

中华人民共和国国家标准

GB/T 18336.3—2015/ISO/IEC 15408-3:2008
代替 GB/T 18336.3—2008

信息技术 安全技术 信息技术安全评估准则 第3部分：安全保障组件

Information technology—Security techniques—
Evaluation criteria for IT security—
Part 3: Security assurance components

(ISO/IEC 15408-3:2008, IDT)

2015-05-15 发布

2016-01-01 实施



中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会

发布

目 次

前言 V

引言 VII

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 概述 1

 4.1 本部分的结构 1

5 保障范型 2

 5.1 ISO/IEC 15408 的基本原则 2

 5.2 保障方法 2

 5.3 ISO/IEC 15408 评估保障尺度 3

6 安全保障组件 3

 6.1 安全保障类、族和组件结构 3

 6.2 评估保障级结构 7

 6.3 组合保障包结构 8

7 评估保障级 10

 7.1 评估保障级(EAL)概述 11

 7.2 评估保障级细节 12

 7.3 评估保障级 1(EAL1)——功能测试 12

 7.4 评估保障级 2(EAL2)——结构测试 13

 7.5 评估保障级 3(EAL3)——系统地测试和检查 14

 7.6 评估保障级 4(EAL4)——系统地设计、测试和复查 15

 7.7 评估保障级 5(EAL5)——半形式化设计和测试 17

 7.8 评估保障级 6(EAL6)——半形式化验证的设计和测试 18

 7.9 评估保障级 7(EAL7)——形式化验证的设计和测试 19

8 组合保障包 21

 8.1 组合保障包(CAP)概述 21

 8.2 组合保障包细节 22

 8.3 组合保障级 A(CAP-A)——结构组合 22

 8.4 组合保障级别 B(CAP-B)——系统组合 23

 8.5 组合保障级 C(CAP-C)——系统组合、测试和复查 24

9 APE 类:保护轮廓评估 25

 9.1 PP 引言(APE_INT) 26

 9.2 符合性声明(APE_CCL) 26

 9.3 安全问题定义(APE_SPD) 28

9.4	安全目的(APE_OBJ)	29
9.5	扩展组件定义(APE_ECD)	30
9.6	安全要求(APE_REQ)	31
10	ASE类:安全目标评估	33
10.1	ST引言(ASE_INT)	34
10.2	符合性声明(ASE_CCL)	35
10.3	安全问题定义(ASE_SPD)	36
10.4	安全目的(ASE_OBJ)	37
10.5	扩展组件定义(ASE_ECD)	38
10.6	安全要求(ASE_REQ)	39
10.7	TOE概要规范(ASE_TSS)	41
11	ADV类:开发	43
11.1	安全架构(ADV_ARC)	46
11.2	功能规范(ADV_FSP)	48
11.3	实现表示(ADV_IMP)	56
11.4	TSF内部(ADV_INT)	58
11.5	安全策略模型(ADV_SPM)	61
11.6	TOE设计(ADV_TDS)	63
12	AGD类:指导性文档	71
12.1	操作用户指南(AGD_OPE)	71
12.2	准备程序(AGD_PRE)	73
13	ALC类:生命周期支持	74
13.1	CM能力(ALC_CMC)	75
13.2	CM范围(ALC_CMS)	82
13.3	交付(ALC_DEL)	86
13.4	开发安全(ALC_DVS)	87
13.5	缺陷纠正(ALC_FLR)	89
13.6	生命周期定义(ALC_LCD)	92
13.7	工具和技术(ALC_TAT)	94
14	ATE类:测试	97
14.1	覆盖(ATE_COV)	98
14.2	深度(ATE_DPT)	100
14.3	功能测试(ATE_FUN)	103
14.4	独立测试(ATE_IND)	105
15	AVA类:脆弱性评定	108
15.1	应用注释	108
15.2	脆弱性分析(AVA_VAN)	109
16	ACO类:组合	113
16.1	组合基本原理(ACO_COR)	116
16.2	开发证据(ACO_DEV)	116
16.3	依赖部件的依赖性(ACO_REL)	119

16.4 组合 TOE 测试(ACO_CTT) 121

16.5 组合脆弱性分析(ACO_VUL) 123

附录 A (资料性附录) 开发(ADV) 127

附录 B (资料性附录) 组合(ACO) 140

附录 C (资料性附录) 保障组件依赖关系的交叉引用 145

附录 D (资料性附录) PP 和保障组件的交叉引用 150

附录 E (资料性附录) EAL 和保障组件的交叉引用 151

附录 F (资料性附录) CAP 和保障组件的交叉引用 152

前 言

GB/T 18336《信息技术 安全技术 信息技术安全评估准则》分为以下三部分：

- 第1部分：简介和一般模型；
- 第2部分：安全功能组件；
- 第3部分：安全保障组件。

本部分是 GB/T 18336 的第3部分。

本部分按照 GB/T 1.1—2009 给出的规则起草。

本部分代替 GB/T 18336.3—2008《信息技术 安全技术 信息技术安全评估准则 第3部分：安全保证要求》。

本部分与 GB/T 18336.3—2008 的主要差异如下：

- 将“保证”(assurance)改为“保障”；
- 将“6 安全保证要求”改为“6 安全保障组件”；
- 删除了“6.3 保护轮廓和安全目标评估准则类结构”、“6.4 本部分中术语的用法”、“6.5 保证分类”、“6.6 保证类和族概况”；
- 将“6.1.5 EAL 结构”调整为本部分的“6.2 评估保障级结构”；
- 增加了“6.3 组合保障包结构”；
- 删除了“7 保护轮廓与安全目标评估准则”、“11 保证类、族和组件”；
- 增加了“8 组合保障包”；
- 删除了“8.1 TOE 描述”；
- 增加了“9.2 符合性声明”；
- 将“8.2 安全环境”、“8.6 明确陈述的 IT 安全要求”改为本部分的“9.3 安全问题定义”、“9.5 扩展组件定义”；
- 删除了“9.1 TOE 描述”、“9.5 PP 声明”；
- 增加了“10.2 符合性声明”；
- 将“9.2 安全环境”、“9.7 明确陈述的 IT 安全要求”改为本部分的“10.3 安全问题定义”、“10.5 扩展组件定义”；
- 删除了“ADV 类：开发”中的“高层设计(ADV_HLD)”、“低层设计(ADV_LLD)”、“表示对应性(ADV_RCR)”；
- 在“ADV 类：开发”中增加了“安全架构(ADV_ARC)”、“TOE 设计(ADV_TDS)”；
- 将 AGD 类的“管理员指南(AGD_ADM)”和“用户指南(AGD_USR)”调整为本部分的“操作用户指南(AGD_OPE)”和“准备程序(AGD_PRE)”；
- 将 ACM 类的“CM 能力(ACM_CAP)”、“CM 范围(ACM_SCP)”，ADO 类的“交付(ADO_DEL)”合到了 ALC 类中；
- 删除了“ACM 类：配置管理”中的“CM 自动化(ACM_AUT)”；
- 删除了“ADO 类：交付和运行”中的“安装、生成和启动(ADO_IGS)”；
- 将“测试覆盖(ATE_COV)”改为“覆盖(ATE_COV)”，将“测试深度(ATE_DPT)”改为“深度(ATE_DPT)”；
- 删除了“AVA 类：脆弱性评定”中的“隐蔽信道分析(AVA_CCA)”、“误用(AVA_MSU)”、“TOE 安全功能强度(AVA_SOF)”；

- 将“脆弱性分析(AVA_VLA)”改为“脆弱性分析(AVA_VAN)”；
- 增加了“16 ACO类:组合”；
- 增加了“附录A 开发(ADV)”、“附录B 组合(ACO)”、“附录D PP和保障组件的交叉引用”、“附录F CAP和保障组件的交叉引用”；
- 将“附录A 保证组件依赖关系的交叉引用”调整为“附录C 保障组件依赖关系的交叉引用”，将“附录B EAL和保证组件的交叉引用”调整为“附录E EAL和保障组件的交叉引用”。

本部分使用翻译法等同采用ISO/IEC 15408-3:2008《信息技术 安全技术 信息技术安全评估准则 第3部分:安全保障组件》。

与本部分中规范性引用的国际文件有一致性对应关系的我国文件如下：

- GB/T 18336.1 信息技术 安全技术 信息技术安全评估准则 第1部分:简介和一般模型 (GB/T 18336.1—2015,ISO/IEC 15408-1:2009,IDT)
- GB/T 18336.2 信息技术 安全技术 信息技术安全评估准则 第2部分:安全功能组件 (GB/T 18336.2—2015,ISO/IEC 15408-2:2008,IDT)

本部分由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本部分起草单位:中国信息安全测评中心、信息产业信息安全测评中心、公安部第三研究所。

本部分主要起草人:张翀斌、郭颖、石竑松、毕海英、张宝峰、高金萍、王峰、杨永生、李国俊、董晶晶、谢蒂、王鸿娴、张怡、顾健、邱梓华、宋好好、陈妍、杨元原、许源、饶华一、吴毓书、毛军捷。

本部分所代替标准的历次版本发布情况为：

- GB/T 18336.3—2001
- GB/T 18336.3—2008

引言

本部分定义的安全保障组件是在保护轮廓(PP)或安全目标(ST)中描述安全保障要求的基础。

这些要求建立了一种描述评估对象(TOE)保障要求的标准方法。本部分列出了一组保障组件、族和类,还定义了评估 PP 和 ST 的准则,定义了评估保障级别(EAL)来描述 ISO/IEC 15408 中预定义的用于评定 TOE 保障要求满足情况的尺度。

本部分的目标读者主要包括安全 IT 产品的消费者、开发者和评估者。ISO/IEC 15408-1 提供了关于 ISO/IEC 15408 的目标读者,以及目标读者群体如何使用 ISO/IEC 15408 的补充信息。这些读者群体可以如下方式使用本部分:

- a) 消费者,在选取组件描述保障要求,以满足 PP 或 ST 中提出的安全目的,以及确定所需的安全保障级别时都可使用本部分;
- b) 开发者,在构造 TOE 以响应实际的或预想的消费者安全要求时,可参考本部分以解释保障要求陈述并确定 TOE 的保障方法;
- c) 评估者,在确定 TOE 的保障级别以及评估 PP 和 ST 时,使用本部分所定义的保障要求作为评估准则的强制性陈述。

信息技术 安全技术
信息技术安全评估准则
第 3 部分：安全保障组件

1 范围

GB/T 18336 的本部分定义了保障要求,包括:评估保障级(EAL)——为度量部件 TOE 的保障定义了一种尺度;组合保障包(CAP)——为度量组合 TOE 的保障提供了一种尺度;组成保障级和保障包的单个保障组件;PP 和 ST 的评估准则。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

ISO/IEC 15408-1 信息技术 安全技术 信息技术安全评估准则 第 1 部分:简介和一般模型 (Information technology—Security techniques—Evaluation criteria for IT security—Part 1: Introduction and general model)

ISO/IEC 15408-2 信息技术 安全技术 信息技术安全评估准则 第 2 部分:安全功能组件 (Information technology—Security techniques—Evaluation criteria for IT security—Part 2: Security functional components)

