les tests aident à établir que les exigences fonctionnelles pour la sécurité de la TOE sont satisfaites. Les tests fournissent l'assurance que la TOE satisfait au moins aux exigences fonctionnelles de sécurité de la TOE, bien qu'ils ne puissent pas établir le fait que la TOE ne fait rien de plus que ce qui a été spécifié. Les tests peuvent aussi s'appliquer aux structures internes de la TSF, comme les tests des sous-systèmes et des modules par rapport à leurs spécifications.

406 Les aspects de couverture et de profondeur ont été séparés des tests fonctionnels pour obtenir une flexibilité accrue dans l'application des composants des familles concernées. Cependant, les exigences de ces trois familles sont destinées à être appliquées ensemble.

407 La famille "Tests indépendants" a des dépendances vers les autres familles afin de fournir les informations nécessaires pour appuyer ces exigences, mais elle traite principalement des actions de l'évaluateur effectuées de manière indépendante.

408 L'accent dans cette classe porte sur la confirmation du fait que la TSF fonctionne conformément à ses spécifications. Cela inclut à la fois des tests à caractère positif basés sur les exigences fonctionnelles et des tests à caractère négatif pour vérifier que tout comportement non désiré est absent. La présente classe ne traite pas des tests de pénétration, qui visent à trouver les vulnérabilités permettant à un utilisateur de violer la politique de sécurité. Les tests de pénétration sont basés sur une analyse de la TOE qui cherche à identifier de manière spécifique les vulnérabilités introduites dans la conception et l'implémentation de la TSF et sont traités séparément comme un aspect de l'estimation des vulnérabilités dans la classe AVA.

409 La figure 13.1 présente les familles de cette classe et la hiérarchie des composants au sein des familles.

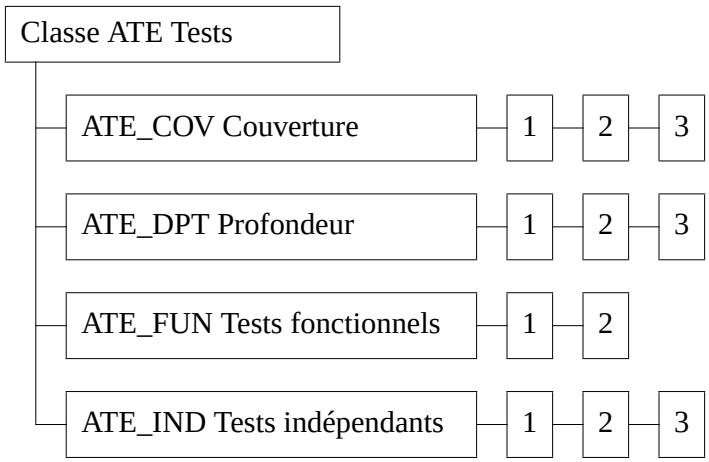

Figure 13.1 - Décomposition de la classe “Tests”

13.1 Couverture (ATE_COV)

Objectifs

410 La présente famille concerne les aspects du test relatifs à la complétude de la couverture des tests, c'est-à-dire qu'elle traite de l'étendue avec laquelle est testée la TSF et du fait de savoir si les tests sont suffisamment étendus pour démontrer que la TSF fonctionne conformément à ses spécifications.

Classement des composants

411 Les composants de cette famille sont classés suivant la rigueur croissante du test des interfaces et la rigueur croissante de l'analyse du caractère suffisant des tests afin de démontrer que la TSF fonctionne conformément à ses spécifications fonctionnelles.

ATE_COV.1 Éléments de preuve de la couverture

Objectifs

412 Dans ce composant, l'objectif est d'établir que la TSF a été testée par rapport à ses spécifications fonctionnelles. Cela doit être accompli au moyen d'un examen des éléments de preuve de correspondance du développeur.

Notes d'application

413 Alors que l'objectif des tests est de couvrir la TSF, la seule exigence pour contrôler cette affirmation est de fournir une correspondance informelle des tests avec les spécifications fonctionnelles et les données de test elles-mêmes.

414 Dans ce composant, le développeur doit montrer comment les tests qui ont été identifiés correspondent à la TSF telle qu'elle est décrite dans les spécifications fonctionnelles. Cela peut être obtenu par un énoncé de correspondance, éventuellement en utilisant un tableau. Ces informations doivent aider l'évaluateur à planifier le programme de tests pour l'évaluation. À ce niveau, il n'y a pas d'exigence pour une couverture complète de chaque aspect de la TSF par le développeur, et l'évaluateur devra prendre en compte toute déficience dans ce domaine.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_COV.1.1D **Le développeur doit fournir les éléments de preuve de la couverture des tests.**

Contenu et présentation des éléments de preuve :

ATE_COV.1.1C **Les éléments de preuve de la couverture des tests doivent montrer la correspondance entre les tests identifiés dans la documentation de test et la TSF, telle qu'elle est décrite dans les spécifications fonctionnelles.**

Tâches de l'évaluateur :

ATE_COV.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ATE_COV.2 Analyse de la couverture

Objectifs

415 Dans ce composant, l'objectif est d'établir que la TSF a été testée par rapport à ses spécifications fonctionnelles d'une manière systématique. Cela doit être accompli au moyen d'un examen de l'analyse de correspondance du développeur.

Notes d'application

416 Le développeur doit démontrer que les tests qui ont été identifiés incluent les tests de toutes les fonctions de sécurité, telle qu'elles sont décrites dans les spécifications fonctionnelles. L'analyse ne devrait pas seulement montrer la correspondance entre les tests et les fonctions de sécurité, mais devrait également fournir suffisamment d'informations pour que l'évaluateur détermine comment les fonctions ont été utilisées. Ces informations peuvent être utilisées pour planifier des tests supplémentaires par les évaluateurs. Bien qu'à ce niveau, le développeur doive démontrer que chacune des fonctions figurant dans les spécifications fonctionnelles a été testée, les tests effectués sur chaque fonction n'ont pas besoin d'être exhaustifs.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_COV.2.1D Le développeur doit fournir **une analyse** de la couverture des tests.

Contenu et présentation des éléments de preuve :

ATE_COV.2.1C L'**analyse** de la couverture des tests doit **démontrer** la correspondance entre les tests identifiés dans la documentation de test et la TSF, telle qu'elle est décrite dans les spécifications fonctionnelles.

ATE_COV.2.2C **L'analyse de la couverture des tests doit démontrer que la correspondance entre la TSF, telle qu'elle est décrite dans les spécifications fonctionnelles, et les tests identifiés dans la documentation de test est complète.**

Tâches de l'évaluateur :

ATE_COV.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ATE_COV.3 Analyse rigoureuse de la couverture

Objectifs

417 Dans ce composant, l'objectif est d'établir que la TSF a été testée par rapport à ses spécifications fonctionnelles d'une manière systématique et exhaustive. Cela doit être accompli au moyen d'un examen de l'analyse de correspondance du développeur.

Notes d'application

418 Le développeur doit fournir un argument convaincant montrant que les tests qui ont été identifiés couvrent toutes les fonctions de sécurité et que les tests de chaque fonction de sécurité sont complets. Il restera peu d'opportunité à l'évaluateur pour imaginer des tests fonctionnels supplémentaires pour les interfaces de la TSF, basés sur les spécifications fonctionnelles, car elles auront dû être testées de façon exhaustive. Néanmoins, l'évaluateur devrait s'efforcer d'imaginer de tels tests.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_COV.3.1D **Le développeur doit fournir une analyse de la couverture des tests.**

Contenu et présentation des éléments de preuve :

ATE_COV.3.1C **L'analyse de la couverture des tests doit démontrer la correspondance entre les tests identifiés dans la documentation de test et la TSF, telle qu'elle est décrite dans les spécifications fonctionnelles.**

ATE_COV.3.2C L'analyse de la couverture des tests doit démontrer que la correspondance entre la TSF, telle qu'elle est décrite dans les spécifications fonctionnelles, et les tests identifiés dans la documentation de test est complète.

ATE_COV.3.3C **L'analyse de la couverture des tests doit démontrer de façon rigoureuse que toutes les interfaces externes de la TSF identifiées dans les spécifications fonctionnelles ont été complètement testées.**

Tâches de l'évaluateur :

ATE_COV.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

13.2 Profondeur (ATE_DPT)

Objectifs

419 Les composants de cette famille traitent du niveau de détail avec lequel la TSF est testée. Les tests des fonctions de sécurité sont basés sur la profondeur croissante des informations obtenues de l'analyse des représentations.

420 L'objectif est de réduire le risque de ne pas repérer une erreur pendant le développement de la TOE. De plus, les composants de la présente famille, en particulier lorsque les tests concernent plus spécialement la structure interne de la TSF, sont plus susceptibles de permettre de découvrir tout code malveillant qui aurait été introduit.

421 Les tests destinés aux interfaces internes spécifiques peuvent procurer l'assurance non seulement sur le fait que la TSF produit le comportement externe de sécurité souhaité, mais aussi que ce comportement provient du fonctionnement correct de mécanismes internes.

Classement des composants

422 Les composants de cette famille sont classés suivant le caractère de plus en plus détaillé des représentations de la TSF, depuis la conception de haut niveau jusqu'à la représentation de l'implémentation. Ce classement reflète les représentations de la TSF présentées dans la classe ADV.

Notes d'application

423 Le volume et le type spécifiques de documentation et d'éléments de preuve seront déterminés, en général, par le composant choisi dans ATE_FUN.

424 Les tests correspondant aux spécifications fonctionnelles sont traités par la famille ATE_COV.

425 Le principe adopté dans la présente famille est que le niveau des tests doit être cohérent avec le niveau d'assurance recherché. Lorsque des composants plus élevés sont appliqués, les résultats des tests devront démontrer que l'implémentation de la TSF est conforme à sa conception. Par exemple, la conception de haut niveau devrait décrire chacun des sous-systèmes et décrire également les interfaces entre ces sous-systèmes avec suffisamment de détails. Les éléments de preuve des tests doivent montrer que les interfaces internes entre les sous-systèmes ont été utilisées. Cela peut être obtenu par des tests via les interfaces externes de la TSF ou par des tests des interfaces des sous-systèmes en les isolant et en bridant éventuellement les tests. Dans les cas où certains aspects d'une interface interne ne peuvent pas être testés via les interfaces externes, alors soit une justification devra être apportée indiquant que ces aspects n'ont pas besoin d'être testés, soit l'interface interne devra être testée directement. Dans le dernier cas, la conception de haut niveau doit être suffisamment détaillée afin de faciliter les tests directs. Les composants les plus élevés dans cette famille visent à vérifier le fonctionnement correct des interfaces

internes qui deviennent visibles quand la conception devient moins abstraite. Une fois que ces composants auront été appliqués, il sera plus difficile de fournir des éléments de preuve adéquats de la profondeur des tests en utilisant seulement les interfaces externes de la TSF, et des tests modulaires seront généralement nécessaires.

ATE_DPT.1 Tests : conception de haut niveau

Objectifs

426 Les sous-systèmes d'une TSF fournissent une description générale du fonctionnement interne de la TSF. Les tests effectués au niveau des sous-systèmes pour démontrer la présence d'anomalies, procurent l'assurance que les sous-systèmes de la TSF ont été correctement réalisés.

Notes d'application

427 On attend du développeur qu'il décrive les tests de la conception de haut niveau de la TSF en termes de "sous-systèmes". Le terme "sous-système" est utilisé pour exprimer la notion de décomposition de la TSF dans un nombre de parties relativement faible.

Dépendances :

ADV_HLD.1 Conception de haut niveau descriptive

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_DPT.1.1D **Le développeur doit fournir l'analyse de profondeur des tests.**

Contenu et présentation des éléments de preuve :

ATE_DPT.1.1C **L'analyse de profondeur doit démontrer que les tests identifiés dans la documentation de test sont suffisants pour démontrer que la TSF fonctionne conformément à sa conception de haut niveau.**

Tâches de l'évaluateur :

ATE_DPT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ATE_DPT.2 Tests : conception de bas niveau

Objectifs

428 Les sous-systèmes d'une TSF fournissent une description générale du fonctionnement interne de la TSF. Les tests effectués au niveau des sous-systèmes pour démontrer la présence d'anomalies, procurent l'assurance que les sous-systèmes de la TSF ont été correctement réalisés.

429 Les modules d'une TSF fournissent une description du fonctionnement interne de la TSF. Les tests effectués au niveau des modules pour démontrer la présence d'anomalies, procurent l'assurance que les modules de la TSF ont été correctement réalisés.

Notes d'application

430 On attend du développeur qu'il décrive les tests de la conception de haut niveau de la TSF en termes de "sous-systèmes". Le terme "sous-système" est utilisé pour exprimer la notion de décomposition de la TSF dans un nombre de parties relativement faible.

431 On attend du développeur qu'il décrive les tests de la conception de bas niveau de la TSF en termes de "modules". Le terme "module" est utilisé pour exprimer la notion de décomposition de chaque "sous-système" de la TSF dans un nombre de parties relativement faible.

Dépendances :

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

ADV_LLD.1 Conception de bas niveau descriptive

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_DPT.2.1D **Le développeur doit fournir l'analyse de profondeur des tests.**

Contenu et présentation des éléments de preuve :

ATE_DPT.2.1C L'analyse de profondeur doit démontrer que les tests identifiés dans la documentation de test sont suffisants pour démontrer que la TSF fonctionne conformément à sa conception de haut niveau **et à sa conception de bas niveau**.

Tâches de l'évaluateur :

ATE_DPT.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ATE_DPT.3 Tests : représentation de l'implémentation

Objectifs

432 Les sous-systèmes d'une TSF fournissent une description générale du fonctionnement interne de la TSF. Les tests effectués au niveau des sous-systèmes pour démontrer la présence d'anomalies, procurent l'assurance que les sous-systèmes de la TSF ont été correctement réalisés.

433 Les modules d'une TSF fournissent une description du fonctionnement interne de la TSF. Les tests effectués au niveau des modules pour démontrer la présence

d'anomalies, procurent l'assurance que les modules de la TSF ont été correctement réalisés.

434 La représentation de l'implémentation d'une TSF fournit une description détaillée du fonctionnement interne de la TSF. Les tests effectués au niveau de l'implémentation pour démontrer la présence d'anomalies, procurent l'assurance que l'implémentation de la TSF a été correctement réalisée.

Notes d'application

435 On attend du développeur qu'il décrive les tests de la conception de haut niveau de la TSF en termes de "sous-systèmes". Le terme "sous-système" est utilisé pour exprimer la notion de décomposition de la TSF dans un nombre de parties relativement faible.

436 On attend du développeur qu'il décrive les tests de la conception de bas niveau de la TSF en termes de "modules". Le terme "module" est utilisé pour exprimer la notion de décomposition de chaque "sous-système" de la TSF dans un nombre de parties relativement faible.

437 La représentation de l'implémentation est celle utilisée pour générer la TSF elle-même (e.g. le code source qui est ensuite compilé).

Dépendances :

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

ADV_IMP.2 Implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_DPT.3.1D Le développeur doit fournir l'analyse de profondeur des tests.

Contenu et présentation des éléments de preuve :

ATE_DPT.3.1C L'analyse de profondeur doit démontrer que les tests identifiés dans la documentation de test sont suffisants pour démontrer que la TSF fonctionne conformément à sa conception de haut niveau, à sa conception de bas niveau **et à la représentation de son implémentation**.