3 术语和定义

ISO/IEC 15408-1 中界定的术语、定义和缩略语适用于本文件。

4 概述

4.1 本部分的结构

第 5 章描述了在本部分的安全保障要求中使用的范型。

第 6 章描述了保障类、族、组件和评估保障级的表示结构以及它们之间的关系,包括组合保障包的结构。同时还刻画了第 9 章到第 16 章中陈述的保障类和族的特征。

第 7 章给出了评估保障级(EAL)的详尽定义。

第 8 章给出了组合保障包(CAP)的详尽定义。

第 9 章到第 16 章给出了本部分保障类的详尽定义。

附录 A 给出了开发类相关概念的进一步解释和实例。

附录 B 给出了组合 TOE 评估和组合类概念的解释。

附录 C 给出了保障组件之间依赖关系的总结。

附录 D 给出了 PP 和族以及 APE 类组件之间的交叉引用。

附录 E 给出了评估保障级(EAL)和保障组件之间的交叉引用。

附录 F 给出了组合保障包(CAP)和保障组件之间的交叉引用。

5 保障范型

本章旨在阐述支撑 ISO/IEC 15408 保障方法的理念。通过对本章的认识将使读者理解隐含在本部分保障要求中的基本原理。

5.1 ISO/IEC 15408 的基本原则

ISO/IEC 15408 的基本原则是,安全威胁和组织安全策略承诺应描述清楚,并且所提出的安全措施应可论证足以达到所期望的安全目的。

进一步地说,应采取措施来减少出现脆弱性的可能性,降低使用(有意利用或者无意触发)脆弱性的能力,以及由此导致的破坏程度。另外,还应采取一些措施,以便于后续标识脆弱性,消除、减轻脆弱性的影响,并/或在脆弱性被利用或触发时进行通告。

5.2 保障方法

ISO/IEC 15408 的理念是,通过对将被信任的 IT 产品进行评估(主动调查)来提供保障。评估是提供保障的传统手段,并且是先前评估准则文档的基础。为了与现行的方法保持一致,ISO/IEC 15408 采用了相同的理念。ISO/IEC 15408 建议由专业的评估人员以递增评估范围、深度和严格程度的方式来度量文档和对应 IT 产品的有效性。

ISO/IEC 15408 不排斥也不评论其他获得保障方法的相关优点。有关获得保障的其他方法仍在研究当中,一旦产生成熟的、可选择的方法,可以考虑把它们纳入到 ISO/IEC 15408 中,因为 ISO/IEC 15408 的结构允许引入新的内容。

5.2.1 脆弱性的意义

假定存在威胁者,他们将积极寻求违反安全策略的可乘之机,无论是为了非法获利还是出于善意的目的,其行为都是不安全的。威胁者也可能偶然触发了脆弱性,造成对组织的危害。由于存在处理敏感信息的需求而又缺乏充分可信的产品,IT 失效将会导致很大的风险。因此,破坏 IT 安全性可能造成重大的损失。

通过脆弱性的有意利用或无意触发,破坏 IT 安全性的事件常发生在商用方面的 IT 应用过程中。应采取一定的措施防止在 IT 产品中出现脆弱性。在可行的情况下,脆弱性应该被:

- a) 消除——应采取积极的措施发现、排除或者抑制所有可利用的脆弱性;
- b) 最小化——应采取积极的措施减少任何因脆弱性利用而造成的潜在影响,使残留的脆弱性达到一个可接受的程度;
- c) 监视——应采取积极的措施确保任何利用残余脆弱性的企图都将被察觉,以便采取措施限制由此带来的损失。

5.2.2 脆弱性产生的原因

下列过程的失效可产生脆弱性:

- a) 要求——IT 产品可具有所有必需的功能和特征,也仍然可能包含脆弱性,致使产品在安全性方面不适当或者不起作用;
- b) 开发——IT 产品不符合其规范要求,和/或由于遵循了不良的开发标准或不正确的设计抉择而引入了脆弱性;
- c) 运行——IT 产品已经按正确的规范被正确构造,但是在其运行过程中由于缺乏控制而引入了脆弱性。

5.2.3 ISO/IEC 15408 保障

保障是信任一个 IT 产品满足其安全目的的基础。保障可从诸如未经证实的声明、先前相关经验或者特定经验等中导出。然而,ISO/IEC 15408 通过主动调查来提供保障。主动调查就是以确定 IT 产品安全特性为目的对 IT 产品进行评估。

5.2.4 通过评估获得保障

评估是获取保障的传统手段,并且是 ISO/IEC 15408 方法的基础。评估技术包括但不限于以下这些:

- a) 分析并检查过程和程序;
- b) 检查过程和程序是否正在被使用;
- c) 分析 TOE 各设计表示之间的一致性;
- d) 对照要求,分析 TOE 的设计表示;
- e) 验证证据;
- f) 分析指导性文档;
- g) 分析所开发的功能测试和所提供的结果;
- h) 独立的功能测试;
- i) 脆弱性分析(包括缺陷假设);
- j) 穿透性测试。

5.3 ISO/IEC 15408 评估保障尺度

ISO/IEC 15408 的基本原则确信,更高的保障级别源于更多的评估努力,其目的是运用最小的努力来获得必要的保障级。努力程度的增长基于:

- a) 范围——努力越多表明该 IT 产品被纳入了评估范围的部分越多;
- b) 深度——努力越多表明对该 IT 产品的设计和实现细节评估得越细;
- c) 严格性——努力越多表明对该 IT 产品的评估采用了越具结构性和形式化的方法。

6 安全保障组件

6.1 安全保障类、族和组件结构

以下内容描述了用于表示保障类、族和组件的结构。

图 1 图示说明了本部分所定义的安全保障要求(SAR)。需注意的是保障要求中最抽象的集合称作类。每一类包含多个保障族,每一族又包含多个保障组件,每一组件同样又包含多个保障元素。类和族用于提供对保障要求进行分类的分类法,而组件用来详细定义 PP/ST 中的安全保障要求。

6.1.1 保障类结构

保障类的结构如图 1 所示。

6.1.1.1 类名

每一保障类被分配了一个唯一的类名。类名表明该保障类所涵盖的主题。

还提供了保障类名对应的唯一缩写形式,这是引用该保障类的主要方式。采用“A”后跟两个与类名有关的字母来表示。

6.1.1.2 类介绍

每个保障类有一段介绍,描述类的组成,并且包含涉及该类意图的支持性文字。

6.1.1.3 保障族

每个保障类至少包含一个保障族。保障族的结构将在以下内容中进行介绍。

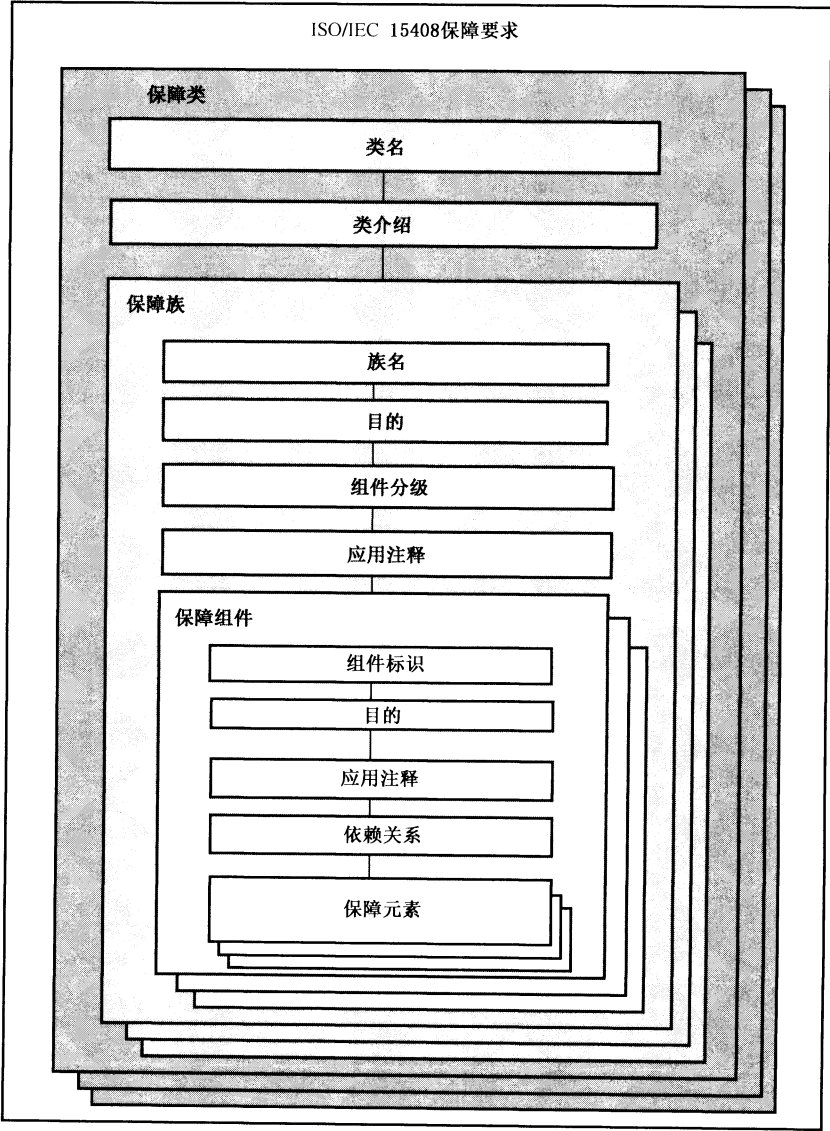


图 1 保障类/族/组件/元素的层次

6.1.2 保障族结构

保障族的结构如图 1 所示。

6.1.2.1 族名

每个保障族被分配了一个唯一的族名。该名字提供了与保障族所涵盖主题相关的描述性信息。每个保障族归属于一个保障类,这个保障类也包括具有相同意图的其他保障族。

保障族名也有一个唯一的缩写形式,这是引用该保障族的主要方式。其表示方法是所在类名的缩写,紧跟着一个下划线,然后再加上与族名有关的三个字母。

6.1.2.2 目的

保障族的目的内容说明了保障族的意图。
描述了该保障族所要满足的目的,特别是那些与 ISO/IEC 15408 保障范型有关的目的。保障族的这部分描述是一般性描述。任何目的的细节描述都包含在特定的保障组件中。

6.1.2.3 组件分级

每个保障族包含一个或多个保障组件。保障族部分描述可供使用的组件并且解释各组件之间的差异。一旦确定该保障族对 PP/ST 保障要求而言是必需或是有用的,就要区分这些保障组件,这是组件分级的主要目的。
含有超过一个组件的保障族将被分级,并提供组件是如何分级的原理性解释。原理是按照范围、深度和/或严格性三方面来界定的。

6.1.2.4 应用注释

如果有保障族应用注释,应包含该保障族的一些附加信息。这些信息应该是保障族用户(例如 PP 和 ST 的作者、TOE 的设计者、评估者)特别感兴趣的。这部分描述是非正式的,常包括关于使用限制的一些警告和特别需要注意的地方。