Tâches de l'évaluateur :

ATE_DPT.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

13.3 Tests fonctionnels (ATE_FUN)

Objectifs

438 Les tests fonctionnels réalisés par le développeur établissent que la TSF montre les propriétés nécessaires pour satisfaire aux exigences fonctionnelles des PP ou ST correspondants. De tels tests fonctionnels procurent l'assurance que la TSF satisfait au moins aux exigences fonctionnelles de sécurité, bien qu'ils ne peuvent établir que la TSF ne fait rien de plus que ce qui a été spécifié. La famille "Tests fonctionnels" est centrée sur le type et le volume de documentation ou d'outils de support exigés et sur ce qui doit être démontré au moyen des tests du développeur. Les tests fonctionnels ne sont pas limités à la confirmation que les fonctions de sécurité exigées sont fournies, mais peuvent également inclure des tests à caractère négatif (souvent basés sur l'inversion des exigences fonctionnelles) pour vérifier l'absence de comportements particuliers non souhaités.

439 La présente famille contribue à procurer l'assurance que la probabilité d'anomalies non découvertes est relativement faible.

440 Les familles ATE_COV, ATE_DPT et ATE_FUN sont utilisées de façon combinée pour définir les éléments de preuve des tests à fournir par un développeur. Les tests fonctionnels indépendants de l'évaluateur sont spécifiés par ATE_IND.

Classement des composants

441 La présente famille contient deux composants, le composant le plus élevé exigeant que l'ordonnancement soit analysé.

Notes d'application

442 Les procédures d'exécution des tests sont censées donner des instructions d'utilisation des programmes et des suites de test, ce qui inclut l'environnement de test, les conditions de test, les paramètres et les valeurs des données de test. Les procédures de test devraient aussi montrer comment les résultats des tests sont déduits des données d'entrée des tests.

443 La présente famille spécifie les exigences pour la présentation de tous les plans, procédures et résultats des tests. Ainsi la quantité d'informations qui doit être présentée varie suivant l'utilisation de ATE_COV et de ATE_DPT.

444 L'ordonnancement est pertinent lorsque l'exécution réussie d'un test particulier dépend de l'existence d'un état particulier. Par exemple, on pourrait exiger que le test A soit exécuté immédiatement avant le test B, puisque l'état résultant de l'exécution réussie du test A constitue une condition préalable pour la réussite de l'exécution du test B. Ainsi, l'échec du test B pourrait avoir un rapport avec l'ordonnancement. Dans l'exemple ci-dessus, le test B pourrait échouer parce que le test C (plutôt que le test A) aurait été exécuté immédiatement avant lui, ou bien l'échec du test B pourrait être lié à l'échec du test A.

ATE_FUN.1 Tests fonctionnels

Objectifs

445 L'objectif pour le développeur est de démontrer que toutes les fonctions de sécurité fonctionnent conformément à leurs spécifications. Le développeur doit exécuter les tests et fournir la documentation de test.

Dépendances :

Pas de dépendances.

Tâches du développeur :

ATE_FUN.1.1D **Le développeur doit tester la TSF et documenter les résultats.**

ATE_FUN.1.2D **Le développeur doit fournir la documentation de test.**

Contenu et présentation des éléments de preuve :

ATE_FUN.1.1C **La documentation de test doit être constituée des plans de test, des descriptions de procédures de test, des résultats de tests attendus et des résultats de tests réellement obtenus.**

ATE_FUN.1.2C **Les plans de test doivent identifier les fonctions de sécurité à tester et décrire le but des tests à exécuter.**

ATE_FUN.1.3C **Les descriptions des procédures de test doivent identifier les tests à exécuter et décrire les scenarii de test de chaque fonction de sécurité. Ces scénario doivent inclure tous les ordonnancements relatifs aux résultats des autres tests.**

ATE_FUN.1.4C **Les résultats de tests attendus doivent montrer les résultats prévus à la suite d'une exécution réussie des tests.**

ATE_FUN.1.5C **Les résultats de tests provenant de l'exécution des tests par le développeur doivent démontrer que chaque fonction de sécurité testée s'est comportée conformément à ses spécifications.**

Tâches de l'évaluateur :

ATE_FUN.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ATE_FUN.2 Tests fonctionnels ordonnés

Objectifs

446 L'objectif pour le développeur est de démontrer que toutes les fonctions de sécurité fonctionnent conformément à leurs spécifications. Le développeur doit exécuter les tests et fournir la documentation de test.

447 Dans ce composant, un objectif supplémentaire est d'assurer que les tests sont structurés de façon à éviter que soit remise en cause la conformité des parties de la TSF qui sont testées.

Notes d'application

448 Bien que les procédures de test puissent stipuler des conditions de tests initiales préalables en terme d'ordonnancement des tests, elles peuvent ne pas fournir un argumentaire pour l'ordonnancement. Une analyse d'ordonnancement des tests constitue un facteur important pour déterminer l'adéquation des tests, car il existe une possibilité que des erreurs soient dissimulées à cause de l'ordonnancement des tests.

Dépendances :

Pas de dépendances.

Tâches du développeur :

ATE_FUN.2.1D Le développeur doit tester la TSF et documenter les résultats.

ATE_FUN.2.2D Le développeur doit fournir la documentation de test.

Contenu et présentation des éléments de preuve :

ATE_FUN.2.1C **La documentation de test doit être constituée des plans de test, des descriptions de procédures de test, des résultats de tests attendus et des résultats de tests réellement obtenus.**

ATE_FUN.2.2C Les plans de test doivent identifier les fonctions de sécurité à tester et décrire le but des tests à exécuter.

ATE_FUN.2.3C Les descriptions des procédures de test doivent identifier les tests à exécuter et décrire les scenarii de test de chaque fonction de sécurité. Ces scénario doivent inclure tous les ordonnancements relatifs aux résultats des autres tests.

ATE_FUN.2.4C Les résultats de tests attendus doivent montrer les résultats prévus à la suite d'une exécution réussie des tests.

ATE_FUN.2.5C Les résultats de tests provenant de l'exécution des tests par le développeur doivent démontrer que chaque fonction de sécurité testée s'est comportée conformément à ses spécifications.

ATE_FUN.2.6C **La documentation de test doit inclure une analyse de l'ordonnancement dans la procédure de test.**

Tâches de l'évaluateur :

ATE_FUN.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

13.4 Tests indépendants (ATE_IND)

Objectifs

449 Un des objectifs consiste à démontrer que les fonctions de sécurité fonctionnent conformément à leurs spécifications.

450 Un objectif supplémentaire est de réduire le risque d'une mauvaise estimation des résultats de tests produits par le développeur, ce qui aurait pour conséquence l'implémentation incorrecte des spécifications ou risquerait de négliger une partie du code non conforme aux spécifications.

Classement des composants

451 Le classement est basé sur le volume de la documentation de test, le support de test et la quantité de tests effectués par l'évaluateur.

Notes d'application

452 Les tests spécifiés dans la présente famille peuvent être mis en œuvre par une partie autre que l'évaluateur, dotée d'une connaissance spécialisée (e.g. un laboratoire indépendant, une organisation de consommateurs objective). Les tests nécessitent une compréhension de la TOE à un degré cohérent avec les autres activités d'assurance effectuées, et l'évaluateur conserve la responsabilité de garantir que les exigences de la présente famille sont correctement prises en compte dans ce cas.

453 La présente famille traite du degré d'indépendance avec lequel les tests fonctionnels de la TSF sont effectués. Les tests fonctionnels indépendants peuvent consister en une réexécution des tests fonctionnels du développeur, en totalité ou en partie. Ils peuvent aussi consister en un enrichissement de ces derniers, soit en étendant le champ d'application ou la profondeur des tests du développeur, soit en testant les faiblesses évidentes de sécurité qui relèvent du domaine public et qui pourraient être applicables à la TOE. Ces activités sont complémentaires et une combinaison appropriée de celles-ci doit être planifiée pour chaque TOE, prenant en compte la disponibilité et la couverture des résultats des tests, ainsi que la complexité fonctionnelle de la TSF. Un plan de test devrait être développé en cohérence avec le niveau des autres activités de l'assurance et qui, lorsqu'une assurance plus élevée est exigée, inclut des échantillons plus larges de tests à réexécuter, ainsi que plus de tests fonctionnels indépendants, à caractères positif et négatif, à effectuer par l'évaluateur.

454 L'échantillonnage des tests du développeur est destiné à apporter la confirmation que le développeur a bien mené son programme de test de la TSF planifié et a enregistré correctement les résultats. La taille de l'échantillon sélectionné sera influencée par le niveau de détail et la qualité des résultats des tests fonctionnels du développeur. L'évaluateur devra aussi prendre en compte le champ d'application afin d'imaginer des tests supplémentaires, ainsi que l'avantage relatif qui peut être obtenu d'un effort consenti dans ces deux domaines. Il est admis qu'il peut être faisable et souhaitable de réexécuter tous les tests du développeur dans certains cas,

mais que cela peut se révéler très ardu et moins productif dans d'autres cas. Le composant le plus élevé de cette famille devrait par conséquent être utilisé avec précaution. L'échantillonnage prendra en compte l'étendue complète des résultats des tests disponibles, y compris ceux fournis pour satisfaire aux exigences de ATE_COV et de ATE_DPT.

455 Il existe également un besoin de prendre en compte les différentes configurations de la TOE qui sont comprises dans l'évaluation. L'évaluateur devra estimer le degré d'applicabilité des résultats fournis et planifier ses propres tests en conséquence.

456 Les tests fonctionnels indépendants sont distincts des tests de pénétration, ces derniers étant basés sur une recherche avertie et systématique des vulnérabilités introduites dans la conception ou dans l'implémentation. Les tests de pénétration sont spécifiés au moyen de la famille AVA_VLA.

457 Le fait que la TOE se prête bien aux tests repose sur l'accès à celle-ci, ainsi qu'à la documentation et aux informations de support exigées (incluant tous logiciels ou outils de test) pour exécuter les tests. Le besoin d'un tel support est pris en compte par les dépendances vers d'autres familles d'assurance.

458 De plus, la façon dont la TOE se prête aux tests peut être basée sur d'autres considérations. Par exemple, la version de la TOE soumise par le développeur peut ne pas être la version finale.

459 Les références à un sous-ensemble de la TSF sont prévues pour permettre à l'évaluateur de concevoir un ensemble approprié de tests qui soit cohérent avec les objectifs de l'évaluation en cours.

ATE_IND.1 Tests indépendants - conformité

Objectifs

460 Dans ce composant, l'objectif est de démontrer que les fonctions de sécurité fonctionnent conformément à leurs spécifications.

Notes d'application

461 Le présent composant ne traite pas de l'utilisation des résultats des tests du développeur. Il est applicable quand de tels résultats ne sont pas disponibles et aussi dans les cas où les tests du développeur sont acceptés sans aucune validation. L'évaluateur doit imaginer et exécuter les tests avec l'objectif de confirmer que les exigences fonctionnelles de sécurité de la TOE sont satisfaites. L'approche retenue consiste à obtenir la confiance dans un fonctionnement correct au moyen de tests représentatifs, plutôt que d'exécuter tous les tests possibles. L'étendue des tests à planifier dans ce but constitue un problème de méthodologie, et nécessite d'être pris en compte dans le contexte d'une TOE particulière, en respectant l'équilibre avec les autres activités de l'évaluation.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

ATE_IND.1.1D **Le développeur doit fournir la TOE afin d'exécuter les tests.**

Contenu et présentation des éléments de preuve :

ATE_IND.1.1C **La TOE doit se prêter à l'exécution de tests.**

Tâches de l'évaluateur :

ATE_IND.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

ATE_IND.1.2E **L'évaluateur doit tester un sous-ensemble de la TSF quand cela est approprié pour confirmer que la TOE fonctionne conformément à ses spécifications.**

ATE_IND.2 Tests indépendants - échantillonnage

Objectifs

462 L'objectif est de démontrer que les fonctions de sécurité fonctionnent conformément à leurs spécifications. Les tests de l'évaluateur comprennent la sélection et la réexécution d'un échantillon des tests du développeur.

Notes d'application

463 L'intention est de faire en sorte que le développeur fournisse à l'évaluateur les éléments nécessaires pour reproduire efficacement les tests du développeur. Cela peut impliquer par exemple une documentation de test au format électronique, des programmes de tests, etc.

464 Le présent composant contient une exigence indiquant que l'évaluateur dispose des résultats des tests du développeur pour compléter le programme de test. L'évaluateur exécutera à nouveau un échantillon des tests du développeur pour acquérir la confiance dans les résultats obtenus. Après avoir établi une telle confiance, l'évaluateur s'appuiera sur les tests du développeur pour exécuter des tests supplémentaires qui font fonctionner la TOE d'une manière différente. En utilisant une base de résultats de tests du développeur validés, l'évaluateur est capable d'acquérir la confiance que la TOE fonctionne correctement pour un ensemble de conditions plus étendu que ce qu'il serait possible d'obtenir en ne prenant en compte que les propres efforts du développeur, étant donné un niveau de ressource fixé. Après avoir acquis la confiance que le développeur a testé la TOE, l'évaluateur disposera d'une plus grande liberté, quand cela est approprié, pour

concentrer les tests dans des domaines où l'examen de la documentation ou bien une connaissance spécialisée a soulevé des préoccupations particulières.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_IND.2.1D Le développeur doit fournir la TOE afin d'exécuter les tests.

Contenu et présentation des éléments de preuve :

ATE_IND.2.1C La TOE doit se prêter à l'exécution de tests.

ATE_IND.2.2C **Le développeur doit fournir un ensemble de ressources équivalent à celui qui a été utilisé pour les tests fonctionnels de la TSF qu'il a effectués.**

Tâches de l'évaluateur :

ATE_IND.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ATE_IND.2.2E L'évaluateur doit tester un sous-ensemble de la TSF quand cela est approprié pour confirmer que la TOE fonctionne conformément à ses spécifications.

ATE_IND.2.3E **L'évaluateur doit exécuter un échantillon de tests choisi dans la documentation de test afin de contrôler les résultats des tests du développeur.**

ATE_IND.3 Tests indépendants - totalité

Objectifs

465 L'objectif est de démontrer que toutes les fonctions de sécurité fonctionnent conformément à leurs spécifications. Les tests de l'évaluateur comprennent la réexécution de tous les tests du développeur.

Notes d'application

466 L'intention est de faire en sorte que le développeur fournisse à l'évaluateur les éléments nécessaires pour reproduire efficacement les tests du développeur. Cela peut impliquer par exemple une documentation de test au format électronique, des programmes de tests, etc.

467 Dans ce composant, l'évaluateur doit exécuter à nouveau tous les tests du développeur comme partie du programme de test. Comme dans le composant

précédent, l'évaluateur exécutera également des tests qui visent à faire fonctionner la TOE d'une manière différente de celle que le développeur a utilisée. Dans les cas où les tests du développeur se sont révélés exhaustifs, la probabilité de pouvoir imaginer de tels tests peut se révéler très faible.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

ATE_FUN.1 Tests fonctionnels

Tâches du développeur :

ATE_IND.3.1D Le développeur doit fournir la TOE afin d'exécuter les tests.

Contenu et présentation des éléments de preuve :

ATE_IND.3.1C La TOE doit se prêter à l'exécution de tests.

ATE_IND.3.2C Le développeur doit fournir un ensemble de ressources équivalent à celui qui a été utilisé pour les tests fonctionnels de la TSF qu'il a effectués.

Tâches de l'évaluateur :

ATE_IND.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ATE_IND.3.2E L'évaluateur doit tester un sous-ensemble de la TSF quand cela est approprié pour confirmer que la TOE fonctionne conformément à ses spécifications.

ATE_IND.3.3E L'évaluateur doit exécuter **tous** les tests figurant dans la documentation de test afin de contrôler les résultats des tests du développeur.

14 Classe AVA : Estimation des vulnérabilités

468 Cette classe traite de l'existence de canaux cachés exploitables, de la possibilité d'avoir une configuration de la TOE qui soit impropre ou incorrecte, de la possibilité de mettre en échec les mécanismes faisant appel au calcul des probabilités ou des permutations et de la possibilité que des vulnérabilités exploitables soient introduites pendant le développement ou l'exploitation de la TOE.

469 La figure 14.1 présente les familles de cette classe et la hiérarchie des composants au sein des familles.

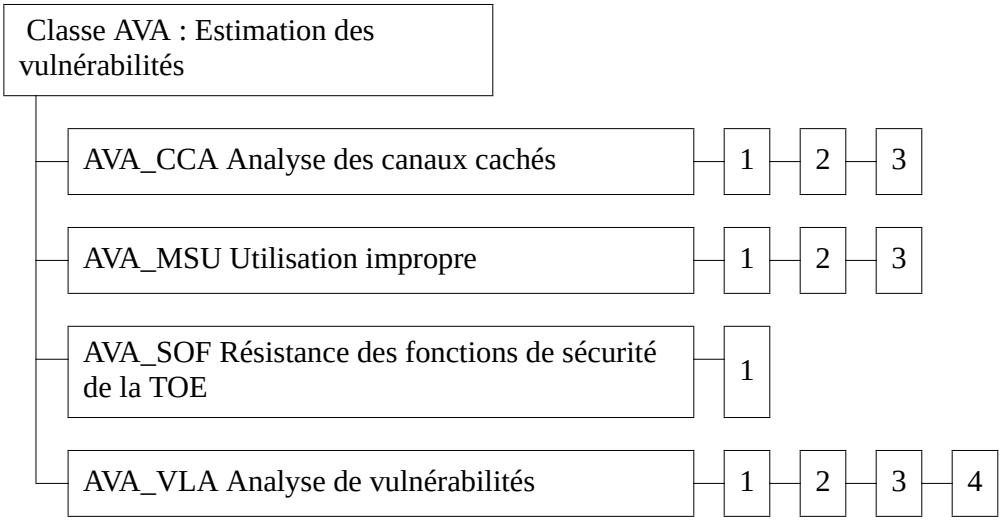

Figure 14.1 -Décomposition de la classe "Estimation des vulnérabilités"

14.1 Analyse des canaux cachés (AVA_CCA)

Objectifs

470 L'analyse des canaux cachés est menée afin de déterminer l'existence et la capacité potentielle des canaux de communication non intentionnels (i.e. des flux d'information illicites) qui peuvent être exploités.

471 Les exigences d'assurance concernent la menace qu'il existe des chemins de communication non intentionnels et exploitables, pouvant être utilisés pour violer la SFP.

Classement des composants

472 Les composants sont classés suivant la rigueur croissante de l'analyse des canaux cachés.

Notes d'application

473 Les estimations de capacité des canaux sont basées sur des mesures techniques informelles, ainsi que sur de véritables mesures provenant de tests.

474 Parmi des exemples d'hypothèses sur lesquelles se base l'analyse des canaux cachés, on peut citer la vitesse du processeur, la configuration du système ou du réseau, la taille mémoire et la taille de la mémoire cache.

475 La validation sélective de l'analyse des canaux cachés par des tests donne à l'évaluateur l'opportunité de contrôler n'importe quel aspect de l'analyse des canaux cachés (e.g. les scenarii d'identification, d'estimation de capacité, d'élimination, de surveillance et d'exploitation). Cette validation n'impose pas d'exigence pour démontrer l'ensemble complet des résultats d'analyse des canaux cachés.

476 S'il n'existe pas de SFP de contrôle de flux d'informations dans la ST, la présente famille d'exigences d'assurance n'est plus applicable, car elle ne s'applique qu'à des SFP de contrôle de flux d'informations.

AVA_CCA.1 Analyse des canaux cachés

Objectifs

477 L'objectif consiste à identifier les canaux cachés qui sont identifiables, au moyen d'une recherche informelle des canaux cachés.

Dépendances :

ADV_FSP.2 Définition exhaustive des interfaces externes

ADV_IMP.2 Implémentation de la TSF

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_CCA.1.1D **Le développeur doit conduire une recherche des canaux cachés pour chaque politique de contrôle de flux d'informations.**

AVA_CCA.1.2D **Le développeur doit fournir une documentation d'analyse des canaux cachés.**

Contenu et présentation des éléments de preuve :

AVA_CCA.1.1C **La documentation d'analyse doit identifier les canaux cachés et estimer leur capacité.**

AVA_CCA.1.2C **La documentation d'analyse doit décrire les procédures utilisées pour déterminer l'existence des canaux cachés, ainsi que les informations nécessaires pour mener l'analyse des canaux cachés.**

AVA_CCA.1.3C **La documentation d'analyse doit décrire toutes les hypothèses faites pendant l'analyse des canaux cachés.**

AVA_CCA.1.4C **La documentation d'analyse doit décrire la méthode utilisée pour estimer la capacité des canaux, en se basant sur les pires scenarii.**

AVA_CCA.1.5C **La documentation d'analyse doit décrire le pire scénario d'exploitation pour chaque canal caché identifié.**

Tâches de l'évaluateur :

AVA_CCA.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AVA_CCA.1.2E **L'évaluateur doit confirmer que les résultats de l'analyse des canaux cachés montrent que la TOE satisfait à ses exigences fonctionnelles.**

AVA_CCA.1.3E **L'évaluateur doit valider sélectivement l'analyse des canaux cachés par des tests.**

AVA_CCA.2 Analyse systématique des canaux cachés

Objectifs

478 L'objectif consiste à identifier les canaux cachés qui sont identifiables, au moyen d'une recherche systématique des canaux cachés.

Notes d'application

479 La réalisation d'une analyse des canaux cachés d'une façon systématique exige que le développeur identifie les canaux cachés d'une manière structurée et répétable, par opposition à une manière opportuniste.

Dépendances :

ADV_FSP.2 Définition exhaustive des interfaces externes

ADV_IMP.2 Implémentation de la TSF

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_CCA.2.1D Le développeur doit conduire une recherche des canaux cachés pour chaque politique de contrôle de flux d'informations.

AVA_CCA.2.2D Le développeur doit fournir une documentation d'analyse des canaux cachés.

Contenu et présentation des éléments de preuve :

AVA_CCA.2.1C La documentation d'analyse doit identifier les canaux cachés et estimer leur capacité.

AVA_CCA.2.2C La documentation d'analyse doit décrire les procédures utilisées pour déterminer l'existence des canaux cachés, ainsi que les informations nécessaires pour mener l'analyse des canaux cachés.

AVA_CCA.2.3C La documentation d'analyse doit décrire toutes les hypothèses faites pendant l'analyse des canaux cachés.

AVA_CCA.2.4C La documentation d'analyse doit décrire la méthode utilisée pour estimer la capacité des canaux, en se basant sur les pires scenarii.

AVA_CCA.2.5C La documentation d'analyse doit décrire le pire scénario d'exploitation pour chaque canal caché identifié.

AVA_CCA.2.6C **La documentation d'analyse doit fournir des éléments de preuve indiquant que la méthode utilisée pour identifier les canaux cachés est systématique.**

Tâches de l'évaluateur :

AVA_CCA.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_CCA.2.2E L'évaluateur doit confirmer que les résultats de l'analyse des canaux cachés montrent que la TOE satisfait à ses exigences fonctionnelles.

AVA_CCA.2.3E L'évaluateur doit valider sélectivement l'analyse des canaux cachés par des tests.

AVA_CCA.3 Analyse exhaustive des canaux cachés

Objectifs

480 L'objectif consiste à identifier les canaux cachés qui sont identifiables, au moyen d'une recherche exhaustive des canaux cachés.

Notes d'application

481 La réalisation d'une analyse des canaux cachés d'une façon exhaustive exige que des éléments de preuve supplémentaires soient fournis, indiquant que le plan qui a été suivi pour identifier les canaux cachés est suffisant pour garantir que toutes les façons possibles d'explorer les canaux cachés ont été utilisées.

Dépendances :

ADV_FSP.2 Définition exhaustive des interfaces externes

ADV_IMP.2 Implémentation de la TSF

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_CCA.3.1D Le développeur doit conduire une recherche des canaux cachés pour chaque politique de contrôle de flux d'informations.

AVA_CCA.3.2D Le développeur doit fournir une documentation d'analyse des canaux cachés.

Contenu et présentation des éléments de preuve :

AVA_CCA.3.1C La documentation d'analyse doit identifier les canaux cachés et estimer leur capacité.

AVA_CCA.3.2C La documentation d'analyse doit décrire les procédures utilisées pour déterminer l'existence des canaux cachés, ainsi que les informations nécessaires pour mener l'analyse des canaux cachés.

AVA_CCA.3.3C La documentation d'analyse doit décrire toutes les hypothèses faites pendant l'analyse des canaux cachés.

AVA_CCA.3.4C La documentation d'analyse doit décrire la méthode utilisée pour estimer la capacité des canaux, en se basant sur les pires scenarii.

AVA_CCA.3.5C La documentation d'analyse doit décrire le pire scénario d'exploitation pour chaque canal caché identifié.

AVA_CCA.3.6C La documentation d'analyse doit fournir des éléments de preuve indiquant que la méthode utilisée pour identifier les canaux cachés est **exhaustive**.

Tâches de l'évaluateur :

AVA_CCA.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_CCA.3.2E L'évaluateur doit confirmer que les résultats de l'analyse des canaux cachés montrent que la TOE satisfait à ses exigences fonctionnelles.

AVA_CCA.3.3E L'évaluateur doit valider sélectivement l'analyse des canaux cachés par des tests.

14.2 Utilisation impropre (AVA_MSU)

Objectifs

482 La famille Utilisation impropre permet de rechercher si la TOE peut être configurée ou utilisée d'une manière qui n'est pas sûre mais qu'un administrateur ou un utilisateur de la TOE pourrait raisonnablement croire sûre.

483 Les objectifs sont :

a) de minimiser la probabilité de configurer ou d'installer la TOE d'une manière non sûre, sans que l'utilisateur ou l'administrateur soit capable de le détecter ;

b) de minimiser le risque d'erreurs humaines ou autres dans l'exploitation qui puissent désactiver, mettre hors d'état ou empêcher d'activer des fonctions de sécurité, ce qui aurait pour résultat un état non sûr et non détecté.

Classement des composants

484 Les composants sont classés suivant le nombre croissant des éléments de preuve devant être fournis par le développeur et la rigueur croissante de l'analyse.

Notes d'application

485 Des guides contradictoires, trompeurs, incomplets ou déraisonnables peuvent avoir pour conséquence qu'un utilisateur de la TOE croit qu'elle est sûre alors qu'elle ne l'est pas, et peuvent provoquer l'apparition de vulnérabilités.

486 Comme exemple de guides contradictoires, on pourrait citer le cas de deux instructions dont l'application entraînerait des résultats différents quand les mêmes données d'entrée sont fournies.

487 Comme exemple de guide trompeur, on pourrait citer le cas d'une instruction unique dont la description pourrait être analysée de plusieurs façons, l'une d'entre elles ayant pour conséquence un état non sûr.

488 Un guide incomplet serait par exemple constitué d'une liste d'exigences de sécurité physiques significatives dont un élément important aurait été omis, avec pour conséquence le fait que cet élément échappe à l'administrateur, qui pense que la liste est complète.

489 Un guide déraisonnable serait par exemple constitué par une recommandation de suivre une procédure imposant indûment une charge administrative onéreuse.

490 Des guides sont exigés. Ils peuvent être constitués par les guides de l'utilisateur ou de l'administrateur existants, ou être fournis séparément. Dans ce cas, l'évaluateur devrait confirmer que la documentation est fournie en même temps que la TOE.

AVA_MSU.1 Examen des guides

Objectifs

491 L'objectif consiste à garantir que des éléments trompeurs, déraisonnables et contradictoires sont absents des guides et que les procédures sûres pour tous les modes d'exploitation ont été prises en compte. Les états non sûrs devraient être faciles à détecter.

Dépendances :

ADO_IGS.1 Procédures d'installation, de génération et de démarrage

ADV_FSP.1 Spécifications fonctionnelles informelles

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_MSU.1.1D **Le développeur doit fournir des guides.**

Contenu et présentation des éléments de preuve :

AVA_MSU.1.1C **Les guides doivent identifier tous les modes possibles d'exploitation de la TOE (comprenant la reprise d'exploitation suite à une défaillance ou des erreurs d'exploitation), leurs conséquences et leurs implications pour maintenir une exploitation sûre.**

AVA_MSU.1.2C **Les guides doivent être complets, clairs, cohérents et raisonnables.**

AVA_MSU.1.3C **Les guides doivent énumérer toutes les hypothèses relatives à l'environnement prévu.**

AVA_MSU.1.4C **Les guides doivent énumérer toutes les exigences pour les mesures de sécurité externes (comprenant les contrôles externes organisationnels, physiques et touchant au personnel).**

Tâches de l'évaluateur :

AVA_MSU.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AVA_MSU.1.2E **L'évaluateur doit appliquer à nouveau toutes les procédures de configuration et d'installation afin de confirmer que la TOE peut être configurée et utilisée de manière sûre en utilisant seulement les guides fournis.**

AVA_MSU.1.3E **L'évaluateur doit déterminer si l'utilisation des guides permet de détecter tous les états non sûrs.**

AVA_MSU.2 Validation de l'analyse

Objectifs

492 L'objectif consiste à garantir que des éléments trompeurs, déraisonnables et contradictoires sont absents des guides et que les procédures sûres pour tous les modes d'exploitation ont été prises en compte. Les états non sûrs devraient être faciles à détecter. Dans le présent composant, une analyse des guides est exigée du développeur afin de fournir une assurance supplémentaire que l'objectif a été atteint.

Dépendances :

ADO_IGS.1 Procédures d'installation, de génération et de démarrage

ADV_FSP.1 Spécifications fonctionnelles informelles

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_MSU.2.1D Le développeur doit fournir des guides.

AVA_MSU.2.2D **Le développeur doit fournir une analyse des guides.**

Contenu et présentation des éléments de preuve :

AVA_MSU.2.1C Les guides doivent identifier tous les modes possibles d'exploitation de la TOE (comprenant la reprise d'exploitation suite à une défaillance ou des erreurs d'exploitation), leurs conséquences et leurs implications pour maintenir une exploitation sûre.

AVA_MSU.2.2C Les guides doivent être complets, clairs, cohérents et raisonnables.

AVA_MSU.2.3C Les guides doivent énumérer toutes les hypothèses relatives à l'environnement prévu.

AVA_MSU.2.4C Les guides doivent énumérer toutes les exigences pour les mesures de sécurité externes (comprenant les contrôles externes organisationnels, physiques et touchant au personnel).