6.1.2.5 保障组件

每个保障族至少包括一个保障组件。保障组件的结构将在 6.1.3 中进行描述。

6.1.3 保障组件结构

保障组件的结构如图 2 所示。

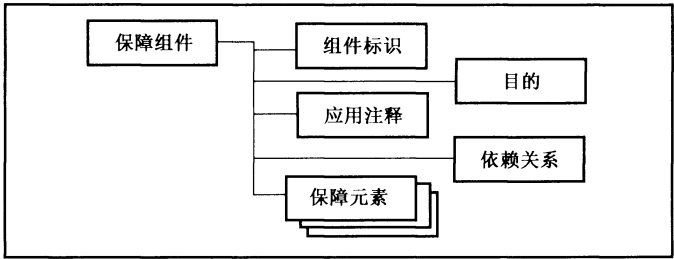


图 2 保障组件结构

保障族内组件之间的关系用粗体突出表示。新的、增强的或在同一类已有组件之上修订的要求,用粗体突出表示。

6.1.3.1 组件标识

组件标识内容提供了识别、分类、注册和引用某一组件所必要的描述信息。
每个保障组件被分配了一个唯一的组件名。该组件名提供关于该保障组件所涵盖主题的描述性信息。每个保障组件均被放置在与它具有相同安全目的的保障族之内。
也提供了保障组件名字的一个唯一缩写形式,这是引用保障组件的主要手段。其形式为族名的缩写,后面加一个点,然后是一个数字,这个数字是根据组件在族内的顺序从 1 开始编号的。

6.1.3.2 目的

如果有保障组件目的,应包含该特定保障组件的特定目的以及该组件的特定意图和目的的详尽解释。

6.1.3.3 应用注释

如果有保障组件的应用注释,应包含一些附加的信息,以便于使用该组件。

6.1.3.4 依赖关系

当一个组件无法自我满足而依赖于另一个组件的存在时,依赖关系就出现在这些保障组件中。

每一个保障组件提供了对其他保障组件的依赖关系的一个完整列表。某些组件可能列出“无依赖关系”,这表明没有确定的依赖关系。被依赖的组件也可能依赖于其他组件。

依赖关系列表标识了所依赖的保障组件的最小集合。在依赖关系列表中,等级比该组件低的那些组件也可以要满足该依赖关系。

在特殊情况下,所指示的依赖关系可能并不适用。PP/ST 作者在提供为什么给定的依赖关系不适用的理由后,可以选择不满足该依赖关系。

6.1.3.5 保障元素

每一个保障组件包含一组保障元素。一个保障元素就是一个安全要求,如果再进行进一步细分的话,将不会产生有意义的评估结果。它是 ISO/IEC 15408 认可的最小的安全要求。

每一个保障元素都被确定为属于以下 3 组保障元素中的一组:

- a) 开发者行为元素:应由开发者实施的行为。这组行为靠随后的一组元素中所引用的证据材料来进一步限制。开发者行为要求用元素号后附加一个字母“D”来标识。
- b) 证据的内容和形式元素:所需证据,证据应证实的内容和证据应表达哪些信息。证据的内容和形式要求用元素号后附加字母“C”来标识。
- c) 评估者行为元素:应由评估者实施的行为。这组行为明确包含确认在“证据的内容和形式”元素中规定的要求是否都已满足,也包含开发者除已完成的动作之外还需实施的明确行为和分析。隐藏的评估者行为作为开发者行为元素的结果,虽然没有被“证据的内容和形式”元素覆盖,也应当被实施。评估者行为要求用元素号后附加字母“E”来标识。

“开发者行为”和“证据的内容和形式”元素定义了一组保障要求来表示开发者的职责,以便于论证 TOE 满足 PP 或 ST 中安全功能要求的保障程度。

“评估者行为”从评估的两个方面定义评估者的职责。一方面是依据第 9 章 APE“保护轮廓评估”类和第 10 章 ASE“安全目标评估”类对 PP/ST 的验证;另一方面是验证 TOE 与其功能要求和保障要求的符合性。通过证实 PP/ST 是有效的并且 TOE 满足这些要求,评估者可以确信 TOE 在运行环境中能够解决已定义的安全问题。

开发者行为元素、证据的内容和形式元素以及显式的评估者行为元素,确定了在验证 TOE 的 ST 所作安全声明时评估者应耗费的精力。

6.1.4 保障元素

每个元素代表一个需要满足的要求。这些要求的陈述应当清晰、简洁且无歧义。因此,不能出现复杂句式,即每一可分离的要求将作为单独的元素来说明。

6.1.5 组件分类

本部分包含一些族和组件,以相关保障为基础进行分类。在每一个类的开始是一个图表,指出该类

中的族和族中的组件。

图 3 中所示的类只包含一个族。这个族包含 3 个按线性层次关系排列的组件(即组件 2 在特定行为、特定证据、行为或证据的严格性等方面上比组件 1 要求得更多)。在本部分中保障族都是按线性分层关系组织的,尽管“线性”关系对以后可能增加的保障族而言,不是一个强制性标准。

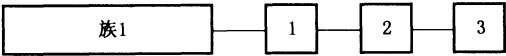


图 3 类分解图实例

6.2 评估保障级结构

在本部分中定义的评估保障级(EAL)和相应的结构如图 4 所示。注意,图中显示出保障组件的内容,意图是通过引用 ISO/IEC 15408 中定义的实际组件把这些信息包含到一个 EAL 中。

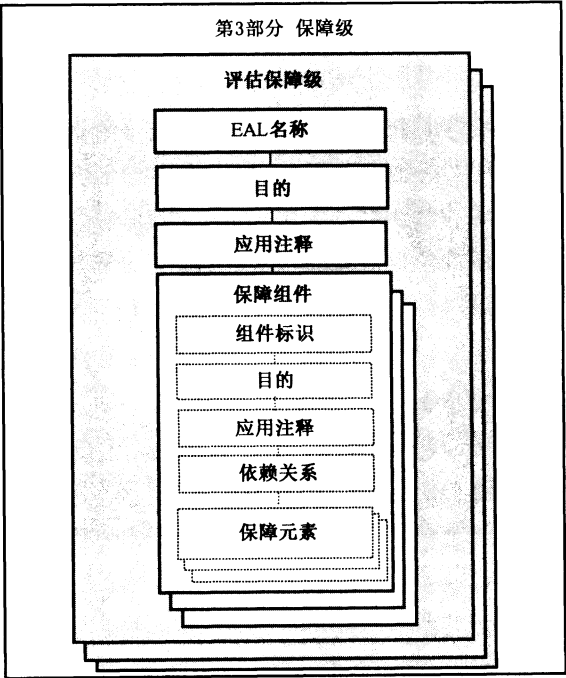


图 4 评估保障级结构

6.2.1 评估保障级名称

每个评估保障级被分配了一个唯一的名称。该名称提供关于该评估保障级意图的描述性信息。也提供了评估保障级名称的一个唯一缩写形式,这是引用评估保障级的主要手段。

6.2.2 目的

本条表示的是评估保障级的意图。

6.2.3 应用注释

如果有评估保障级的应用注释条,其中应包含评估保障级的使用者(如 PP 和 ST 作者,以该评估保障级为目标的 TOE 设计者、评估者)特别感兴趣的信息。这部分描述是非正式的,常包括关于使用限制的一些警告和特别需要注意的地方。

6.2.4 保障组件

每一个评估保障级均选择了一组保障组件。
可以用以下方法,得到一个比给定的 EAL 更高的保障级别:
a) 包含来自其他保障族的额外的保障组件;或
b) 从相同的保障族中用更高级别的保障组件替换保障组件。

6.2.5 保障和保障级之间的关系

图 5 说明了 ISO/IEC 15408 定义的保障要求和保障级之间的关系。当保障组件进一步分解为保障元素时,保障元素不能被保障级单独引用。需要注意的是,图中的箭头表示评估保障级(EAL)与保障类中组件的引用关系。

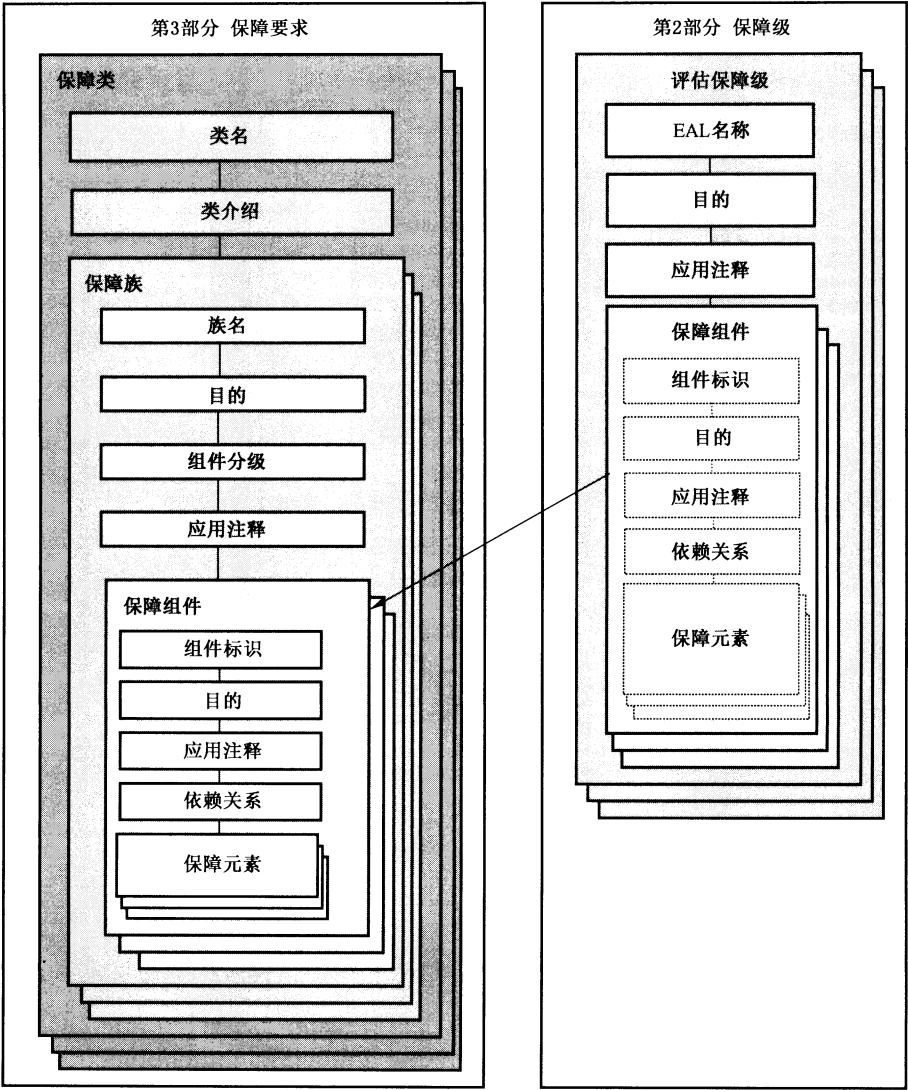


图 5 保障和保障级的关系

6.3 组合保障包结构

组合保障包的结构和评估保障级的结构类似。这两种类型包的主要不同之处是他们所应用的

TOE 类型不同,评估保障级适用于组件 TOE,组合保障包适用于组合 TOE。

在本部分中定义的组合保障包(CAP)和相应的结构如图 6 所示。需要注意的是,该图中显示出保障组件的内容,目的是通过引用 ISO/IEC 15408 中定义的实际组件把这些信息包含到一个组合保障包中。