AVA_MSU.2.5C **L'analyse de la documentation doit démontrer que les guides sont complets.**

Tâches de l'évaluateur :

AVA_MSU.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_MSU.2.2E L'évaluateur doit appliquer à nouveau toutes les procédures de configuration, d'installation **et sélectivement d'autres procédures**, afin de confirmer que la TOE

peut être configurée et utilisée de manière sûre en utilisant seulement les guides fournis.

AVA_MSU.2.3E L'évaluateur doit déterminer si l'utilisation des guides permet de détecter tous les états non sûrs.

AVA_MSU.2.4E **L'évaluateur doit confirmer que la documentation d'analyse montre que sont donnés des conseils pour une exploitation sûre de la TOE dans tous les modes d'exploitation.**

AVA_MSU.3 Analyse et test des états non sûrs

Objectifs

493 L'objectif consiste à garantir que des éléments trompeurs, déraisonnables et contradictoires sont absents des guides et que les procédures sûres pour tous les modes d'exploitation ont été prises en compte. Les états non sûrs devraient être faciles à détecter. Dans le présent composant, une analyse des guides est exigée du développeur afin de fournir une assurance supplémentaire que l'objectif a été atteint et cette analyse est ensuite validée et confirmée par l'évaluateur au moyen de tests.

Notes d'application

494 Dans le présent composant, l'évaluateur doit entreprendre des tests afin de s'assurer que dans l'éventualité où la TOE se met dans un état non sûr, cela peut être facilement détecté. Ces tests peuvent être considérés comme un aspect spécifique des tests de pénétration.

Dépendances :

ADO_IGS.1 Procédures d'installation, de génération et de démarrage

ADV_FSP.1 Spécifications fonctionnelles informelles

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_MSU.3.1D Le développeur doit fournir des guides.

AVA_MSU.3.2D Le développeur doit fournir une analyse des guides.

Contenu et présentation des éléments de preuve :

AVA_MSU.3.1C Les guides doivent identifier tous les modes possibles d'exploitation de la TOE (comprenant la reprise d'exploitation suite à une défaillance ou des erreurs d'exploitation), leurs conséquences et leurs implications pour maintenir une exploitation sûre.

AVA_MSU.3.2C Les guides doivent être complets, clairs, cohérents et raisonnables.

AVA_MSU.3.3C Les guides doivent énumérer toutes les hypothèses relatives à l'environnement prévu.

AVA_MSU.3.4C Les guides doivent énumérer toutes les exigences pour les mesures de sécurité externes (comprenant les contrôles externes organisationnels, physiques et touchant au personnel).

AVA_MSU.3.5C L'analyse de la documentation doit démontrer que les guides sont complets.

Tâches de l'évaluateur :

AVA_MSU.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_MSU.3.2E L'évaluateur doit appliquer à nouveau toutes les procédures de configuration, d'installation et sélectivement d'autres procédures, afin de confirmer que la TOE peut être configurée et utilisée de manière sûre en utilisant seulement les guides fournis.

AVA_MSU.3.3E L'évaluateur doit déterminer si l'utilisation des guides permet de détecter tous les états non sûrs.

AVA_MSU.3.4E **L'évaluateur doit confirmer que la documentation d'analyse montre que sont donnés des conseils pour une exploitation sûre de la TOE dans tous les modes d'exploitation.**

AVA_MSU.3.5E **L'évaluateur doit effectuer des tests indépendants afin de déterminer si un administrateur ou un utilisateur, ayant acquis une bonne compréhension des guides, serait raisonnablement capable de déterminer si la TOE est configurée et exploitée d'une manière non sûre.**

14.3 Résistance des fonctions de sécurité de la TOE (AVA_SOF)

Objectifs

495 Même si une fonction de sécurité de la TOE ne peut pas être court-circuitée, désactivée ou altérée, il est encore possible de la mettre en échec parce qu'une vulnérabilité s'est introduite dans le concept de ses mécanismes de sécurité sous-jacents. Pour ces fonctions, une qualification de leur comportement de sécurité peut être faite en utilisant les résultats d'une analyse quantitative ou statistique du comportement de sécurité de tels mécanismes et de l'effort requis pour les mettre en échec. Cette qualification se fait sous la forme d'une annonce de la résistance des fonctions de sécurité de la TOE.

Classement des composants

496 Il n'y a qu'un seul composant dans la présente famille.

Notes d'application

497 Les fonctions de sécurité sont implémentées par des mécanismes de sécurité. Par exemple, un mécanisme de mot de passe peut être utilisé dans l'implémentation d'une fonction de sécurité d'identification et d'authentification.

498 L'évaluation de la résistance des fonctions de sécurité de la TOE est effectuée au niveau du mécanisme de sécurité, mais ses résultats fournissent des informations sur la capacité de la fonction de sécurité correspondante à contrer les menaces identifiées.

499 L'analyse de la résistance des fonctions de sécurité de la TOE devrait au moins prendre en compte le contenu de toutes les fournitures de la TOE, y compris la ST, pour le niveau d'assurance de l'évaluation visé.

AVA_SOF.1 Évaluation de la résistance des fonctions de sécurité de la TOE

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_HLD.1 Conception de haut niveau descriptive

Tâches du développeur :

AVA_SOF.1.1D **Le développeur doit effectuer une analyse de la résistance des fonctions de sécurité de la TOE pour chaque mécanisme identifié dans la ST faisant l'objet d'une annonce de résistance des fonctions de sécurité de la TOE.**

Contenu et présentation des éléments de preuve :

AVA_SOF.1.1C **Pour chaque mécanisme faisant l'objet d'une annonce de résistance des fonctions de sécurité de la TOE, l'analyse de la résistance des fonctions de**

sécurité de la TOE doit montrer qu'il atteint ou dépasse le niveau de résistance minimum défini dans le PP ou la ST.

AVA_SOF.1.2C **Pour chaque mécanisme faisant l'objet d'une annonce spécifique de résistance des fonctions de sécurité de la TOE, l'analyse de la résistance des fonctions de sécurité de la TOE doit montrer qu'il atteint ou dépasse la métrique spécifique de la résistance des fonctions définie dans le PP ou la ST.**

Tâches de l'évaluateur :

AVA_SOF.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AVA_SOF.1.2E **L'évaluateur doit confirmer que les annonces de la résistance sont correctes.**

14.4 Analyse de vulnérabilités (AVA_VLA)

Objectifs

500 L'analyse de vulnérabilités est une estimation visant à déterminer si des vulnérabilités identifiées pendant l'évaluation de la construction et de l'exploitation prévue de la TOE, ou identifiées par d'autres méthodes (e.g. par des hypothèses sur les anomalies), pourraient permettre à des utilisateurs de violer la TSP.

501 L'analyse de vulnérabilités traite des menaces selon lesquelles un utilisateur serait capable de découvrir des anomalies qui lui donneraient la possibilité d'accéder à des ressources (e.g. des données) sans y être autorisé, d'interférer avec la TSF, de l'altérer ou d'interférer avec les privilèges d'autres utilisateurs.

Classement des composants

502 Le classement est basé sur une rigueur croissante de l'analyse de vulnérabilités par le développeur et par l'évaluateur.

Notes d'application

503 Une analyse de vulnérabilités est effectuée par le développeur afin de vérifier la présence de vulnérabilités de sécurité et devrait au moins prendre en compte le contenu de toutes les fournitures de la TOE, y compris la ST, pour le niveau d'assurance de l'évaluation visé. Le développeur doit documenter les caractéristiques des vulnérabilités identifiées afin de permettre à l'évaluateur d'utiliser cette information, si elle se révèle utile, pour appuyer son analyse indépendente de vulnérabilités.

504 Le but de l'analyse du développeur est de confirmer qu'aucune vulnérabilité de sécurité identifiée ne peut être exploitée dans l'environnement prévu pour la TOE et que la TOE résiste aux attaques de pénétration évidentes.

505 On considère que les vulnérabilités évidentes sont celles dont exploitation n'exige qu'un minimum de compréhension de la TOE, de compétences, d'expertise technique et de ressources . Ces dernières caractéristiques pourraient être devinées au vu de la description de l'interface de la TSF. Les vulnérabilités évidentes comprennent celles qui sont dans le domaine public, dont les détails devraient être connus d'un développeur ou bien disponibles auprès d'une autorité d'évaluation.

506 La réalisation d'une analyse de vulnérabilités de façon systématique exige que le développeur identifie les vulnérabilités d'une manière structurée et répétable, par opposition à une manière opportuniste. Les éléments de preuve associés au fait que la recherche de vulnérabilités a été faite de façon systématique devraient inclure l'identification de toute la documentation concernant la TOE sur laquelle la recherche d'anomalies a été basée.

507 L'analyse de vulnérabilités indépendante va au-delà des vulnérabilités identifiées par le développeur. Le principal but de l'analyse de l'évaluateur est de déterminer

si la TOE est résistante aux attaques de pénétration effectuées par un attaquant ayant un potentiel d'attaque élémentaire (pour AVA_VLA.2), moyen (pour AVA_VLA.3) ou élevé (pour AVA_VLA.4). Pour atteindre ce but, l'évaluateur fait d'abord une estimation du caractère exploitable de toutes les vulnérabilités identifiées. Cela est accompli en effectuant des tests de pénétration. L'évaluateur devrait assumer le rôle d'un attaquant doté d'un potentiel d'attaque élémentaire (pour AVA_VLA.2), moyen (pour AVA_VLA.3) ou élevé (pour AVA_VLA.4) quand il essaye de pénétrer la TOE. Toute exploitation des vulnérabilités par un tel attaquant devrait être considérée par l'évaluateur comme des "attaques de pénétration évidentes" (par rapport aux éléments de AVA_VLA.*.2C) dans le contexte des composants AVA_VLA.2 à AVA_VLA.4.

AVA_VLA.1 Analyse de vulnérabilités du développeur

Objectifs

508 Une analyse de vulnérabilités est effectuée par le développeur pour vérifier la présence de vulnérabilités de sécurité évidentes et pour confirmer qu'elles ne peuvent pas être exploitées dans l'environnement prévu pour la TOE.

Notes d'application

509 L'évaluateur devrait envisager d'effectuer des tests supplémentaires à cause des vulnérabilités potentielles exploitables, identifiées pendant les autres tâches de l'évaluation.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_HLD.1 Conception de haut niveau descriptive

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_VLA.1.1D **Le développeur doit effectuer et documenter une analyse des fournitures de la TOE pour rechercher les moyens évidents par lesquels un utilisateur peut violer la TSP.**

AVA_VLA.1.2D **Le développeur doit documenter les caractéristiques des vulnérabilités évidentes.**

Contenu et présentation des éléments de preuve :

AVA_VLA.1.1C **La documentation doit montrer, pour toutes les vulnérabilités identifiées, que la vulnérabilité ne peut pas être exploitée dans l'environnement prévu pour la TOE.**

Tâches de l'évaluateur :

AVA_VLA.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AVA_VLA.1.2E **L'évaluateur doit effectuer des tests de pénétration, en s'appuyant sur l'analyse de vulnérabilités du développeur, pour garantir que les vulnérabilités évidentes ont été prises en compte.**

AVA_VLA.2 Analyse de vulnérabilités indépendante

Objectifs

510 Une analyse de vulnérabilités est effectuée par le développeur pour vérifier la présence de vulnérabilités de sécurité et pour confirmer qu'elles ne peuvent pas être exploitées dans l'environnement prévu pour la TOE.

511 L'évaluateur effectue des tests de pénétration indépendants, en s'appuyant sur une analyse de vulnérabilités indépendante, afin de déterminer si la TOE est résistante aux attaques de pénétration effectuées par des attaquants ayant un potentiel d'attaque élémentaire.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_VLA.2.1D Le développeur doit effectuer et documenter une analyse des fournitures de la TOE pour rechercher les **moyens** par lesquels un utilisateur peut violer la TSP.

AVA_VLA.2.2D Le développeur doit documenter les caractéristiques des **vulnérabilités identifiées**.

Contenu et présentation des éléments de preuve :

AVA_VLA.2.1C La documentation doit montrer, pour toutes les vulnérabilités identifiées, que la vulnérabilité ne peut pas être exploitée dans l'environnement prévu pour la TOE.

AVA_VLA.2.2C **La documentation doit justifier que la TOE, compte tenu des vulnérabilités identifiées, est résistante aux attaques de pénétration évidentes**.

Tâches de l'évaluateur :

AVA_VLA.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_VLA.2.2E L'évaluateur doit effectuer des tests de pénétration, en s'appuyant sur l'analyse de vulnérabilités du développeur, pour garantir que les vulnérabilités **identifiées** ont été prises en compte.

AVA_VLA.2.3E **L'évaluateur doit effectuer une analyse de vulnérabilités indépendante.**

AVA_VLA.2.4E **L'évaluateur doit effectuer des tests de pénétration indépendants, basés sur une analyse de vulnérabilités indépendante, afin de déterminer le caractère exploitable des vulnérabilités supplémentaires identifiées dans l'environnement prévu.**

AVA_VLA.2.5E **L'évaluateur doit déterminer si la TOE est résistante aux attaques de pénétration effectuées par un attaquant ayant un potentiel d'attaque élémentaire.**

AVA_VLA.3 Résistance moyenne

Objectifs

512 Une analyse de vulnérabilités est effectuée par le développeur pour vérifier la présence de vulnérabilités de sécurité et pour confirmer qu'elles ne peuvent pas être exploitées dans l'environnement prévu pour la TOE.

513 L'évaluateur effectue des tests de pénétration indépendants, en s'appuyant sur une analyse de vulnérabilités indépendante, afin de déterminer si la TOE est résistante aux attaques de pénétration effectuées par des attaquants ayant un potentiel d'attaque moyen.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_VLA.3.1D **Le développeur doit effectuer et documenter une analyse des fournitures de la TOE pour rechercher les moyens par lesquels un utilisateur peut violer la TSP.**

AVA_VLA.3.2D **Le développeur doit documenter les caractéristiques des vulnérabilités identifiées.**

Contenu et présentation des éléments de preuve :

AVA_VLA.3.1C La documentation doit montrer, pour toutes les vulnérabilités identifiées, que la vulnérabilité ne peut pas être exploitée dans l'environnement prévu pour la TOE.

AVA_VLA.3.2C La documentation doit justifier que la TOE, compte tenu des vulnérabilités identifiées, est résistante aux attaques de pénétration évidentes.

AVA_VLA.3.3C **Les éléments de preuve doivent montrer que la recherche de vulnérabilités est systématique.**

Tâches de l'évaluateur :

AVA_VLA.3.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_VLA.3.2E **L'évaluateur doit effectuer des tests de pénétration, en s'appuyant sur l'analyse de vulnérabilités du développeur, pour garantir que les vulnérabilités identifiées ont été prises en compte.**

AVA_VLA.3.3E L'évaluateur doit effectuer une analyse de vulnérabilités indépendante.

AVA_VLA.3.4E **L'évaluateur doit effectuer des tests de pénétration indépendants, basés sur une analyse de vulnérabilités indépendante, afin de déterminer le caractère exploitable des vulnérabilités supplémentaires identifiées dans l'environnement prévu.**

AVA_VLA.3.5E L'évaluateur doit déterminer si la TOE est résistante aux attaques de pénétration effectuées par un attaquant ayant un potentiel d'attaque **moyen**.

AVA_VLA.4 Résistance élevée

Objectifs

514 Une analyse de vulnérabilités est effectuée par le développeur pour vérifier la présence de vulnérabilités de sécurité et pour confirmer qu'elles ne peuvent pas être exploitées dans l'environnement prévu pour la TOE.

515 L'évaluateur effectue des tests de pénétration indépendants, en s'appuyant sur une analyse de vulnérabilités indépendante, afin de déterminer si la TOE est résistante aux attaques de pénétration effectuées par des attaquants ayant un potentiel d'attaque élevé.

Dépendances :

ADV_FSP.1 Spécifications fonctionnelles informelles

ADV_HLD.2 Conception de haut niveau - identification des sous-sytèmes dédiés à la sécurité

ADV_IMP.1 Sous-ensemble de l'implémentation de la TSF

ADV_LLD.1 Conception de bas niveau descriptive

AGD_ADM.1 Guide de l'administrateur

AGD_USR.1 Guide de l'utilisateur

Tâches du développeur :

AVA_VLA.4.1D Le développeur doit effectuer et documenter une analyse des fournitures de la TOE pour rechercher les moyens par lesquels un utilisateur peut violer la TSP.

AVA_VLA.4.2D Le développeur doit documenter les caractéristiques des vulnérabilités identifiées.

Contenu et présentation des éléments de preuve :

AVA_VLA.4.1C La documentation doit montrer, pour toutes les vulnérabilités identifiées, que la vulnérabilité ne peut pas être exploitée dans l'environnement prévu pour la TOE.

AVA_VLA.4.2C La documentation doit justifier que la TOE, compte tenu des vulnérabilités identifiées, est résistante aux attaques de pénétration évidentes.

AVA_VLA.4.3C Les éléments de preuve doivent montrer que la recherche de vulnérabilités est systématique.

AVA_VLA.4.4C **La documentation d'analyse doit fournir une justification selon laquelle l'analyse prend complètement en compte les fournitures de la TOE.**

Tâches de l'évaluateur :

AVA_VLA.4.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AVA_VLA.4.2E L'évaluateur doit effectuer des tests de pénétration, en s'appuyant sur l'analyse de vulnérabilités du développeur, pour garantir que les vulnérabilités identifiées ont été prises en compte.

AVA_VLA.4.3E L'évaluateur doit effectuer une analyse de vulnérabilités indépendante.

AVA_VLA.4.4E L'évaluateur doit effectuer des tests de pénétration indépendants, basés sur une analyse de vulnérabilités indépendante, afin de déterminer le caractère exploitable des vulnérabilités supplémentaires identifiées dans l'environnement prévu.

AVA_VLA.4.5E L'évaluateur doit déterminer si la TOE est résistante aux attaques de pénétration effectuées par un attaquant ayant un potentiel d'attaque **élevé**.

15 Paradigme de la maintenance de l'assurance

15.1 Introduction

516 Le présent chapitre propose l'exposé d'un paradigme de la maintenance de l'assurance qui est supporté par la classe "Maintenance de l'assurance" (AMA). En tant que tel, il fournit des informations utiles pour comprendre une approche possible pour appliquer les exigences de la classe AMA.

517 La maintenance de l'assurance est un concept destiné a être appliqué après l'évaluation et la certification d'une TOE par rapport aux critères des chapitres 4 à 5 et 8 à 14. Les exigences de la maintenance de l'assurance visent à garantir que la TOE continuera à satisfaire à sa cible de sécurité alors que des changements auront été apportés à la TOE ou à son environnement. De tels changements incluent la découverte de menaces ou de vulnérabilités nouvelles, les changements dans les exigences de l'utilisateur, la correction des bogues trouvés dans la TOE qui a été certifiée, et les autres mises à jour des fonctionnalités fournies.

518 Une manière de déterminer que l'assurance a été maintenue est de faire une réévaluation de la TOE. Le terme 'réévaluation' fait ici référence à l'évaluation d'une nouvelle version de la TOE qui couvre tous les changements touchant à la sécurité appliqués à la version certifiée de la TOE et qui réutilise les résultats de l'évaluation précédente quand ils sont encore valides. Cependant, dans de nombreux cas, il est peu vraisemblable de réaliser de façon pratique une réévaluation de chaque nouvelle version de la TOE dans le but de garantir que l'assurance continue à être maintenue.

519 Le but principal de la classe AMA est par conséquent de définir un ensemble d'exigences qui peuvent être appliquées pour procurer la confiance que l'assurance établie dans une TOE est maintenue, sans nécessiter à chaque fois une réévaluation formelle des nouvelles versions de la TOE. La classe AMA ne supprime pas entièrement le besoin de réévaluation. Dans certains cas, les changements peuvent être tellement importants que seule une réévaluation peut être assez fiable pour garantir que l'assurance est maintenue. Les exigences de cette classe ont donc comme objectif secondaire d'appuyer la réévaluation d'une TOE avec un bon rapport coût-efficacité quand cela est nécessaire.

520 Il est à noter qu'il est possible de réévaluer toute nouvelle version d'une TOE par rapport aux critères des chapitres 4 à 5 et 8 à 14 sans qu'aucune des exigences de la classe AMA n'ait été satisfaite. Cependant, la classe AMA inclut des exigences qui peuvent être utilisées pour aider à effectuer une telle réévaluation.

521 Les tâches du développeur et de l'évaluateur relatives à la maintenance sont destinées à être appliquées après que la TOE a été évaluée et certifiée bien que, comme cela est décrit ci-dessous, certaines exigences puissent être appliquées au moment de l'évaluation. Dans un souci de clarté, les termes suivants sont utilisés dans la description de ce paradigme :

a) la *version certifiée* de la TOE fait référence à la version qui a été évaluée et certifiée ;

b) la *version courante* de la TOE fait référence à une version qui est différente à certains égards de la version certifiée ; elle pourrait être, par exemple :

- une nouvelle version de la TOE,
- la version certifiée incluant des corrections (patches) pour corriger des bogues qui ont été découverts ultérieurement,
- la même version de base de la TOE, mais fonctionnant sur une plate-forme matérielle ou logicielle différente.

522 Les rôles du développeur et de l'évaluateur dans la présente classe sont ceux décrits dans la partie 1 des CC. Cependant, il n'est pas nécessaire que l'évaluateur auquel il est fait référence dans les exigences de cette classe soit le même que celui qui a évalué la version certifiée de la TOE.

523 Afin de faire en sorte que l'assurance soit maintenue dans une TOE sans nécessiter systématiquement une réévaluation formelle, les exigences de la présente classe expriment l'obligation pour le développeur de maintenir les éléments de preuve montrant que la TOE continue à satisfaire à sa cible de sécurité (e.g. éléments de preuve relatifs aux tests du développeur).

15.2 Cycle de maintenance de l'assurance

524 La présente section décrit une approche possible pour l'utilisation des familles et des composants relatifs à la maintenance de l'assurance, destinée à illustrer l'utilisation des concepts. L'exemple prend comme modèle un 'cycle de maintenance de l'assurance' qui peut être décomposé selon les trois phases suivantes :

a) la *phase d'acceptation*, au début d'un cycle, dans laquelle les plans et les procédures du développeur pour la maintenance de l'assurance pendant le cycle sont établis par le développeur et validés de manière indépendante par un évaluateur ;

b) la *phase de surveillance*, dans laquelle le développeur fournit à un ou plusieurs moments du cycle des éléments de preuve que l'assurance est maintenue dans la TOE en conformité avec les plans et les procédures établies, ces éléments de preuve de la maintenance de l'assurance étant vérifiés de manière indépendante par un évaluateur ;

c) la *phase de réévaluation*, qui termine le cycle, dans laquelle une version à jour de la TOE est soumise à une réévaluation basée sur les changements qui ont affecté la version certifiée de la TOE.

525 Les familles de la classe AMA traitent principalement les deux premières de ces phases, tout en apportant une aide pour la troisième. Ces phases sont introduites ici simplement pour aider à décrire l'application des exigences pour la maintenance de

l'assurance. Il n'y a aucune intention d'imposer un schéma de maintenance de l'assurance qui incorpore formellement ces phases.

526 Le cycle de maintenance de l'assurance est représenté dans la figure 15.1 ci-dessous.

527 Dans cet exemple, une TOE peut passer en phase de surveillance seulement quand la phase d'acceptation s'est terminée avec succès (i.e. que les plans et les procédures du développeur pour la maintenance de l'assurance ont été acceptés). Si le développeur effectue des changements à ces plans ou procédures pendant la phase de surveillance, alors la TOE devra entrer à nouveau en phase d'acceptation pour que les changements soient acceptés.

528 Pendant la phase de surveillance, le développeur suit les plans et procédures pour la maintenance de l'assurance, en effectuant une analyse de l'impact sur la sécurité des changements affectant la TOE (analyse de l'impact sur la sécurité). À certains moments de cette phase, un évaluateur vérifie de manière indépendante (au moyen d'un audit) le travail du développeur. Le développeur doit garantir que les plans et les procédures sont suivis et que l'analyse de l'impact sur la sécurité est correctement réalisée.

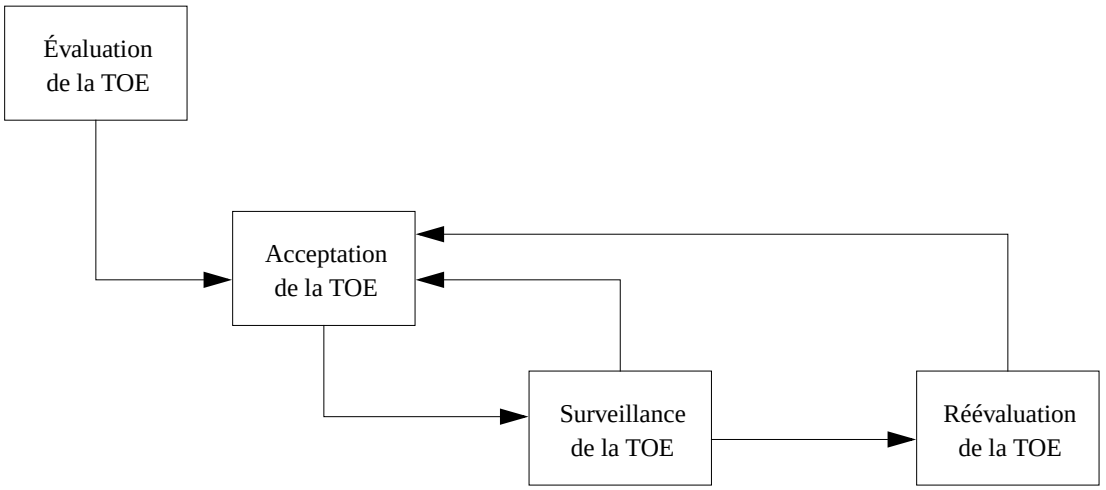

Figure 15.1 - Exemple de cycle de maintenance de l'assurance

529 Par conséquent, une fois qu'une TOE est en phase de surveillance, il devient possible d'acquérir la confiance que l'assurance a été maintenue dans la TOE pour les nouvelles versions de la TOE produites par le développeur.

530 Une TOE qui fait l'objet de changements ne resterait pas indéfiniment en phase de surveillance : à un certain moment, une réévaluation de la TOE deviendrait

nécessaire. La décision de savoir quand une réévaluation serait exigée dépend des changements cumulés, ainsi que des changements vraiment importants, qui sont appliqués à la TOE : par exemple, un grand nombre de changements mineurs pourrait avoir un impact sur l’assurance équivalent à celui d’un changement majeur. Le plan de maintenance de l’assurance du développeur définit le champ d’application des changements qui peuvent être appliqués à la TOE pendant la phase de surveillance (voir la section 15.3.1 ci-après).

531 De la même façon, il ne serait pas possible de ‘surcoter’ une TOE (i.e. augmenter le niveau d’assurance) pendant la phase de surveillance : cela ne pourrait seulement être accompli qu’au moyen d’une évaluation de la TOE (en faisant une utilisation appropriée des résultats d’évaluation précédents).

532 L’état dans lequel se trouve la maintenance de l’assurance de la TOE devra être revu si l’on découvre que les procédures de maintenance de l’assurance ne sont pas suivies et, qu’en conséquence, l’assurance dans la TOE est amoindrie. Dans certains cas, on peut exiger du développeur qu’il soumette la TOE à une réévaluation et par la suite qu’il démarre un nouveau cycle de maintenance de l’assurance.

15.2.1 Acceptation de la TOE

533 Dans l’exemple, la phase d’acceptation de la TOE dans le cycle de maintenance de l’assurance peut être raffinée comme suit, en utilisant les familles “Plan de maintenance de l’assurance” et “Rapport de classification des composants de la TOE”, tirées de la classe AMA.

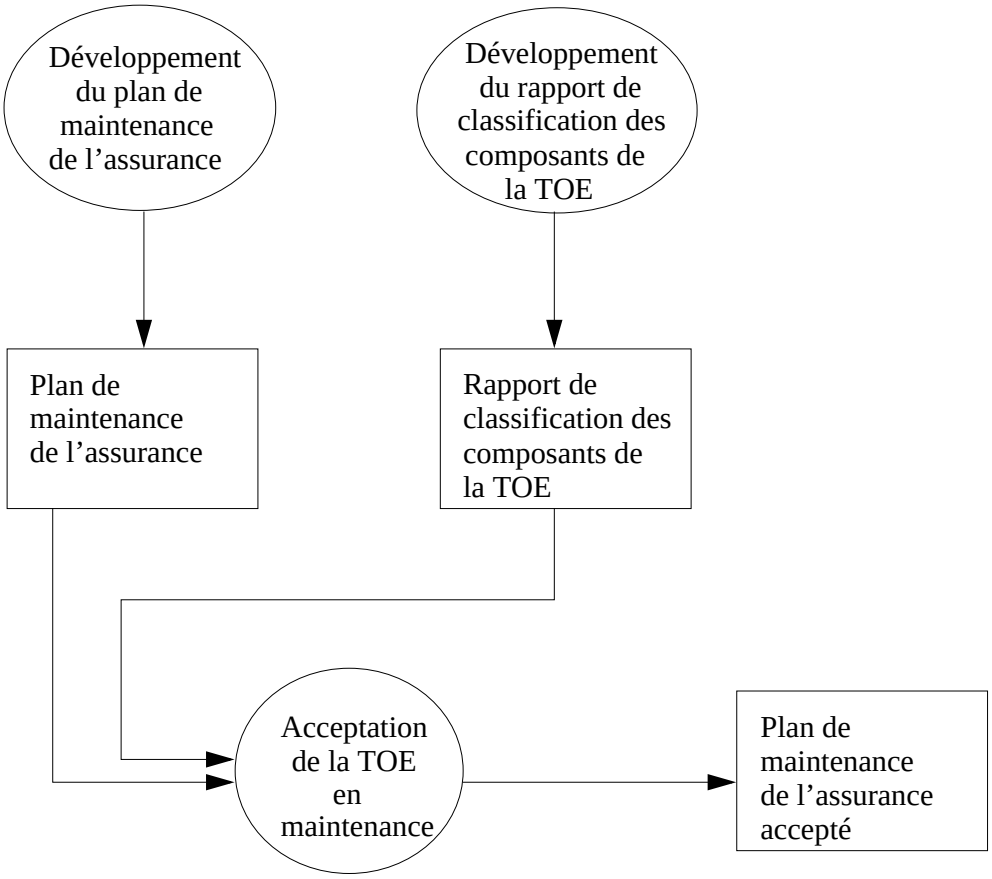

Figure 15.2 - Exemple d’approche d’acceptation d’une TOE

15.2.2 Surveillance de la TOE

534 La phase de surveillance de la TOE du cycle de maintenance de l’assurance serait raffinée comme suit, en utilisant les familles “Éléments de preuve de la

maintenance de l’assurance” et “Analyse de l’impact sur la sécurité”, de la classe AMA.