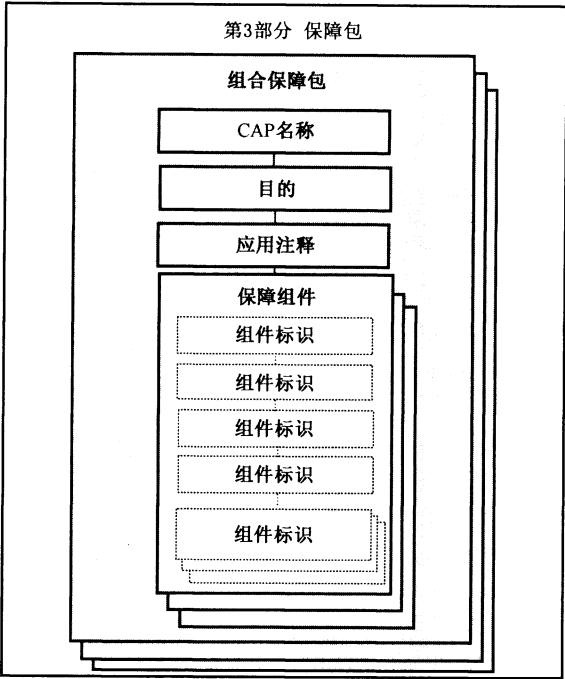


图 6 组合保障包结构

6.3.1 组合保障包名称

每个组合保障包被分配了唯一的名称。该名称提供关于该组合保障包目的的描述性信息。每个组合保障包还有唯一的缩写形式,这是引用组合保障包的主要方法。

6.3.2 目的

本条是对组合保障包目的的描述。

6.3.3 应用注释

如果有组合保障包的应用注释,则该部分包含组合保障包使用者(如 PP 和 ST 作者、以该组合保障包为目标的组合 TOE 的集成者、评估者)特别感兴趣的信息。这部分描述是非正式化的,常包括诸如使用限制的一些警告和特别需要注意的地方。

6.3.4 保障组件

为每一个组合保障包选择了一组保障组件。

一些依赖关系标识了对组合 TOE 活动所依赖组件的评估过程中所完成的活动。在未明确指明与依赖组件活动存在依赖关系时,依赖关系存在于组合 TOE 其他的评估活动中。

比给定的组合保障包更高的保障级别可通过以下方式获得:

- a) 包含来自于其他保障族的额外保障组件;或
- b) 用来自于同一保障族中的更高级别的保障组件替换保障组件。

ACO 类:在 CAP 保障包中的组合组件不能用于提升组件 TOE 的评估级别,因为这无法给组件 TOE 的评估提供任何有意义的保障。

6.3.5 保障和组合保障包之间的关系

图 7 描述了 ISO/IEC 15408 定义的安全保障要求和组合保障包之间的关系。当保障组件进一步分解为保障元素时,保障元素不能被保障包单独引用。注意,图中的箭头表示组合保障包与保障类组件的引用关系。

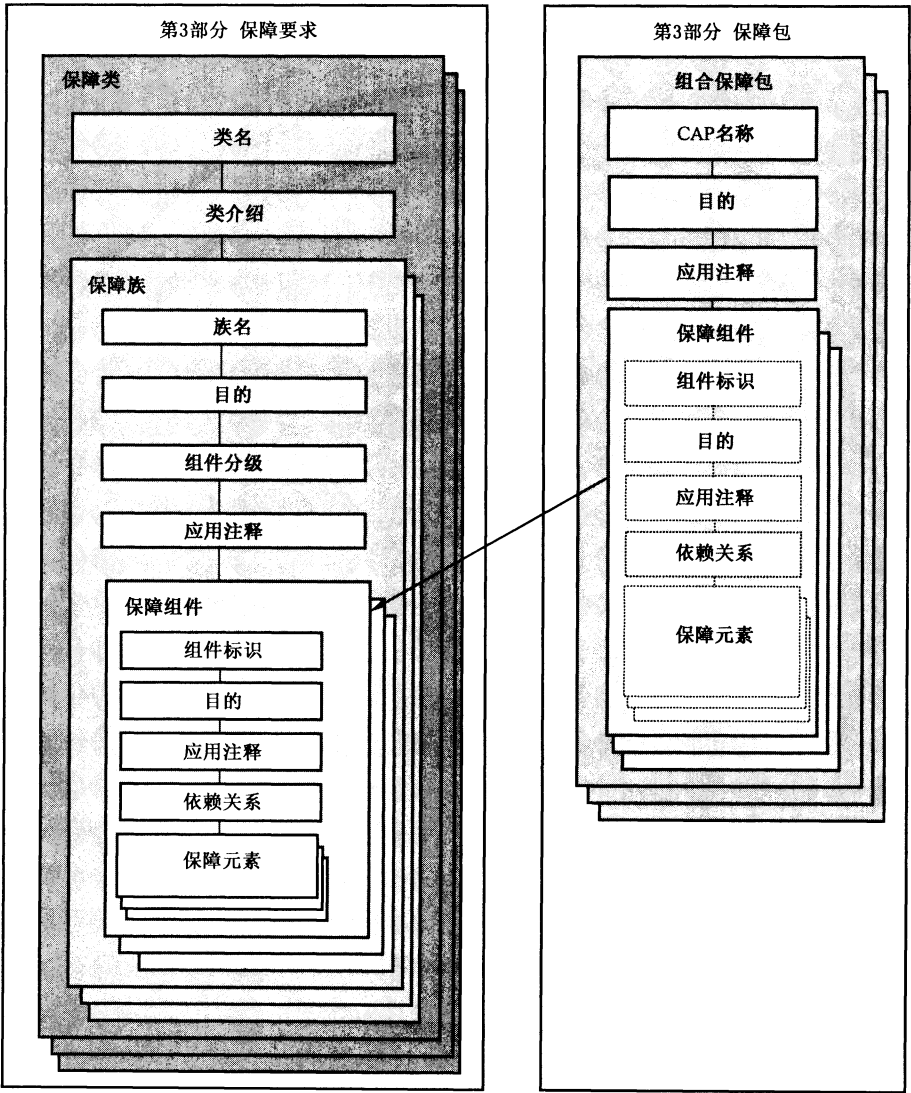


图 7 保障和组合保障包的关系

7 评估保障级

评估保障级(EAL)提供了一种递增的尺度,以保障度的获取开销和可行性来权衡保障的级别。ISO/IEC 15408 使用的方法分离了评估结束时对一个 TOE 建立的保障概念,以及在运行使用该 TOE 时维护该保障的概念。

需要特别注意的是并非本部分中的所有族和组件都包含在这些评估保障级中。这并不是说没有包

含在评估保障级中的这些族和组件不提供有意义的和所需要的保障。相反,期望能把这些族和组件提供的效用作为 PP 和 ST 中的评估保障级的增强。

7.1 评估保障级(EAL)概述

表 1 概括性地描述了评估保障级。其中列表示的是一组按级排序的评估保障级,行表示的是保障族。在表格矩阵中的每一个数字标识出了此处适宜的具体保障组件。

正如在下一条里所总结的那样,在 ISO/IEC 15408 中对 TOE 的保障等级定义了七个级别。它们按级别排序,因为每一个评估保障级比其较低的评估保障级表达更多的保障。从评估保障级到评估保障级的保障增加,靠替换成同一保障族中的一个更高级别的保障组件(即增加严格性、范围和/或深度)和添加另外一个保障族的保障组件(即添加新的要求)来实现。

正如在本部分第 6 章所阐述的那样,这些评估保障级由保障组件的一个适当组合组成。更确切地说,每个评估保障级最多包含每个保障族中的一个组件,以及每个组件的所有保障依赖关系。

虽然这些评估保障级是在 ISO/IEC 15408 中定义的,但还是可以用保障的其他组合来表示。特别是“增强”这个概念允许(从没有包括在评估保障级中的保障族)向评估保障级中增加保障组件,或允许替换评估保障级中的(用同一个保障族的其他更高级别的保障组件)组件。在 ISO/IEC 15408 中定义的保障结构中,只有评估保障级可以增强。“评估保障级减去一个组成保障组件”这样的想法在本部分中不认为是一个有效的主张。要求“增强”者有义务论证对评估保障级添加保障组件的实际意义和额外价值。一个评估保障级也可以用扩展的保障要求来增强。

表 1 评估保障级汇总

保障类	保障族		评估保障级依据的保障组件						
			EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
开发	ADV_ARC	安全架构		1	1	1	1	1	1
	ADV_FSP	功能规范	1	2	3	4	5	5	6
	ADV_IMP	实现表示				1	1	2	2
	ADV_INT	TSF 内部					2	3	3
	ADV_SPM	安全策略模型						1	1
	ADV_TDS	TOE 设计		1	2	3	4	5	6
指导性文档	AGD_OPE	操作用户指南	1	1	1	1	1	1	1
	AGD_PRE	准备程序	1	1	1	1	1	1	1
生命周期支持	ALC_CMC	CM 能力	1	2	3	4	4	5	5
	ALC_CMS	CM 范围	1	2	3	4	5	5	5
	ALC_DEL	交付		1	1	1	1	1	1
	ALC_DVS	开发安全			1	1	1	2	2
	ALC_FLR	缺陷纠正							
	ALC_LCD	生命周期定义			1	1	1	1	2
	ALC_TAT	工具和技术				1	2	3	3

表 1 (续)

保障类	保障族		评估保障级依据的保障组件						
			EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
ST 评估	ASE_CCL	符合性声明	1	1	1	1	1	1	1
	ASE_ECD	扩展组件定义	1	1	1	1	1	1	1
	ASE_INT	ST 引言	1	1	1	1	1	1	1
	ASE_OBJ	安全目的	1	2	2	2	2	2	2
	ASE_REQ	安全要求	1	2	2	2	2	2	2
	ASE_SPD	安全问题定义		1	1	1	1	1	1
	ASE_TSS	TOE 概要规范	1	1	1	1	1	1	1
测试	ATE_COV	覆盖		1	2	2	2	3	3
	ATE_DPT	深度			1	2	3	3	4
	ATE_FUN	功能测试		1	1	1	1	2	2
	ATE_IND	独立测试	1	2	2	2	2	2	3
脆弱性评定	AVA_VAN	脆弱性分析	1	2	2	3	4	5	5

7.2 评估保障级细节

以下各条提供这些评估保障级的定义,用粗体字强调了各级特定要求之间的区别和这些要求的一般特征。

7.3 评估保障级 1(EAL1)——功能测试

7.3.1 目的

EAL1 适用于对(产品的)正确运行需要一定信心,但安全威胁又并不太被看重的场合。对于需要进行独立的保障评估来支持个人信息或类似信息已经得到适当保护的情况,EAL1 具有一定的价值。

EAL1 仅要求一个简化的 ST。该 ST 只需简单地陈述 TOE 满足的安全功能要求,而不用通过从假设、威胁和组织安全策略,进而从安全目的来推导安全功能要求。

EAL1 提供了一个对客户可用的 TOE 的评估,包括依据独立测试和对所提供的指导性文档的检查。意图是在没有 TOE 开发者的帮助下,EAL1 评估也能成功地进行,而且评估所需费用最少。

在这个级别上的评估应当提供这样的证据,即 TOE 的功能与其文档是一致的。

7.3.2 保障组件

EAL1 提供了一种基本的保障级别来理解安全行为,该保障级别是在利用功能和接口规范以及指导性文档的基础上,通过分析一个部分 ST 中的安全功能要求而建立的。

这种分析是通过从公开领域搜索潜在脆弱性并开展 TSF 的独立测试(功能测试和穿透性测试)来获得支持的。

EAL1 还通过 TOE 和相关评估文档的唯一标识来提供保障。表 2 列出 EAL1 级的保障组件。

与未经评估的 IT 产品相比,本 EAL 在保障方面提供了有意义的增强。

表 2 评估保障级 1

保障类	保障组件
ADV:开发	ADV_FSP.1 基本功能规范
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.1 TOE 标识
	ALC_CMS.1 TOE CM 覆盖
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.1 运行环境安全目的
	ASE_REQ.1 陈述性的安全要求
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_IND.1 独立测试—符合性
AVA:脆弱性评定	AVA_VAN.1 脆弱性调查