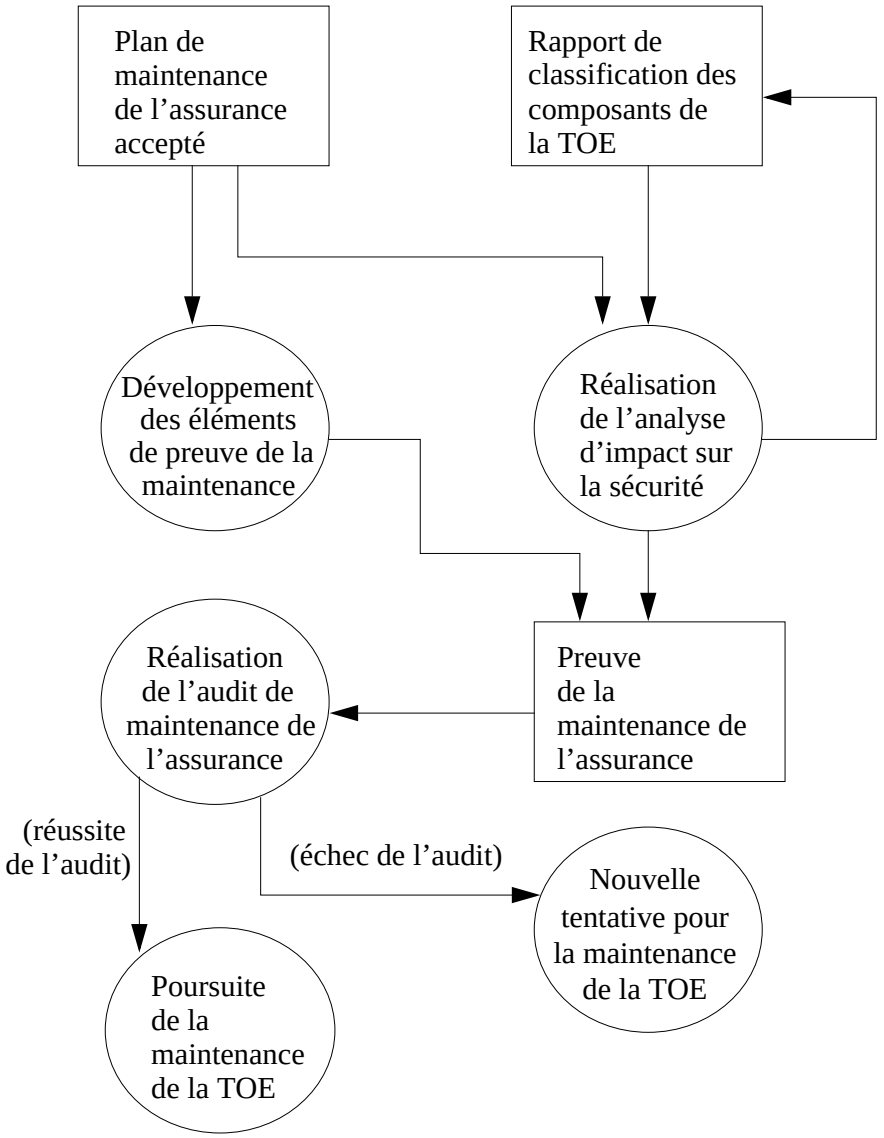

Figure 15.3 - Exemple d’approche de surveillance de la TOE

15.2.3 Réévaluation

535 La troisième phase de cet exemple relatif au cycle de maintenance est celle de réévaluation, dans laquelle l’évaluateur se sert de l’analyse d’impact et des éléments de preuve de maintenance de l’assurance pour réexaminer des parties de la TOE en utilisant les composants d’assurance applicables pour le niveau d’assurance visé.

536 Les activités relatives à la réévaluation pourraient être planifiées dans le plan AM ou exigées en réponse à des changements importants non prévus appliqués à la TOE ou à son environnement, pour lesquels les activités de maintenance de l'assurance ne sont pas considérées comme appropriées.

15.3 Classe et familles relatives à la maintenance de l'assurance

537 Pour appuyer les approches relatives à la maintenance de l'assurance, la classe AMA a été développée et comprend quatre familles comme indiqué dans le tableau 15.1.

Classe d'assurance	Famille d'assurance	Abréviation
Classe AMA : Maintenance de l'assurance	Plan de maintenance de l'assurance	AMA_AMP
	Rapport de classification des composants de la TOE	AMA_CAT
	Preuve de la maintenance de l'assurance	AMA_EVD
	Analyse d'impact sur la sécurité	AMA_SIA

Tableau 15.1 - Décomposition et correspondances de la famille de maintenance de l'assurance

15.3.1 Plan de maintenance de l'assurance

538 Le plan AM fournit une identification claire des éléments de base pour la maintenance de l'assurance, en termes de résultats de l'évaluation et de définition de la classification des composants de la TOE.

539 Le plan de maintenance de l'assurance (Plan AM) identifie les plans et les procédures qu'un développeur met en œuvre pour garantir que l'assurance qui a été établie dans la TOE certifiée est maintenue alors que des changements sont effectués sur la TOE ou dans son environnement. Un plan AM couvre un cycle de maintenance de l'assurance.

540 Le plan AM définit le champ d'application des changements qui peuvent être effectués sur la TOE sans déclencher une réévaluation. L'approche spécifique à suivre dépend du schéma, mais les types de changements suivants sont vraisemblablement en dehors du champ d'application du plan AM et donc ne pourraient être traités qu'au moyen d'une réévaluation :

a) changements importants appliqués à la cible de sécurité (i.e. changements importants dans l'environnement de sécurité, des objectifs de sécurité ou des exigences fonctionnelles de sécurité ou *tout* accroissement des exigences d'assurance) ;

b) changements importants appliqués aux interfaces externes de la TSF considérées comme dédiées à l'application de la TSP ;

c) changements importants apportés aux sous-systèmes de la TSF considérés comme dédiés à l'application de la TSP (dans le cas où les exigences d'assurance incluent ADV_HLD.1 ou des composants plus élevés).

541 Il est à noter que l'approche concernant les changements effectués pendant la maintenance peut être influencée par toutes les fonctions offertes par la TOE qui contribuent à la validation automatique de la sécurité de la configuration qui a été évaluée. De telles fonctions peuvent empêcher que des changements non appropriés ou préjudiciables soient appliqués à une TOE en exploitation.

542 Une spécification plus précise des règles est en dehors du champ d'application des CC, une raison, et non la moindre, étant que la définition de ce qui constitue un changement *important* dépend du type de TOE évaluée et du contenu de la cible de sécurité.

543 Le plan AM est exigé pour définir ou faire référence aux procédures qui seront appliquées pour garantir que l'assurance dans la TOE est maintenue pendant le cycle de maintenance de l'assurance. Quatre types de procédures qui devraient être appliquées, sont identifiés :

a) les procédures de gestion de configuration, qui contrôlent et enregistrent les changements effectués à la TOE pour aider l'analyse de l'impact sur la sécurité faite par le développeur, ainsi que la documentation d'aide (y compris le plan AM lui-même) ;

b) les procédures pour maintenir 'les éléments de preuve de l'assurance' (i.e. la maintenance des éléments de preuve à caractère documentaire comme cela est requis par les exigences d'assurance appropriées), dont un aspect clé est constitué par les tests fonctionnels des fonctions de sécurité de la TOE et, en particulier, par la politique de test de non régression mise en œuvre par le développeur ;

c) les procédures régissant l'analyse de l'impact sur la sécurité des changements qui affectent la TOE (notons que ceci inclut les changements dans l'environnement de la TOE, tels que les menaces ou les méthodes d'attaque nouvelles qui peuvent nécessiter d'être identifiées et suivies), ainsi que la maintenance du rapport de classification des composants de la TOE quand des changements sont effectués ;

d) les procédures de correction d'anomalies, qui couvrent le suivi et la correction des anomalies de sécurité signalées (comme cela est exigé par ALC_FLR.1).

544 Le plan AM est prévu pour rester valide jusqu'à la fin du cycle de maintenance de l'assurance (i.e. jusqu'à l'achèvement de la réévaluation planifiée), après quoi un nouveau plan AM sera exigé. Le plan AM est censé être invalidé si le développeur ne suit pas le plan ou effectue des changements à la TOE qui sont en dehors du champ d'application du plan, ou bien doit effectuer de tels changements afin que la TOE demeure efficace dans son environnement. Un plan AM mis à jour devrait être

à nouveau proposé pour validation et accepté, avant qu'une TOE entre dans une nouvelle phase de surveillance.

545 Le plan AM exige que le développeur désigne un analyste de sécurité du développeur dont la responsabilité est de surveiller le processus de maintenance de l'assurance. Le rôle peut être rempli par plusieurs individus. Une condition préalable essentielle pour remplir ce rôle est que l'analyste de sécurité du développeur soit familier avec la TOE, avec les résultats de l'évaluation et avec les exigences d'assurance applicables. Les exigences ne spécifient pas comment ce niveau de connaissance et d'expérience auront été acquis ; cependant, il est vraisemblable qu'un futur analyste de sécurité du développeur doive subir un certain programme d'apprentissage pour corriger toute lacune dans ses connaissances et son expérience. L'analyste de sécurité du développeur doit avoir une autorité suffisante au sein de l'organisation du développeur pour garantir que les exigences du plan AM et de ses procédures associées seront suivies.

15.3.2 Rapport de classification des composants de la TOE

546 Le but du rapport de classification des composants de la TOE est de compléter le plan AM en fournissant une classification des composants d'une TOE (e.g. des sous-systèmes de la TSF) suivant leur pertinence vis-à-vis de la sécurité. Cette classification est déterminante pour l'analyse de l'impact sur la sécurité faite par le développeur et également pour la réévaluation ultérieure de la TOE.

547 La vérification du rapport de classification des composants de la TOE intervient pendant la phase d'acceptation ; les vérifications de l'évaluateur portent seulement sur la version du rapport relatif à la version certifiée de la TOE. Alors que les procédures de maintenance de l'assurance identifiées dans le plan AM exigent que le développeur mette à jour le rapport de classification des composants de la TOE quand des changements sont effectués à la TOE, les évaluateurs ne sont pas tenus de réviser à nouveau le document ; cependant, de telles mises à jour seront vraisemblablement contrôlées pendant la phase de surveillance.

548 Le rapport de classification des composants de la TOE couvre toutes les représentations de la TSF pour le niveau d'assurance qui est maintenu. Le rapport de classification des composants de la TOE identifie également :

a) tous les composants matériels, logiciels ou micro-programmés qui sont externes à la TOE (e.g. plates-formes matérielles ou logicielles) et qui satisfont aux exigences de sécurité des TI comme cela est défini dans la ST ;

b) tous les outils de développement qui, s'ils sont modifiés, auront un impact sur l'assurance dans le fait que la TOE satisfait à sa ST.

549 Le rapport de classification des composants de la TOE fournit également une description de l'approche utilisée pour la classification des composants de la TOE. Au minimum, les composants de la TOE doivent être classés comme étant soit dédiés à l'application de la TSP soit non dédiés à l'application de la TSP. La description du schéma de classification est destinée à permettre à l'analyste de sécurité du développeur de décider dans quelle catégorie devrait être classé tout

nouveau composant de la TOE, et également du moment où changer de catégorie un composant existant de la TOE, à la suite de changements effectués à la TOE ou à sa ST.

550 La classification initiale des composants de la TOE sera basée sur les éléments de preuve fournis par le développeur pour appuyer l'évaluation de la TOE, validés de manière indépendante par les évaluateurs. Bien que la maintenance du document soit de la responsabilité de l'analyste de sécurité du développeur, son contenu initial peut être basé sur les résultats de l'évaluation de la TOE.

551 Il peut être utile que la ST inclue AMA_CAT.1 quand il y a une exigence que l'assurance soit maintenue pour des versions futures de la TOE. Ceci s'applique, malgré le fait que la maintenance de l'assurance doive être obtenue par l'application des exigences de la présente classe ou par des réévaluations périodiques de la TOE.

15.3.3 Éléments de preuve de la maintenance de l'assurance

552 La confiance que l'assurance dans la TOE est maintenue par le développeur doit être établie, en conformité avec le plan AM. Cela doit être obtenu par la fourniture d'éléments de preuve démontrant que l'assurance dans la TOE a été maintenue, ce qui est vérifié de manière indépendante par un évaluateur. Cette vérification (intitulée 'audit AM') serait effectuée typiquement de façon périodique pendant la phase de surveillance du cycle de maintenance de l'assurance de la TOE.

553 Les audits AM sont effectués en conformité avec la planification définie dans le plan AM. Les tâches du développeur et de l'évaluateur exigées par AMA_EVD.1 seront par conséquent appelées à être effectuées une ou plusieurs fois pendant la phase de surveillance du cycle de maintenance de l'assurance. Les évaluateurs peuvent avoir besoin de visiter l'environnement de développement de la TOE pour examiner les éléments de preuve exigés, mais il n'est pas interdit d'utiliser d'autres moyens pour mener les vérifications.

554 Le développeur doit fournir les éléments de preuve que les procédures de maintenance de l'assurance auxquelles il est fait référence dans le plan AM sont suivies. Cela doit inclure :

a) les enregistrements de gestion de configuration ;

b) la documentation référencée dans l'analyse de l'impact sur la sécurité, comprenant la version courante du rapport de classification des composants de la TOE et les éléments de preuve relatifs à toutes les exigences d'assurance applicables, telles les mises à jour de la conception, la documentation de test, les nouvelles versions de guides, et ainsi de suite ;

c) les éléments de preuve concernant la surveillance des anomalies de sécurité.

555 Les vérifications faites par l'évaluateur de l'analyse de l'impact sur la sécurité effectuée par le développeur (exigée par AMA_SIA.1 duquel dépend AMA_EVD.1) permettront de cibler l'audit AM. L'audit AM permettra, à son tour, de corroborer l'analyse du développeur (et donc la confiance dans la qualité de

l'analyse), servant par là même à valider l'annonce du développeur que l'assurance a été maintenue dans la version courante de la TOE.

556 Un audit AM exige des évaluateurs qu'ils confirment que les tests fonctionnels ont été effectués sur la version courante de la TOE. Ceci est mis en évidence en tant que vérification séparée parce que la documentation de test fournit des éléments de preuve solides que les fonctions de sécurité de la TOE continuent à fonctionner conformément à leurs spécifications. Les évaluateurs confirment, en effectuant un échantillonnage de la documentation de test, que les tests du développeur montrent bien que les fonctions de sécurité fonctionnent conformément à leurs spécifications et que la couverture ainsi que la profondeur des tests sont homogènes avec le niveau d'assurance qui est maintenu.

15.3.4 Analyse de l'impact sur la sécurité

557 Le but de l'analyse de l'impact sur la sécurité est de procurer la confiance que l'assurance a été maintenue dans la TOE, au moyen d'une analyse, effectuée par le développeur, de l'impact sur la sécurité de tous les changements affectant la TOE après qu'elle ait été certifiée. Ces exigences peuvent être appliquées pendant une phase de surveillance ou une phase de réévaluation.

558 L'analyse de l'impact sur la sécurité effectuée par le développeur est basée sur le rapport de classification des composants de la TOE : les changements apportés aux composants de la TOE dédiés à l'application de la TSP peuvent avoir un impact sur l'assurance dans le fait que la TOE continue à satisfaire à sa ST à la suite des changements. Tous ces changements exigent par conséquent une analyse de leur impact sur la sécurité afin de montrer qu'ils ne diminuent pas l'assurance dans la TOE.

559 Les composants de la présente famille peuvent être utilisés en appui soit d'un audit AM ultérieur, soit d'une réévaluation de la TOE.

560 Dans le cas d'un audit AM, la revue de l'analyse de l'impact sur la sécurité faite par les évaluateurs devrait permettre de cibler les activités d'audit ultérieures, qui devraient à leur tour corroborer l'analyse du développeur.

561 L'analyse de l'impact sur la sécurité identifie les changements effectués sur la version certifiée de la TOE, en indiquant soit les composants de la TOE qui sont nouveaux, soit ceux qui ont été modifiés. Les évaluateurs vérifient l'exactitude de ces informations soit pendant l'audit AM associé, soit pendant la réévaluation de la TOE associée.

562 La fourniture de l'analyse de l'impact sur la sécurité en appui d'une réévaluation devrait réduire le niveau d'effort déployé par l'évaluateur, nécessaire pour établir le niveau d'assurance exigé pour la TOE. L'application de AMA_SIA.2, qui nécessite un examen complet de l'analyse de l'impact sur la sécurité, devrait probablement bénéficier au maximum à la réévaluation. Les conditions précises et détaillées dans lesquelles une autorité d'évaluation pourrait souhaiter utiliser en pratique l'analyse de l'impact sur la sécurité dans une réévaluation sont en dehors du champ d'application des CC.

16 Classe AMA : Maintenance de l’assurance

563 La classe “Maintenance de l’assurance” fournit des exigences destinées à être appliquées après qu’une TOE ait été certifiée selon les CC. Ces exigences ont pour but de garantir que la TOE continuera à satisfaire à sa cible de sécurité alors que des changements ont été apportés à la TOE ou à son environnement. De tels changements incluent la découverte de nouvelles menaces ou vulnérabilités, des changements dans les exigences de l’utilisateur et la correction de bogues découverts dans la TOE certifiée.

564 La classe comprend quatre familles et la figure 16.1 présente la hiérarchie des composants au sein de ces familles :

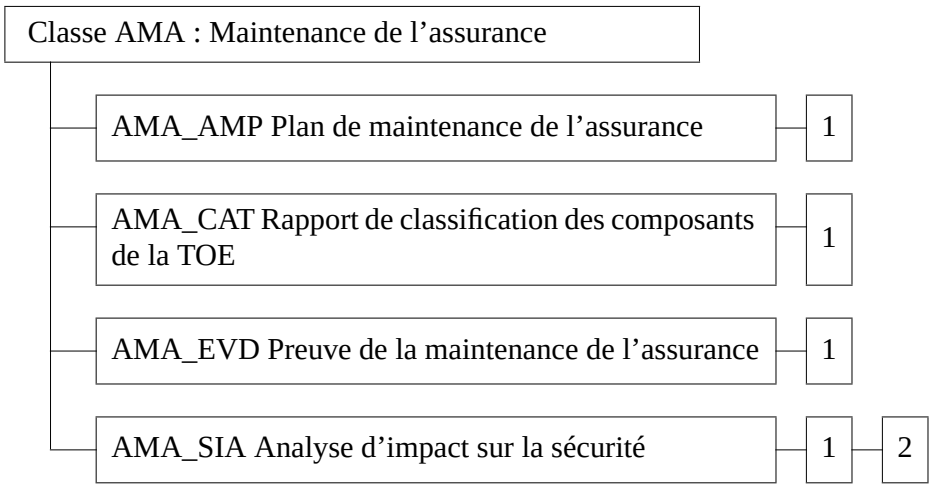

Figure 16.1 - Décomposition de la classe “Maintenance de l’assurance”

16.1 Plan de maintenance de l'assurance (AMA_AMP)

Objectifs

565 Le plan de maintenance de l'assurance (plan AM) identifie les plans et les procédures qu'un développeur doit mettre en œuvre afin de garantir que l'assurance qui a été établie pour la TOE certifiée est maintenue, alors que des changements sont effectués dans la TOE ou dans son environnement. Le plan AM est spécifique à la TOE et est adapté aux pratiques et procédures propres au développeur.

Classement des composants

566 Cette famille ne contient qu'un composant.

Notes d'application

567 Un plan AM couvre un cycle de maintenance de l'assurance, c'est-à-dire la période qui va de l'achèvement de l'évaluation la plus récente de la TOE jusqu'à l'achèvement de la réévaluation planifiée suivante.

568 Les exigences AMA_AMP.1.2C et AMA_AMP.1.3C servent à fournir une identification claire des bases pour la maintenance de l'assurance, en termes de résultats d'évaluation et de définition de la classification des composants de la TOE. Le rapport de classification des composants de la TOE est soumis aux exigences de la famille AMA_CAT et fournit les bases pour l'analyse d'impact sur la sécurité réalisée par l'analyste de sécurité du développeur.

569 La définition du domaine d'application des changements couverts par le plan, telle qu'elle est exigée par AMA_AMP.1.4C, devrait être faite en indiquant les catégories de composants de la TOE qui peuvent être changés et le niveau de représentation auquel les changements peuvent être effectués (en se référant au rapport de classification des composants de la TOE lorsque cela est approprié).

570 AMA_AMP.1.5C exige une description des plans du développeur *en vigueur* pour toute nouvelle version de la TOE. Ces plans peuvent faire l'objet de changements et ainsi nécessiter une mise à jour du plan AM. Il convient de noter cependant que, dans ce contexte, le terme *nouvelle version* n'inclut pas, par exemple, des modifications mineures ('non planifiées') qui intègrent la correction de bogues.

571 AMA_AMP.1.6C exige une définition du planning prévisionnel des audits AM (voir ci-après la famille AMA_EVD) et des réévaluations de la TOE prévues, associée à la justification des plannings proposés. Les plannings peuvent être définis en termes d'échéances (e.g. audits AM annuels) ou ils peuvent être liés à des nouvelles versions spécifiques de la TOE. Les plannings prévisionnels devraient prendre en compte les changements de la TOE attendus durant la période considérée ainsi qu'entre toutes les échéances comprises entre l'évaluation de la TOE et l'établissement du plan AM. En particulier, tous les changements non prévus par le plan AM entraîneront une réévaluation.

AMA_AMP.1 Plan de maintenance de l'assurance

Dépendances :

ACM_CAP.2 Éléments de configuration

ALC_FLR.1 Correction d'anomalies élémentaire

AMA_CAT.1 Rapport de classification des composants de la TOE

Tâches du développeur :

AMA_AMP.1.1D **Le développeur doit fournir un plan AM**.

Contenu et présentation des éléments de preuve :

AMA_AMP.1.1C **Le plan AM doit contenir une brève description de la TOE, incluant les fonctionnalités de sécurité qu'elle offre, ou y faire référence.**

AMA_AMP.1.2C **Le plan AM doit identifier la version certifiée de la TOE et doit faire référence aux résultats d'évaluation.**

AMA_AMP.1.3C **Le plan AM doit faire référence au rapport de classification des composants de la TOE pour la version certifiée de la TOE.**

AMA_AMP.1.4C **Le plan AM doit définir le domaine d'application des changements de la TOE qu'il couvre.**

AMA_AMP.1.5C **Le plan AM doit décrire le cycle de vie de la TOE et doit identifier les prévisions de toute nouvelle version de la TOE, ainsi qu'une brève description de tous les changements planifiés qui auront vraisemblablement un impact significatif sur la sécurité.**

AMA_AMP.1.6C **Le plan AM doit décrire le cycle de maintenance de l'assurance, en établissant et en justifiant le planning prévisionnel des audits AM et la date prévue pour la prochaine réévaluation de la TOE.**

AMA_AMP.1.7C **Le plan AM doit identifier le ou les individus qui assureront le rôle d'analyste de sécurité du développeur pour la TOE.**

AMA_AMP.1.8C **Le plan AM doit décrire comment le rôle d'analyste de sécurité du développeur garantira que les procédures documentées ou référencées dans le plan AM sont suivies.**

AMA_AMP.1.9C **Le plan AM doit décrire comment le rôle d'analyste de sécurité du développeur garantira que toutes les tâches du développeur liées à l'analyse d'impact sur la sécurité des changements affectant la TOE sont réalisées correctement.**

AMA_AMP.1.10C **Le plan AM doit justifier pourquoi le ou les analyste(s) de sécurité du développeur sont considérés comme étant suffisamment familiarisés avec la cible de sécurité, les spécifications fonctionnelles et (lorsque cela est approprié)**

la conception de haut niveau de la TOE, ainsi qu'avec les résultats d'évaluation et toutes les exigences d'assurance applicables pour la version certifiée de la TOE.

AMA_AMP.1.11C **Le plan AM doit décrire les procédures qui doivent être appliquées pour maintenir l'assurance de la TOE ou y faire référence, ce qui doit inclure au minimum les procédures de gestion de configuration, les éléments de preuve relatifs à la maintenance de l'assurance, la réalisation de l'analyse d'impact sur la sécurité des changements opérés sur la TOE et la correction d'anomalies.**

Tâches de l'évaluateur :

AMA_AMP.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AMA_AMP.1.2E **L'évaluateur doit confirmer que les plannings proposés pour les audits AM et la réévaluation de la TOE sont acceptables et cohérents vis-à-vis des changements proposés dans la TOE.**

16.2 Rapport de classification des composants de la TOE (AMA_CAT)

Objectifs

572 L'objectif du rapport de classification des composants de la TOE est de compléter le plan AM en proposant une classification des composants d'une TOE (e.g. les sous-systèmes de la TSF) en fonction de leur pertinence vis-à-vis de la sécurité. Cette classification est déterminante pour l'analyse d'impact sur la sécurité faite par le développeur ainsi que pour la prochaine réévaluation de la TOE.

Classement des composants

573 Cette famille ne contient qu'un composant.

Notes d'application

574 L'expression "niveau de représentation de la TSF le moins abstrait" utilisée dans AMA_CAT.1.1 se réfère au niveau de représentation de la TSF le moins abstrait qui a été fourni pour le niveau d'assurance en cours de maintenance. Par exemple, si la TOE doit être maintenue au niveau d'assurance EAL3, alors le niveau de représentation de la TSF le moins abstrait est la conception de haut niveau, et les composants ci-après de la TOE doivent faire l'objet d'une classification :

a) toutes les interfaces externes de la TSF identifiables dans les spécifications fonctionnelles ;

b) tous les sous-systèmes de la TSF identifiables dans la conception de haut niveau.

575 Alors que AMA_CAT exige qu'au moins deux catégories soient définies, il peut être approprié (en fonction du type de TOE) de continuer à décomposer la catégorie dédiée à l'application de la TSP afin de contribuer à cibler l'analyse d'impact sur la sécurité faite par le développeur. Par exemple, les composants dédiés à l'application de la TSP pourraient être classés soit dans la catégorie *critiques pour la sécurité* soit dans la catégorie *touchant à la sécurité*, à savoir :

a) les composants de la TOE critiques pour la sécurité sont ceux qui sont *directement* responsables de la mise en œuvre d'au moins une fonction de sécurité TI définie dans la cible de sécurité ;

b) les composants de la TOE contribuant à la sécurité sont ceux qui ne sont pas *directement* responsables de la mise en œuvre d'une fonction de sécurité TI (et en conséquence qui ne sont pas critiques pour la sécurité) mais sur lesquels s'appuient néanmoins les fonctions de sécurité TI ; cette catégorie peut à son tour inclure deux types distincts de composants d'une TOE :

- ceux qui offrent des services aux composants de la TOE critiques pour la sécurité et donc sur lesquels elle s'appuie pour fonctionner correctement ;
- ceux qui n'offrent pas de tels services mais pour lesquels il faut cependant avoir confiance dans le fait qu'ils ne se comportent pas d'une façon malveillante (i.e. en introduisant une vulnérabilité).

576 AMA_CAT.1.3C exige une identification de tous les outils de développement qui, s'ils sont modifiés, auront un impact sur l'assurance que la TOE satisfait à sa cible de sécurité (e.g. le compilateur utilisé pour créer le code objet).

AMA_CAT.1 Rapport de classification des composants de la TOE

Dépendances :

ACM_CAP.2 Éléments de configuration

Tâches du développeur :

AMA_CAT.1.1D **Le développeur doit fournir un rapport de classification des composants de la TOE pour la version certifiée de la TOE.**

Contenu et présentation des éléments de preuve :

AMA_CAT.1.1C **Le rapport de classification des composants de la TOE doit classer chaque composant de la TOE identifiable dans chaque représentation de la TSF, de la plus abstraite jusqu'à la moins abstraite, en fonction de sa pertinence vis-à-vis de la sécurité ; au minimum, les composants de la TOE doivent être classés, soit comme étant dédiés à l'application de la TSP, soit comme étant non dédié à l'application de la TSP.**

AMA_CAT.1.2C **Le rapport de classification des composants de la TOE doit décrire le schéma de classification utilisé, de telle sorte que l'on puisse déterminer comment classer de nouveaux composants introduits dans la TOE, et également à quel moment modifier le classement des composants qui existent déjà dans la TOE à la suite de changements apportés à celle-ci ou à sa cible de sécurité.**

AMA_CAT.1.3C **Le rapport de classification des composants de la TOE doit identifier tous les outils utilisés dans l'environnement de développement qui, s'ils sont modifiés, auront un impact sur l'assurance dans le fait que la TOE satisfait à sa cible de sécurité.**

Tâches de l'évaluateur :

AMA_CAT.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AMA_CAT.1.2E **L'évaluateur doit confirmer que la classification des composants de la TOE et des outils, ainsi que le schéma de classification utilisé, sont appropriés et cohérents avec les résultats d'évaluation de la version certifiée.**

16.3 Preuve de la maintenance de l'assurance (AMA_EVD)

Objectifs

577 Le but de cette famille d'exigences est d'établir la confiance dans le fait que l'assurance dans la TOE est maintenue par le développeur, en conformité avec le plan AM. Ceci est réalisé au moyen de la fourniture d'éléments de preuve qui démontrent que l'assurance dans la TOE a été maintenue, ce qui est vérifié indépendamment par un évaluateur. Cette vérification, appelée 'audit AM', est réalisée périodiquement au cours de la période d'application du plan AM.

Classement des composants

578 Cette famille ne contient qu'un composant.

Notes d'application

579 Cette famille inclut des exigences de preuve qui sont similaires aux exigences d'assurance définies dans les classes ACM, ATE et AVA. Cependant, l'audit AM n'impose pas aux évaluateurs d'examiner les éléments de preuve de façon aussi approfondie que cela est exigé par les composants de ces classes ; il impose plutôt une approche par échantillonnage pour établir la confiance dans le fait que les procédures de maintenance de l'assurance sont correctement appliquées.

580 Au cours de l'audit AM, l'évaluateur vérifie (par échantillonnage) que la liste de configuration et l'analyse d'impact sur la sécurité sont cohérentes pour la version courante de la TOE, en ce qui concerne l'identification des composants de la TOE qui ont changé par rapport à la version certifiée de la TOE.