7.4 评估保障级 2(EAL2)——结构测试

7.4.1 目的

EAL2 需要开发者在交付设计信息和测试结果方面提供配合,除了要求其与良好的商业习惯一致外,不应要求开发方付出更多的努力。这样,就不需要增加过多的费用或时间投入。

EAL2 适用于以下情况:在缺乏现成可用的完整的开发记录时,开发者或用户需要一种低到中等级别的安全性保障。这种情况可能出现在对遗留系统进行安全保护、或者不易联系到开发者的时候。

7.4.2 保障组件

EAL2 在利用功能和接口规范、指导性文档和 TOE 结构的基本描述的基础上,通过分析一个完整的 ST 中的安全功能要求来提供保障,以理解安全行为。

这种分析由对 TSF 的独立测试、开发者基于功能规范进行测试的证据、对开发者测试结果的选择性独立确认、证实可抵御具有基本攻击潜力攻击者攻击的脆弱性分析(基于功能规范、TOE 设计、安全架构描述和提供的指导性证据)等证据来支持。

EAL2 还通过配置管理系统的使用和安全交付程序的证据来提供保障。表 3 列出 EA2 级的保障组件。

与 EAL1 相比,本 EAL 通过增加开发者测试、脆弱性分析(除了公开领域的搜索外)和基于更详细的 TOE 规范进行独立测试等内容,在保障方面提供了有意义的增强。

表 3 评估保障级 2

保障类	保障组件
ADV:开发	ADV_ARC.1 安全架构描述
	ADV_FSP.2 安全执行功能规范
	ADV_TDS.1 基础设计
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.2 CM 系统的使用
	ALC_CMS.2 部分 TOE CM 覆盖
	ALC_DEL.1 交付程序
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.1 陈述性的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_COV.1 覆盖证据
	ATE_FUN.1 功能测试
	ATE_IND.2 独立测试—抽样
AVA:脆弱性评定	AVA_VAN.2 脆弱性分析

7.5 评估保障级 3(EAL3)——系统地测试和检查

7.5.1 目的

EAL3 可使负责任的开发者在设计阶段不需要对现有合理的开发实践作实质性变更,就能从正确的安全工程中获得最大限度的保障。

EAL3 适用于以下情况:开发者或用户需要中等级别的安全性保障,同时要求在不进行大规模重建的情况下,对 TOE 及其开发过程进行彻底调查。

7.5.2 保障组件

EAL3 在利用功能和接口规范、指导性文档和 TOE 的设计架构描述的基础上,通过分析一个完整的 ST 中的安全功能要求来提供保障,以理解安全行为。

这种分析由对 TSF 的独立测试、开发者基于功能规范和 TOE 设计进行测试的证据、对开发者测试结果的选择性独立确认、证实可抵御具有基本攻击潜力攻击者攻击的脆弱性分析(基于功能规范、TOE 设计、安全架构描述和提供的指导性证据)等证据来支持。

EAL3 还通过使用开发环境控制措施、TOE 配置管理和安全交付程序的证据来提供保障。表 4 列出 EAL3 级的保障组件。

与 EAL2 相比,本 EAL 通过增加覆盖安全功能的完备测试,以及提供在开发过程中 TOE 不会被篡改的信任机制和/或程序等内容,在保障方面提供了有意义的增强。

表 4 评估保障级 3

保障类	保障组件
ADV:开发	ADV_ARC.1 安全架构描述
	ADV_FSP.3 带完整摘要的功能规范
	ADV_TDS.2 结构化设计
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.3 授权控制
	ALC_CMS.3 实现表示 CM 覆盖
	ALC_DEL.1 交付程序
	ALC_DVS.1 安全措施标识
	ALC_LCD.1 开发者定义的生命周期模型
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_COV.2 覆盖分析
	ATE_DPT.1 测试:基本设计
	ATE_FUN.1 功能测试
	ATE_IND.2 独立测试—抽样
AVA:脆弱性评定	AVA_VAN.2 脆弱性分析

7.6 评估保障级 4(EAL4)——系统地设计、测试和复查

7.6.1 目的

EAL4 可使开发者基于良好的商业开发惯例,从正确的安全工程中获得最大限度的保障,虽然这种实践很严格,但并不需要大量的专业知识、技巧和其他资源。翻新一个已经存在的生产线很可能是经济上切实可行的,此时 EAL4 是所能达到的最高保障级别。

EAL4 适用于以下这些情况:开发者或用户在传统的商品化 TOE 中需要一个中等到高等级别的安全性保障,并准备负担额外的安全专用的工程费用。

7.6.2 保障组件

EAL4 在利用功能和全部接口规范、指导性文档、TOE 基本模块设计的描述和实现的子集的基础

上,通过分析一个完整的 ST 中的安全功能要求来提供保障,以理解安全行为。

这种分析由对 TOE 安全功能的独立测试、开发者基于功能规范和 TOE 设计进行测试的证据、对开发者测试结果的选择性独立确认、证实可抵御具有**增强型基本攻击**潜力攻击者攻击的脆弱性分析(基于功能规范、TOE 设计、**实现表示**、**结构性设计**和提供的指导性证据)等证据来支持。

EAL4 还通过使用开发环境控制措施、**包括配置管理自动化在内的更多的** TOE 配置管理措施和安全交付程序的证据来提供保障。**表 5 列出 EAL4 级的保障组件。**

与 **EAL3** 相比,本 EAL 通过增加更多的**设计描述**、**所有安全功能的实现表示**,以及为在开发过程中 TOE 不会被篡改提供一定信任的**改进机制和/或程序**,在保障方面**提供了有意义的增强**。

表 5 评估保障级 4

保障类	保障组件
ADV:开发	ADV_ARC.1 安全架构描述
	ADV_FSP.4 完备的功能规范
	ADV_IMP.1 TSF 实现表示
	ADV_TDS.3 基础模块设计
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.4 生产支持和接受程序及其自动化
	ALC_CMS.4 问题跟踪 CM 覆盖
	ALC_DEL.1 交付程序
	ALC_DVS.1 安全措施标识
	ALC_LCD.1 开发者定义的生命周期模型
	ALC_TAT.1 明确定义的开发工具
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_COV.2 覆盖分析
	ATE_DPT.2 安全执行模块
	ATE_FUN.1 功能测试
	ATE_IND.2 独立测试—抽样
AVA:脆弱性评定	AVA_VAN.3 关注点脆弱性分析

7.7 评估保障级 5(EAL5)——半形式化设计和测试

7.7.1 目的

EAL5 可使一个开发者从基于严格的商业开发实践的安全工程中获得最大限度的保障,而这种开发实践是靠专业安全工程技术的适度应用来支持的。设计和开发能够达到 EAL5 保障要求的 TOE。相对于没有应用专业技术的严格开发而言,由 EAL5 要求引起的额外的开销也许不会很大。

EAL5 适用于以下这些情况:开发者或用户在一个有计划的开发过程中需要高级别独立的安全性保障,以及开发者或用户在未由于专业安全工程技术而导致不合理开销的条件下,需要有严格的开发手段。

7.7.2 保障组件

EAL5 在利用功能和全部接口规范、指导性文档、TOE 的设计描述和实现的基础上,通过分析一个完整的 ST 中的安全功能要求来提供保障,以理解安全行为。**此外还需要模块化的 TSF 设计。**

这种分析由 TOE 安全功能的独立测试,开发者基于功能规范、TOE 设计进行测试的证据,对开发者测试结果的选择性独立确认,证实可抵御具有**中等**攻击潜力攻击者攻击的**独立的**脆弱性分析等证据来支持。

EAL5 还通过使用开发环境控制措施、包括配置管理自动化在内的**全面的** TOE 配置管理措施和安全交付程序的证据来提供保障。**表 6 列出 EAL5 级的保障组件。**

与 EAL4 相比,本 EAL 通过增加**半形式化的设计描述、具有可经受结构化分析的架构体系**以及为在开发过程中 TOE 不会被篡改提供一定信任的改进机制和/或程序,在保障方面提供了有意义的增强。

表 6 评估保障级 5

保障类	保障组件
ADV:开发	ADV_ARC.1 安全架构描述
	ADV_FSP.5 附加错误信息的完备的半形式化功能规范
	ADV_IMP.1 TSF 实现表示
	ADV_INT.2 内部结构合理
	ADV_TDS.4 半形式化模块设计
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.4 生产支持和接受程序及其自动化
	ALC_CMS.5 开发工具 CM 覆盖
	ALC_DEL.1 交付程序
	ALC_DVS.1 安全措施标识
	ALC_LCD.1 开发者定义的生命周期模型
	ALC_TAT.2 遵从实现标准

表 6（续）

保障类	保障组件
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_COV.2 覆盖分析
	ATE_DPT.3 测试:模块设计
	ATE_FUN.1 功能测试
	ATE_IND.2 独立测试—抽样
AVA:脆弱性评定	AVA_VAN.4 系统的脆弱性分析

7.8 评估保障级 6(EAL6)——半形式化验证的设计和测试

7.8.1 目的

EAL6 可使开发者通过把安全工程技术应用于严格的开发环境情况下,获得的高级别保障。目的是为了生产一个质优价高的 TOE 来保护高价值的资产避免存在重大的风险。

EAL6 适用于应用在高风险环境下 TOE 的开发,其中由于安全所需的额外开销与受保护资产的价值是相当的。

7.8.2 保障组件

EAL6 在利用功能和全部接口规范、指导性文档、TOE 设计和实现的基础上,通过分析一个完整的 ST 中的安全功能要求来提供保障,以理解安全行为。**还通过以下方式额外地获得保障:所选 TOE 安全策略的形式化模型描述,TOE 设计功能规范的半形式化描述。**此外还需要模块化和层次化的 TSF 设计。

这种分析由 TOE 安全功能的独立测试,开发者基于功能规范、TOE 设计进行测试的证据,对开发者测试结果的选择性独立确认,以及证实可抵御具有高等攻击潜力的攻击者攻击的独立的脆弱性分析等来支持。

EAL6 还通过使用**结构化的开发过程、开发环境控制措施、包括配置管理完全自动化在内的全面的 TOE 配置管理措施和安全交付程序的证据**等来提供保障。**表 7 列出 EAL6 级的保障组件。**

与 EAL5 相比,本 EAL 通过增加**更加全面的分析、实现的结构化描述、更体系化的结构(如分层)、更全面的独立脆弱性分析,以及改进的配置管理和开发环境控制**,在保障方面提供了有意义的增强。

表 7 评估保障级 6

保障类	保障组件
ADV:开发	ADV_ARC.1 安全架构描述
	ADV_FSP.5 附加错误信息的完备的半形式化功能规范
	ADV_IMP.2 TSF 实现表示完全映射
	ADV_INT.3 内部复杂度最小化
	ADV_SPM.1 形式化 TOE 安全策略模型
	ADV_TDS.5 完全半形式化模块设计
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.5 高级支持
	ALC_CMS.5 开发工具 CM 覆盖
	ALC_DEL.1 交付程序
	ALC_DVS.2 充分的安全措施
	ALC_LCD.1 开发者定义的生命周期模型
	ALC_TAT.3 遵从实现标准—所有部分
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_COV.3 严格覆盖分析
	ATE_DPT.3 测试:模块设计
	ATE_FUN.2 顺序的功能测试
	ATE_IND.2 独立测试—抽样
AVA:脆弱性评定	AVA_VAN.5 高级的系统的脆弱性分析

7.9 评估保障级 7(EAL7)——形式化验证的设计和测试

7.9.1 目的

EAL7 适用于安全 TOE 的开发,该 TOE 将应用到极高风险环境和/或高资产价值的情况。EAL7 的实际应用目前只局限于一些非常关注安全功能性、经得起详细地形式化分析的 TOE。

7.9.2 保障组件

EAL7 在利用功能和全部接口规范、指导性文档、TOE 设计和**结构化的实现表示**的基础上,通过分析一个完整的 ST 中的安全功能要求来提供保障,以理解安全行为。还通过以下方式额外地获得

保障:所选 TOE 安全策略的形式化模型描述,TOE 设计功能规范的半形式化描述。此外还需要模块化、层次化和简单化的 TSF 设计。

这种分析由 TOE 安全功能的独立测试,开发者基于功能规范、TOE 设计和实现表示进行测试的证据,对开发者测试结果的**全部**独立确认,以及证实可抵御具有高等攻击潜力攻击者攻击的独立脆弱性分析等来支持。

EAL7 还通过使用结构化的开发过程、开发环境控制措施、包括配置管理完全自动化在内的全面的 TOE 配置管理措施和安全交付程序的证据等来提供保障。**表 8 列出 EAL7 级的保障组件。**

与 **EAL6** 相比,本 EAL 通过要求**利用形式化描述、形式化对应性分析以及全面的测试**而进行更加全面的分析,在保障方面提供了有意义的增强。

表 8 评估保障级 7

保障类	保障组件
ADV:开发	ADV_ARC.1 安全架构描述
	ADV_FSP.6 附加形式化描述的完备的半形式化功能规范
	ADV_IMP.2 TSF 实现表示完全映射
	ADV_INT.3 内部复杂度最小化
	ADV_SPM.1 形式化 TOE 安全策略模型
	ADV_TDS.6 带形式化高层设计表示的完全半形式化模块设计
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.5 高级支持
	ALC_CMS.5 开发工具 CM 覆盖
	ALC_DEL.1 交付程序
	ALC_DVS.2 充分的安全措施
	ALC_LCD.2 可测量的生命周期模型
	ALC_TAT.3 遵从实现标准—所有部分
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范
ATE:测试	ATE_COV.3 严格覆盖分析
	ATE_DPT.4 测试:实现表示
	ATE_FUN.2 顺序的功能测试
	ATE_IND.3 独立测试—完全
AVA:脆弱性评定	AVA_VAN.5 高级的系统的脆弱性分析

8 组合保障包

组合保障包(CAP)在权衡所获得的保障级与达到该组合 TOE 保障程度所需的代价和可行性上提供了一个递增的尺度。

需特别注意的是本部分中的族和组件仅有少部分被包含在组合保障包中。这是由于组合保障包是建立在先前已评估的实体(基本部件和依赖部件)之上,并不是说这些未被选择的族和组件不能提供有意义的和期望的保障。

8.1 组合保障包(CAP)概述

CAP 将用于组合 TOE 的评估。组合 TOE 由已经通过评估(或即将通过评估)的 TOE 部件组成(参见附录 B)。单个部件将被认证为满足某个 EAL 或在 ST 中说明的其他保障包。通过从公开渠道获取部件的信息进行 EAL1 评估,一个组合 TOE 预计可达到基本的保障级别。(部件和组合 TOE 的评估都可应用 EAL1。)通过应用高于 EAL1 的 EAL 可以进行组合 TOE 的更高保障级的评估,而 CAP 为此目的提供了另一种解决方法。

一个依赖部件可用一个先前已评估或认证过的基本部件来满足环境中的 IT 平台要求,但这种方式不提供任何关于部件之间的交互或组合是否会引入脆弱性的保障。组合保障包在更高的保障级别中考虑了这些交互,将部件之间的接口纳入了测试范围。同时,通过对组合 TOE 进行脆弱性分析来考虑部件组合引入脆弱性的问题。

表 9 概括性地描述了这几个 CAP。其中列表示的是一组按级别排序的 CAP,行表示的是保障族。表格中的每一个数字标识出了此处适宜的具体保障组件。

正如在下一条里所总结的那样,在 ISO/IEC 15408 中对组合 TOE 的保障等级定义三个组合保障包。它们按级别排序,因为每一个 CAP 比所有较低的 CAP 表达更多的保障。从 CAP 到 CAP 的保障增加,通过替换成同一保障族中的一个更高级别的保障组件(即增加严格性、范围和或深度)和添加另外一个保障族的保障组件(即添加新的要求)来实现。这些增加导致更多的综合分析来识别对单个 TOE 获得的评估结果的影响。

正如在本部分第 6 章所阐述的那样,这些 CAP 由保障组件的一个适当组合组成。更确切地说,每个 CAP 最多包含每个保障族中的一个组件,并满足每个组件的所有保障依赖关系。

CAP 仅考虑能够抵御增强型基本级攻击潜力的攻击者。这是由于从 ACO_DEV 提供的设计信息级别限制一些和攻击潜力有关因素(组合 TOE 的知识),也影响评估者完成的脆弱性分析的严格性。因此,组合 TOE 的保障级别是有限的,而组合 TOE 中的单个部件的保障可能更高。

表 9 组合保障级别汇总

保障类	保障族	组合保障包依据的保障组件		
		CAP-A	CAP-B	CAP-C
组合	ACO_COR	1	1	1
	ACO_CTT	1	2	2
	ACO_DEV	1	2	3
	ACO_REL	1	1	2
	ACO_VUL	1	2	3

表 9 (续)

保障类	保障族	组合保障包依据的保障组件		
		CAP-A	CAP-B	CAP-C
指导性文档	AGD_OPE	1	1	1
	AGD_PRE	1	1	1
生命周期支持	ALC_CMC	1	1	1
	ALC_CMS	2	2	2
	ALC_DEL			
	ALC_DVS			
	ALC_FLR			
	ALC_LCD			
	ALC_TAT			
ST 评估	ASE_CCL	1	1	1
	ASE_ECD	1	1	1
	ASE_INT	1	1	1
	ASE_OBJ	1	2	2
	ASE_REQ	1	2	2
	ASE_SPD		1	1
	ASE_TSS	1	1	1

8.2 组合保障包细节

8.3~8.5 对 CAP 进行了定义,使用粗体强调了在特定要求和其他普通要求之间的不同。

8.3 组合保障级 A(CAP-A)—结构组合

8.3.1 目的

CAP-A 适于组合 TOE 是完整的,且需要对产生组合的正确安全操作具有信心。这要求依赖部件的开发者在设计信息和依赖部件认证的测试结果方面提供配合,不要求涉及基本部件的开发者。

因此 CPA-A 适用于在缺乏现成可用的完整的开发记录时,开发者或用户需要一种低到中等级别的独立保障的安全性的情况。

8.3.2 保障组件

CAP-A 通过分析组合 TOE 的 ST 提供保障。利用 TOE 组成部件的评估输出(如 ST、指导性文档等)和 TOE 组成部件之间的接口规范对组合 TOE 的 ST 中的安全功能要求进行分析,以理解安全行为。

以下内容可为该分析提供支持:针对关联信息中描述的依赖部件与基本部件之间的接口进行的独立测试;开发者基于关联信息、开发信息和组合原理进行测试的证据;以及对开发者测试结果的选择性独立确认;由评估者进行的组合 TOE 的脆弱性审查等。

CAP-A 也通过唯一的组合 TOE 标识(即 IT TOE 和指南性文档)提供保障。表 10 列出组合保障 A 级的保障组件。

表 10 CAP-A

保障类	保障组件
ACO:组合	ACO_COR.1 组合基本原理
	ACO_CTT.1 接口测试
	ACO_DEV.1 功能描述
	ACO_REL.1 基本的依赖信息
	ACO_VUL.1 组合脆弱性审查
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.1 TOE 标识
	ALC_CMS.2 部分 TOE CM 覆盖
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.1 运行环境安全目的
	ASE_REQ.1 陈述性的安全要求
	ASE_TSS.1 TOE 概要规范

8.4 组合保障级别 B(CAP-B)—系统组合

8.4.1 目的

CAP-B 可使尽责的组合 TOE 开发者在子系统层面上通过理解各组成部件之间的相互影响,获得最大程度的保障,同时还可使基本部件开发者的参与量最小。

CAP-B 适用于以下这些情况:开发者或用户需要一个中等级别的独立保障的安全性,以及在没有对组合 TOE 进行实质性改造的情况下,要求对组合 TOE 及其开发过程进行彻底地调查。

8.4.2 保障组件

CAP-B 通过全面分析组合 TOE 的 ST 提供保障。利用 TOE 组成部件的评估输出(如 ST、指导性文档等),TOE 组成部件之间的接口规范,以及**包含在组合开发过程中的 TOE 设计信息(描述 TSF 子系统)**,对组合 TOE 的 ST 中的安全功能要求进行分析,以理解安全行为。

以下内容可为该分析提供支持:对关联信息(**包括 TOE 设计**)中描述的依赖部件与基本部件之间的接口进行的独立测试、开发者基于关联信息、开发相关信息和组合原理而进行测试的证据,以及对开发者测试结果的选择性独立确认。评估者为证实组合 TOE **可抵御基本攻击潜力攻击者的攻击**而进行的脆弱性分析也可提供支持。

与 CAP-A 相比,本 CAP 通过要求更完备的安全功能测试,在保障方面提供了有意义的增强。表 11 列出组合保障 B 级的保障组件。

表 11 CAP-B

保障类	保障组件
ACO:组合	ACO_COR.1 组合基本原理
	ACO_CTT.2 严格的接口测试
	ACO_DEV.2 基本的设计证据
	ACO_REL.1 基本的依赖信息
	ACO_VUL.2 组合脆弱性分析
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.1 TOE 标识
	ALC_CMS.2 部分 TOE CM 覆盖
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范

8.5 组合保障级 C(CAP-C)—系统组合、测试和复查

8.5.1 目的

CAP-C 可使开发者从对组合 TOE 部件之间交互的正确分析获得最大限度的保障,虽然这种实践很严格,但不需要获得所有的对基本部件的评估证据。

因此 CAP-C 适用于以下这些情况:开发者或用户在传统的商品化 TOE 中需要一个中等到高等级别的独立保障的安全性,并准备负担额外的安全专用工程费用。

8.5.2 保障组件

CAP-C 通过全面分析组合 TOE 的 ST 提供保障。利用 TOE 组成部件的评估输出(如 ST、指导性文档等),TOE 组成部件之间的接口规范,以及包含在组合开发过程中的 TOE 设计信息(描述 TSF 模块),对组合 TOE 的 ST 中的安全功能要求进行分析,以理解安全行为。

以下内容可为该分析提供支持:对关联信息(包括 TOE 设计)中描述的依赖部件与基本部件之间的接口进行的独立测试、开发者基于关联信息、开发相关信息和组合原理而进行测试的证据、对开发者测试结果的选择性独立确认来进行。评估者为证实组合 TOE 可抵御**基本增强型**攻击潜力攻击者的攻击而进行的脆弱性分析也可提供支持。