581 AMA_EVD.1.3C exige la fourniture d'éléments de preuve que les procédures de maintenance de l'assurance du plan AM sont appliquées. Ceci concerne toutes les procédures mentionnées dans AMA_AMP.1.11C, c'est-à-dire les éléments de preuve de l'application des procédures relatives à la gestion de configuration, les éléments de preuve relatifs à la maintenance de l'assurance, la réalisation de l'analyse d'impact sur la sécurité et la correction d'anomalies.

582 Les éléments de preuve exigés dans AMA_EVD.1.4C incluent la fourniture d'une liste des vulnérabilités identifiées dans la version courante de la TOE. Cet aspect est mis en évidence dans une exigence séparée car il est important de garantir, à un niveau cohérent avec les exigences d'assurance de l'évaluation initiale, que la version courante ne contient pas de faiblesses de sécurité qui sont exploitables dans l'environnement de la TOE. La liste demandée dans AMA_EVD.1.4C devrait inclure les vulnérabilités qui proviennent de :

a) l'analyse du développeur exigée par AVA_VLA.1 ou par un composant hiérarchique (si cela est exigé pour la version certifiée de la TOE) ;

b) toute autre anomalie de sécurité prise en compte par les procédures de correction d'anomalies exigées par ALC_FLR.1 (ou ALC_FLR.2 si cela est exigé pour la version certifiée de la TOE).

583 AMA_EVD.1.5E exige que l'évaluateur confirme que les tests fonctionnels ont été réalisés sur la version courante de la TOE et que la couverture et la profondeur des tests sont homogènes avec le niveau d'assurance maintenu. Cette vérification est réalisée par échantillonnage de la documentation de test de la version courante de la TOE.

AMA_EVD.1 Éléments de preuve du processus de maintenance

Dépendances :

AMA_AMP.1 Plan de maintenance de l'assurance

AMA_SIA.1 Échantillonnage de l'analyse d'impact sur la sécurité

Tâches du développeur :

AMA_EVD.1.1D **L'analyste de sécurité du développeur doit fournir une documentation AM pour la version courante de la TOE**.

Contenu et présentation des éléments de preuve :

AMA_EVD.1.1C **La documentation AM doit inclure une liste de configuration et une liste des vulnérabilités identifiées de la TOE.**

AMA_EVD.1.2C **La liste de configuration doit décrire les éléments de configuration incluant la version courante de la TOE.**

AMA_EVD.1.3C **La documentation AM doit fournir la preuve que les procédures documentées ou référencées dans le plan AM sont appliquées.**

AMA_EVD.1.4C **La liste des vulnérabilités identifiées dans la version courante de la TOE doit montrer, pour chaque vulnérabilité, qu'elle ne peut pas être exploitée dans l'environnement prévu de la TOE.**

Tâches de l'évaluateur :

AMA_EVD.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AMA_EVD.1.2E **L'évaluateur doit confirmer que les procédures documentées ou référencées dans le plan AM sont appliquées.**

AMA_EVD.1.3E **L'évaluateur doit confirmer que l'analyse d'impact sur la sécurité pour la version courante de la TOE est cohérente avec la liste de configuration.**

AMA_EVD.1.4E **L’évaluateur doit confirmer que tous les changements documentés dans l’analyse d’impact sur la sécurité pour la version courante de la TOE s’inscrivent dans le domaine d’application couvert par le plan AM.**

AMA_EVD.1.5E **L’évaluateur doit confirmer que les tests fonctionnels ont été exécutés sur la version courante de la TOE, avec une profondeur homogène avec le niveau d’assurance maintenu.**

16.4 Analyse d'impact sur la sécurité (AMA_SIA)

Objectifs

584 Le but de l'analyse d'impact sur la sécurité est d'apporter la confiance dans le fait que l'assurance a été maintenue pour la TOE, au moyen d'une analyse réalisée par le développeur des impacts sur la sécurité de tous les changements apportés à la TOE depuis qu'elle a été certifiée.

Classement des composants

585 Cette famille est composée de deux composants classés selon la portée de la validation par l'évaluateur de l'analyse d'impact sur la sécurité effectuée par le développeur.

Notes d'application

586 AMA_SIA.1 exige une approche par échantillonnage pour valider l'analyse d'impact sur la sécurité effectuée par le développeur. Dans certains cas, on peut préférer AMA_SIA.2 lorsqu'une approche par échantillonnage n'est pas jugée suffisante pour établir la confiance dans le fait que l'assurance a été maintenue pour la version courante de la TOE, mais qu'une réévaluation formelle n'est pas jugée nécessaire.

587 Les deux composants de cette famille exigent que l'analyse d'impact sur la sécurité identifie tout composant nouveau ou modifié dans la version courante de la TOE (par rapport à la version certifiée). L'exactitude de cette information est vérifiée soit pendant l'audit AM (par échantillonnage) soit pendant la réévaluation associée de la TOE quand la liste de configuration est vérifiée dans le cadre de ACM_CAP.

AMA_SIA.1 Échantillonnage de l'analyse d'impact sur la sécurité

Dépendances :

AMA_CAT.1 Rapport de classification des composants de la TOE

Tâches du développeur :

AMA_SIA.1.1D **L'analyste de sécurité du développeur doit, pour la version courante de la TOE, fournir une analyse d'impact sur la sécurité qui couvre tous les changements effectués par rapport à la version certifiée de la TOE.**

Contenu et présentation des éléments de preuve :

AMA_SIA.1.1C **L'analyse d'impact sur la sécurité doit identifier la version certifiée de la TOE à partir de laquelle a été constituée la version courante de la TOE.**

AMA_SIA.1.2C **L'analyse d'impact sur la sécurité doit identifier tout composant nouveau ou modifié de la TOE, classé comme étant dédié à l'application de la TSP.**

AMA_SIA.1.3C **L'analyse d'impact sur la sécurité doit, pour tout changement concernant la cible de sécurité ou les représentations de la TSF, décrire brièvement ce changement et tous les effets qu'il peut avoir sur les niveaux de représentation moins abstraits.**

AMA_SIA.1.4C **L'analyse d'impact sur la sécurité doit, pour tout changement concernant la cible de sécurité ou les représentations de la TSF, identifier toutes les fonctions de sécurité TI et tous les composants de la TOE classés comme étant dédiés à l'application de la TSP, qui sont concernés par ce changement.**

AMA_SIA.1.5C **L'analyse d'impact sur la sécurité doit, pour tout changement qui implique une modification de la représentation de l'implémentation de la TSF ou de l'environnement TI, identifier les éléments de preuve relatifs à des tests qui montrent, au niveau d'assurance requis, que la TSF continue à être correctement implémentée à la suite de ce changement.**

AMA_SIA.1.6C **L'analyse d'impact sur la sécurité doit, pour chaque exigence d'assurance applicable dans les classes d'assurance "gestion de configuration" (ACM), "support au cycle de vie" (ALC), "livraison et exploitation" (ADO) et "guides" (AGD), identifier toutes les fournitures d'évaluation qui ont changé et fournir une brève description de chaque changement et de son impact sur l'assurance.**

AMA_SIA.1.7C **L'analyse d'impact sur la sécurité doit, pour chaque exigence d'assurance applicable dans la classe d'assurance "estimation des vulnérabilités" (AVA), identifier les fournitures d'évaluation qui ont changé et celles qui n'ont pas changé, et justifier des décisions concernant la mise à jour ou non de la fourniture.**

Tâches de l'évaluateur :

AMA_SIA.1.1E **L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.**

AMA_SIA.1.2E **L'évaluateur doit vérifier, par échantillonnage, que l'analyse d'impact sur la sécurité documente les modifications à un niveau de détail approprié, avec les justifications appropriées qui montrent que l'assurance a été maintenue dans la version courante de la TOE.**

AMA_SIA.2 Examen de l'analyse d'impact sur la sécurité

Dépendances :

AMA_CAT.1 Rapport de classification des composants de la TOE

Tâches du développeur :

AMA_SIA.2.1D L'analyste de sécurité du développeur doit, pour la version courante de la TOE, fournir une analyse d'impact sur la sécurité qui couvre tous les changements effectués par rapport à la version certifiée de la TOE.

Contenu et présentation des éléments de preuve :

AMA_SIA.2.1C L'analyse d'impact sur la sécurité doit identifier la version certifiée de la TOE à partir de laquelle a été constituée la version courante de la TOE.

AMA_SIA.2.2C L'analyse d'impact sur la sécurité doit identifier tout composant nouveau ou modifié de la TOE, classé comme étant dédié à l'application de la TSP.

AMA_SIA.2.3C L'analyse d'impact sur la sécurité doit, pour tout changement concernant la cible de sécurité ou les représentations de la TSF, décrire brièvement ce changement et tous les effets qu'il peut avoir sur les niveaux de représentation moins abstraits.

AMA_SIA.2.4C L'analyse d'impact sur la sécurité doit, pour tout changement concernant la cible de sécurité ou les représentations de la TSF, identifier toutes les fonctions de sécurité TI et tous les composants de la TOE classés comme étant dédiés à l'application de la TSP, qui sont concernés par ce changement.

AMA_SIA.2.5C L'analyse d'impact sur la sécurité doit, pour tout changement qui implique une modification de la représentation de l'implémentation de la TSF ou de l'environnement TI, identifier les éléments de preuve relatifs à des tests qui montrent, au niveau d'assurance requis, que la TSF continue à être correctement implémentée à la suite de ce changement.

AMA_SIA.2.6C L'analyse d'impact sur la sécurité doit, pour chaque exigence d'assurance applicable dans les classes d'assurance "gestion de configuration" (ACM), "support au cycle de vie" (ALC), "livraison et exploitation" (ADO) et "guides" (AGD), identifier toutes les fournitures d'évaluation qui ont changé et fournir une brève description de chaque changement et de son impact sur l'assurance.

AMA_SIA.2.7C L'analyse d'impact sur la sécurité doit, pour chaque exigence d'assurance applicable dans la classe d'assurance "estimation des vulnérabilités" (AVA), identifier les fournitures d'évaluation qui ont changé et celles qui n'ont pas changé, et justifier des décisions concernant la mise à jour ou non de la fourniture.

Tâches de l'évaluateur :

AMA_SIA.2.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

AMA_SIA.2.2E L'évaluateur doit vérifier que l'analyse d'impact sur la sécurité documente **toutes** les modifications à un niveau de détail approprié, avec les justifications appropriées qui montrent que l'assurance a été maintenue dans la version courante de la TOE.

Annexe A
(Informative)

Références croisées des dépendances entre composants d'assurance

588 Les dépendances renseignées dans les composants des chapitres 8 à 14 et du chapitre 16, sont des dépendances directes entre les composants d'assurance. Le tableau A.1 résume à la fois les dépendances directes et les dépendances indirectes. Les dépendances indirectes correspondent aux résultats cumulés de l'application itérative des dépendances de chaque composant identifié comme ayant une relation de dépendance avec un autre.

Nom du comp.	AUT	CAP	SCP	DEL	IGS	FSP	HLD	IMP	INT	LLD	RCR	SPM	ADM	USR	DVS	FLR	LCD	TAT	COV	DPT	FUN	IND	CCA	MSU	SOF	VLA
AUT.1-2		**3**	*1*												*1*											
CAP.1-2																										
CAP.3-4			**1**												**1**											
CAP.5			**1**												**2**											
SCP.1-3		**3**													*1*											
DEL.1																										
DEL.2-3		**3**	*1*												*1*											
IGS.1-2						*1*					*1*		**1**													
FSP.1-4											**1**															
HLD.1-2						**1**					**1**															
HLD.3-4						**3**					**2**															
HLD.5						**4**					**3**															
IMP.1-2						*1*	*2*			**1**	**1**							**1**								
IMP.3						*1*	*2*		**1**	**1**	**1**							**1**								
INT.1-2						*1*	*2*	**1**		**1**	*1*							*1*								
INT.3						*1*	*2*	**2**		**1**	*1*							*1*								
LLD.1						*1*	**2**				**1**															
LLD.2						*3*	**3**				**2**															
LLD.3						*4*	**5**				**3**															
RCR.1-3																										
SPM.1-3						**1**					*1*															
ADM.1						**1**					*1*															
USR.1						**1**					*1*															

Tableau A.1 - Dépendances des composants d'assurance[a]

Nom du comp.	AUT	CAP	SCP	DEL	IGS	FSP	HLD	IMP	INT	LLD	RCR	SPM	ADM	USR	DVS	FLR	LCD	TAT	COV	DPT	FUN	IND	CCA	MSU	SOF	VLA
DVS.1-2																										
FLR.1-3																										
LCD.1-3																										
TAT.1-3						*1*	*2*	**1**		*1*	*1*															
COV.1-3						**1**					*1*										**1**					
DPT.1						*1*	**1**				*1*										**1**					
DPT.2						*1*	**2**			**1**	*1*										**1**					
DPT.3						*1*	**2**	**2**		**1**	*1*							*1*			**1**					
FUN.1-2																										
IND.1						**1**					*1*		**1**	**1**												
IND.2-3						**1**					*1*		**1**	**1**							**1**					
CCA.1-3						**2**	*2*	**2**		*1*	*1*		**1**	**1**				*1*								
MSU.1-3					**1**	**1**					*1*		**1**	**1**												
SOF.1						**1**	**1**				*1*															
VLA.1						**1**	**1**				*1*		**1**	**1**												
VLA.2-4						**1**	**2**	**1**		**1**	*1*		**1**	**1**				*1*								
AMP.1		**2**														**1**										
CAT.1		**2**																								
EVD.1																										
SIA.1-2																										

Tableau A.1 - Dépendances des composants d'assurance[a]

a. Dans le tableau A.1, la colonne de gauche représente les groupements de composants spécifiques (indiquant uniquement les trois dernières lettres du nom du composant et le numéro ou la série de numéros du ou des composants). Chaque case du tableau qui n'est pas vide indique un composant spécifique, identifié par son nom en haut de la colonne et son numéro figurant dans la case, dont le composant de la colonne de gauche dépend. Les numéros en gras représentent des dépendances directes. Les numéros en italique représentent des dépendances indirectes. Les cases grisées représentent l'intersection d'un composant avec lui-même. Les dépendances des composants AMA vis-à-vis des composants d'assurance sont indiquées dans le tableau A.1, alors que les dépendances internes à AMA sont indiquées dans le tableau A.2 ci-dessous. Il n'y a pas de dépendances des composants n'appartenant pas à AMA vis-à-vis de composants AMA ; ainsi le tableau A.1 n'a pas de colonne représentant les familles AMA.

Noms des comp. AMA	AMP	CAT	EVD	SIA
AMP.1		**1**		
CAT.1				
EVD.1	**1**	*1*		**1**
SIA.1-2		**1**		

Tableau A.2 - Dépendances internes à AMA

Annexe B
(Informative)

Références croisées EAL / composants d'assurance

589 Le tableau B.1 décrit les relations entre les niveaux d'assurance de l'évaluation et les classes, familles et composants d'assurance.

Classe d'assurance	Famille d'assurance	Composants d'assurance par niveau d'assurance de l'évaluation						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Gestion de configuration	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Livraison et exploitation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Développement	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_INT					**1**	**2**	**3**
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guides	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Support au cycle de vie	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Estimation des vulnérabilités	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

Tableau B.1 - Synthèse des niveaux d'assurance de l'évaluation