与 **CAP-B** 相比,本 CAP 通过要求更完备的设计描述和抵御更高的攻击潜力,在保障方面提供了有意义的增强。表 12 列出组合保障 C 级的保障组件。

表 12 CAP-C

保障类	保障组件
ACO:组合	ACO_COR.1 组合基本原理
	ACO_CTT.2 严格的接口测试
	ACO_DEV.3 详细的设计证据
	ACO_REL.2 依赖信息
	ACO_VUL.3 增强的基本组合脆弱性分析
AGD:指导性文档	AGD_OPE.1 操作用户指南
	AGD_PRE.1 准备程序
ALC:生命周期支持	ALC_CMC.1 TOE 标识
	ALC_CMS.2 部分 TOE CM 覆盖
ASE:ST 评估	ASE_CCL.1 符合性声明
	ASE_ECD.1 扩展组件定义
	ASE_INT.1 ST 引言
	ASE_OBJ.2 安全目的
	ASE_REQ.2 推导出的安全要求
	ASE_SPD.1 安全问题定义
	ASE_TSS.1 TOE 概要规范

9 APE 类:保护轮廓评估

PP 评估要求证实 PP 是技术合理和内部一致的,并且,如果 PP 是基于一个或多个 PP 或者包,那么 PP 必须是这些 PP 或包的一个正确的实例化。PP 适于作为编写 ST 或其他 PP 的基础,这些属性是必需的。

本章应该联合 ISO/IEC 15408-1 中的附录 A、附录 B 和附录 C 一起使用,因为这些附录详细阐明了本章的一些概念,并提供了很多示例。

图 8 给出了本类中的族以及族中组件的层次结构。

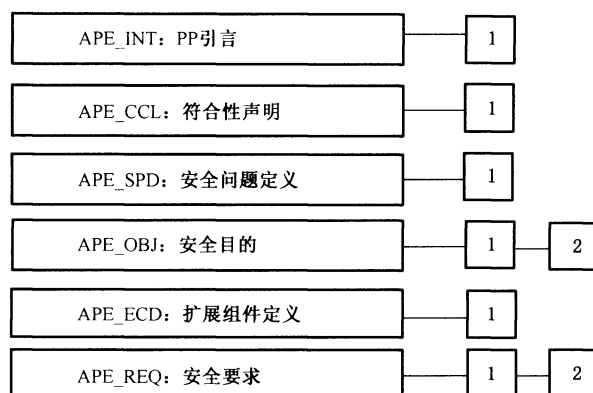


图 8 APE 类:保护轮廓评估类分解

9.1 PP 引言(APE_INT)

9.1.1 目的

本族的目的以叙述的方式描述 TOE。

对 PP 引言的评估需要证实 PP 已被正确标识,并且 PP 参考和 TOE 概述是相互一致的。

9.1.2 APE_INT.1 PP 引言

依赖关系:无。

9.1.2.1 开发者行为元素

9.1.2.1.1 APE_INT.1.1D

开发者应提供 PP 引言。

9.1.2.2 内容和形式元素

9.1.2.2.1 APE_INT.1.1C

PP 引言应包含 PP 参考和 TOE 概述。

9.1.2.2.2 APE_INT.1.2C

PP 参考应唯一标识 PP。

9.1.2.2.3 APE_INT.1.3C

TOE 概述应概括 TOE 的用法及其主要安全特性。

9.1.2.2.4 APE_INT.1.4C

TOE 概述应标识 TOE 类型。

9.1.2.2.5 APE_INT.1.5C

TOE 概述应标识任何 TOE 可用到但又不属于 TOE 的硬件/软件/固件。

9.1.2.3 评估者行为元素

9.1.2.3.1 APE_INT.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

9.2 符合性声明(APE_CCL)

9.2.1 目的

本族的目的是确定符合性声明的正确性。另外,本族详细阐明了如何声明 ST 和其他 PP 是与本 PP 符合的。

9.2.2 APE_CCL.1 符合性声明

依赖关系:APE_INT.1 PP 引言;

APE_ECD.1 扩展组件定义；
APE_REQ.1 规定的安全要求。

9.2.2.1 开发者行为元素

9.2.2.1.1 APE_CCL.1.1D

开发者应提供符合性声明。

9.2.2.1.2 APE_CCL.1.2D

开发者应提供符合性声明的基本原理。

9.2.2.1.3 APE_CCL.1.3D

开发者应提供符合性陈述。

9.2.2.2 内容和形式元素

9.2.2.2.1 APE_CCL.1.1C

PP 的符合性声明应包含 ISO/IEC 15408 的符合性声明,用于标识 PP 声明符合性遵从的 ISO/IEC 15408 的版本。

9.2.2.2.2 APE_CCL.1.2C

ISO/IEC 15408 符合性声明应描述 PP 与 ISO/IEC 15408-2 的符合性,无论是与 ISO/IEC 15408-2 相符还是与 ISO/IEC 15408-2 的扩展部分相符。

9.2.2.2.3 APE_CCL.1.3C

ISO/IEC 15408 符合性声明应描述 PP 与本部分的符合性,无论是与本部分相符还是与本部分的扩展部分相符。

9.2.2.2.4 APE_CCL.1.4C

ISO/IEC 15408 符合性声明应与扩展组件定义是相符的。

9.2.2.2.5 APE_CCL.1.5C

符合性声明应标识 PP 声明符合性遵从的所有 PP 和安全要求包。

9.2.2.2.6 APE_CCL.1.6C

符合性声明应描述 PP 和包的符合性,无论是包的符合性还是包扩展部分。

9.2.2.2.7 APE_CCL.1.7C

符合性声明的基本原理应证实 TOE 类型与符合性声明所遵从的 PP 中的 TOE 类型是相符的。

9.2.2.2.8 APE_CCL.1.8C

符合性声明的基本原理应证实安全问题定义陈述与符合性声明所遵从的 PP 中的安全问题定义陈述是相符的。

9.2.2.2.9 APE_CCL.1.9C

符合性声明的基本原理应证实安全目的陈述与符合性声明所遵从的 PP 中的安全目的陈述是相符的。

9.2.2.2.10 APE_CCL.1.10C

符合性声明的基本原理应证实安全要求陈述与符合性声明所遵从的 PP 中的安全要求陈述是相符的。

9.2.2.2.11 APE_CCL.1.11C

符合性陈述应描述任何 PP/ST 要求的与严格的 PP 或可论证的 PP 的符合性。

9.2.2.3 评估者行为元素

9.2.2.3.1 APE_CCL.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

9.3 安全问题定义(APE_SPD)

9.3.1 目的

PP 的本部分定义 TOE 及其运行环境所负责处理的安全问题。

安全问题定义评估要求证实 TOE 及其运行环境所负责处理的安全问题被明确的定义。

9.3.2 APE_SPD.1 安全问题定义

依赖关系:无。

9.3.2.1 开发者行为元素

9.3.2.1.1 APE_SPD.1.1D

开发者应提供安全问题定义。

9.3.2.2 内容和形式元素

9.3.2.2.1 APE_SPD.1.1C

安全问题定义应描述威胁。

9.3.2.2.2 APE_SPD.1.2C

所有的威胁都应根据威胁者、资产和敌对行为进行描述。

9.3.2.2.3 APE_SPD.1.3C

安全问题定义应描述组织安全策略。

9.3.2.2.4 APE_SPD.1.4C

安全问题定义应描述 TOE 运行环境的相关假设。

9.3.2.3 评估者行为元素

9.3.2.3.1 APE_SPD.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

9.4 安全目的(APE_OBJ)

9.4.1 目的

安全目的是对由 APE_SPD“安全问题定义”族定义的安全问题预期反应的简明陈述。
安全目的评估要求证实安全目的充分而且完备地处理了的安全问题,并证实 TOE 及其运行环境的安全问题被明确的区分和定义。

9.4.2 组件分级

本族中的组件是基于是否仅规定了运行环境安全目的,或者也规定了 TOE 的安全目的而进行分级的。

9.4.3 APE_OBJ.1 运行环境安全目的

依赖关系:无。

9.4.3.1 开发者行为元素

9.4.3.1.1 APE_OBJ.1.1D

开发者应提供安全目的的陈述。

9.4.3.2 内容和形式元素

9.4.3.2.1 APE_OBJ.1.1C

安全目的的陈述应描述运行环境安全目的。

9.4.3.3 评估者行为元素

9.4.3.3.1 APE_OBJ.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

9.4.4 APE_OBJ.2 安全目的

依赖关系:APE_SPD.1 安全问题定义。

9.4.4.1 开发者行为元素

9.4.4.1.1 APE_OBJ.2.1D

开发者应提供安全目的的陈述。

9.4.4.1.2 APE_OBJ.2.2D

开发者应提供安全目的的基本原理。

9.4.4.2 内容和形式元素

9.4.4.2.1 APE_OBJ.2.1C

安全目的的陈述应描述 TOE 的安全目的和运行环境安全目的。

9.4.4.2.2 APE_OBJ.2.2C

安全目的基本原理应追溯到 TOE 的每一个安全目的,以便于能追溯到安全目的所对抗的威胁及其实施的组织安全策略。

9.4.4.2.3 APE_OBJ.2.3C

安全目的基本原理应追溯到运行环境的每一个安全目的,以便于能追溯到安全目的所对抗的威胁、安全目的强制实施的组织安全策略和安全目的支持的假设。

9.4.4.2.4 APE_OBJ.2.4C

安全目的基本原理应证实安全目的能抵抗所有威胁。

9.4.4.2.5 APE_OBJ.2.5C

安全目的基本原理应证实安全目的执行了所有的组织安全策略。

9.4.4.2.6 APE_OBJ.2.6C

安全目的基本原理应证实运行环境安全目的支持所有的假设。

9.4.4.3 评估者行为元素

9.4.4.3.1 APE_OBJ.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

9.5 扩展组件定义(APE_ECD)

9.5.1 目的

扩展安全要求不是基于 ISO/IEC 15408 第 2 部分或第 3 部分的组件要求,而是基于 PF 作者定义的扩展组件。

对扩展组件定义的评估需要确定这些组件是明确的、没有歧义的并且是必要的,如:他们可能没有使用 ISO/IEC 15408 第 2 部分或第 3 部分已有的组件进行清晰的描述。

9.5.2 APE_ECD.1 扩展组件的定义

依赖关系:无。

9.5.2.1 开发者行为元素

9.5.2.1.1 APE_ECD.1.1D

开发者应提供安全要求的陈述。

9.5.2.1.2 APE_ECD.1.2D

开发者应提供扩展组件的定义。

9.5.2.2 内容和形式元素

9.5.2.2.1 APE_ECD.1.1C

安全要求陈述应标识所有扩展的安全要求。

9.5.2.2.2 APE_ECD.1.2C

扩展组件定义应为每一个扩展的安全要求定义一个扩展的组件。

9.5.2.2.3 APE_ECD.1.3C

扩展组件定义应描述每个扩展的组件与已有组件、族和类的关联性。

9.5.2.2.4 APE_ECD.1.4C

扩展组件定义应使用已有的组件、族、类和方法学作为陈述的模型。

9.5.2.2.5 APE_ECD.1.5C

扩展组件应由可测量的和客观的元素组成,以便于证实这些元素之间的符合性或不符合性。

9.5.2.3 评估者行为元素

9.5.2.3.1 APE_ECD.1.1E

评估者应确认所提供的信息满足证据的内容和形式的有关要求。

9.5.2.3.2 APE_ECD.1.2E

评估者应确认扩展组件不能利用已经存在的组件明确的表达。

9.6 安全要求(APE_REQ)

9.6.1 目的

安全功能要求(SFR)对 TOE 预期安全行为进行了清晰、无歧义且定义准确的描述。安全保障要求(SAR)对 TOE 获得保障而采取的预期活动进行了清晰、无歧义且规范的描述。

对安全要求的评估要求确保这些安全要求是清晰的、无歧义的并且是准确定义的。

9.6.2 组件分级

本族中的组件是基于安全功能要求是否被陈述,或是否由 TOE 安全目的得出而进行分级的。

9.6.3 APE_REQ.1 陈述的安全要求

依赖关系:APE_ECD.1 扩展组件的定义。

9.6.3.1 开发者行为元素

9.6.3.1.1 APE_REQ.1.1D

开发者应提供安全要求的陈述。

9.6.3.1.2 APE_REQ.1.2D

开发者应提供安全要求的基本原理。

9.6.3.2 内容和形式元素

9.6.3.2.1 APE_REQ.1.1C

安全要求陈述应描述安全功能要求和安全保障要求。

9.6.3.2.2 APE_REQ.1.2C

应对安全功能要求和安全保障要求中使用的所有主体、客体、操作、安全属性、外部实体及其他术语进行定义。

9.6.3.2.3 APE_REQ.1.3C

安全要求的陈述应对安全要求的所有操作进行标识。

9.6.3.2.4 APE_REQ.1.4C

所有操作应被正确执行。

9.6.3.2.5 APE_REQ.1.5C

安全要求的依赖关系应被满足,或者安全要求原理应论证某个依赖关系不需要满足。

9.6.3.2.6 APE_REQ.1.6C

安全要求的陈述应是内在一致的。

9.6.3.3 评估者行为元素

9.6.3.3.1 APE_REQ.1.1E

评估者应确认所提供的信息满足证据的内容和形式的要求。

9.6.4 APE_REQ.2 推导出的安全要求

依赖关系: APE_OBJ.2 安全目的
APE_ECD.1 扩展组件的定义

9.6.4.1 开发者行为元素

9.6.4.1.1 APE_REQ.2.1D

开发者应提供安全要求的陈述。

9.6.4.1.2 APE_REQ.2.2D

开发者应提供安全要求的基本原理。

9.6.4.2 内容和形式元素

9.6.4.2.1 APE_REQ.2.1C

安全要求的陈述应描述安全功能要求和安全保障要求。

9.6.4.2.2 APE_REQ.2.2C

应对安全功能要求和安全保障要求中使用的所有主体、客体、操作、安全属性、外部实体及其他术语进行定义。

9.6.4.2.3 APE_REQ.2.3C

安全要求的陈述应对安全要求的所有操作进行标识。

9.6.4.2.4 APE_REQ.2.4C

所有操作应被正确地执行。

9.6.4.2.5 APE_REQ.2.5C

安全要求的依赖关系应被满足,或者安全要求原理应论证某个依赖关系不需要满足。

9.6.4.2.6 APE_REQ.2.6C

安全要求的基本原理应描述每一个安全功能要求可追溯对应的 TOE 安全目的。

9.6.4.2.7 APE_REQ.2.7C

安全要求基本原理应证实安全功能要求可满足所有的 TOE 安全目的。

9.6.4.2.8 APE_REQ.2.8C

安全要求基本原理应说明选择安全保障要求的理由。

9.6.4.2.9 APE_REQ.2.9C

安全要求的陈述应是内部一致的。

9.6.4.3 评估者行为元素

9.6.4.3.1 APE_REQ.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10 ASE 类:安全目标评估

ST 评估要求证实 ST 是合理和内在一致的,如果 ST 是基于一个或多个 PP 或者包,那么 ST 必须是 PP 或包的一个正确的实例化。ST 作为 TOE 评估的基础,这些属性是必需的。

本章应该结合 ISO/IEC 15408-1 中的附录 A、B、C 一起使用,因为这些附录阐明了本章的一些概念,并提供了很多例子。

图 9 给出了本类中的族及族中各组件之间的层次关系。

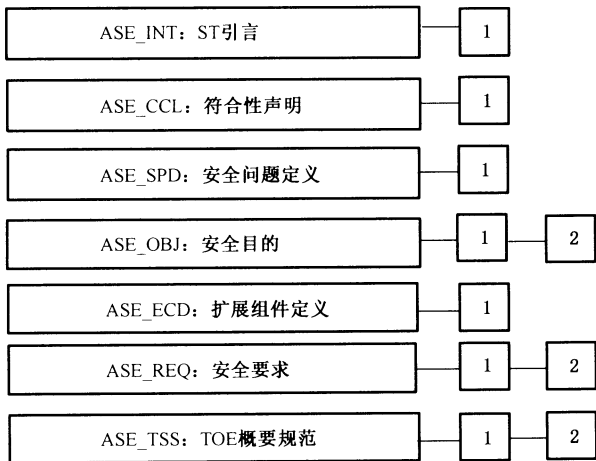


图 9 安全目标评估类分解

10.1 ST 引言(ASE_INT)

10.1.1 目的

本族的目的以叙述性语言从三个抽象层面上描述 TOE:TOE 参照号、TOE 概述和 TOE 描述。

对 ST 引言的评估需要证实 ST 和 TOE 被正确标识,TOE 的三层抽象方式描述正确,并且这三方面的描述相互一致。

10.1.2 ASE_INT.1 ST 引言

依赖关系:无。

10.1.2.1 开发者行为元素

10.1.2.1.1 ASE_INT.1.1D

开发者应提供 ST 引言。

10.1.2.2 内容和形式元素

10.1.2.2.1 ASE_INT.1.1C

ST 引言应包含 ST 参照号,TOE 参照号,TOE 概述和 TOE 描述。

10.1.2.2.2 ASE_INT.1.2C

ST 参照号应唯一标识 ST。

10.1.2.2.3 ASE_INT.1.3C

TOE 参照号应标识 TOE。

10.1.2.2.4 ASE_INT.1.4C

TOE 概述应概括 TOE 的用法及其主要安全特性。

10.1.2.2.5 ASE_INT.1.5C

TOE 概述应标识 TOE 类型。

10.1.2.2.6 ASE_INT.1.6C

TOE 概述应标识任何 TOE 要求的非 TOE 范围内的硬件/软件/固件。

10.1.2.2.7 ASE_INT.1.7C

TOE 描述应描述 TOE 的物理范围。

10.1.2.2.8 ASE_INT.1.8C

TOE 描述应描述 TOE 的逻辑范围。

10.1.2.3 评估者行为元素

10.1.2.3.1 ASE_INT.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.1.2.3.2 ASE_INT.1.2E

评估者应确认 TOE 参考、TOE 概述和 TOE 描述是相互一致的。

10.2 符合性声明(ASE_CCL)

10.2.1 目的

本族的目的确定符合性声明的正确性。另外,本族详细说明了 ST 如何声明与 PP 的符合性。

10.2.2 ASE_CCL.1 符合性声明

依赖关系:ASE_INT.1 ST 引言
ASE_ECD.1 扩展组件定义
ASE_REQ.1 陈述性的安全要求

10.2.2.1 开发者行为元素

10.2.2.1.1 ASE_CCL.1.1D

开发者应提供符合性声明。

10.2.2.1.2 ASE_CCL.1.2D

开发者应提供符合性声明的基本原理。

10.2.2.2 内容和形式元素

10.2.2.2.1 ASE_CCL.1.1C

ST 的符合性声明应包含 ISO/IEC 15408 符合性声明,标识出 ST 和 TOE 声明符合性遵从的 ISO/IEC 15408 的版本。

10.2.2.2.2 ASE_CCL.1.2C

ST 的符合性声明应描述 ST 与 ISO/IEC 15408-2 的符合性,无论是与 ISO/IEC 15408-2 相符或是与 ISO/IEC 15408-2 的扩展部分相符。

10.2.2.2.3 ASE_CCL.1.3C

ISO/IEC 15408 符合性声明应描述 ST 与本部分的符合性,无论是与本部分相符或是与本部分的扩展部分相符。

10.2.2.2.4 ASE_CCL.1.4C

符合性声明应与扩展组件定义是相符的。

10.2.2.2.5 ASE_CCL.1.5C

符合性声明应标识 ST 声明遵从的所有 PP 和安全要求包。

10.2.2.2.6 ASE_CCL.1.6C

符合性声明应描述 ST 和包的符合性,无论是与包的相符或是与扩展包相符。

10.2.2.2.7 ASE_CCL.1.7C

符合性声明的基本原理应证实 TOE 类型与符合性声明所遵从的 PP 中的 TOE 类型是相符的。

10.2.2.2.8 ASE_CCL.1.8C

符合性声明的基本原理应证实安全问题定义的陈述与符合性声明所遵从的 PP 中的安全问题定义陈述是相符的。

10.2.2.2.9 ASE_CCL.1.9C

符合性声明的基本原理应证实安全目的陈述与符合性声明所遵从的 PP 中的安全目的陈述是相符的。

10.2.2.2.10 ASE_CCL.1.10C

符合性声明的基本原理应证实安全要求的陈述与符合性声明所遵从的 PP 中的安全要求的陈述是相符的。

10.2.2.3 评估者行为元素

10.2.2.3.1 ASE_CCL.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.3 安全问题定义(ASE_SPD)

10.3.1 目的

ST 的本部分定义 TOE 及其运行环境所负责处理的安全问题。

安全问题定义评估要求证实 TOE 及其运行环境所负责处理的安全问题被明确地定义。

10.3.2 ASE_SPD.1 安全问题定义

依赖关系:无。

10.3.2.1 开发者行为元素

10.3.2.1.1 ASE_SPD.1.1D

开发者应提供安全问题定义。

10.3.2.2 内容和形式元素

10.3.2.2.1 ASE_SPD.1.1C

安全问题定义应描述威胁。

10.3.2.2.2 ASE_SPD.1.2C

所有的威胁都应根据威胁主体、资产和敌对行为进行描述。

10.3.2.2.3 ASE_SPD.1.3C

安全问题定义应描述组织安全策略。

10.3.2.2.4 ASE_SPD.1.4C

安全问题定义应描述 TOE 运行环境的相关假设。

10.3.2.3 评估者行为元素

10.3.2.3.1 ASE_SPD.1.1E

评估者应确认所提供的信息满足证据的内容和形式的的所有要求。

10.4 安全目的(ASE_OBJ)

10.4.1 目的

安全目的是对 ASE_SPD“安全问题定义”族定义的安全问题预期反应的简明陈述。

安全目的评估要求证实安全目的充分而且完备地处理了定义的安全问题,并证实 TOE 及其运行环境的安全问题被明确的区分和定义。

10.4.2 组件分级

本族中的组件是仅规定使用运行环境安全目的还是也使用 TOE 安全目的来进行分级。

10.4.3 ASE_OBJ.1 运行环境安全目的

依赖关系:无。

10.4.3.1 开发者行为元素

10.4.3.1.1 ASE_OBJ.1.1D

开发者应提供安全目的的陈述。

10.4.3.2 内容和形式元素

10.4.3.2.1 ASE_OBJ.1.1C

安全目的的陈述应描述运行环境安全目的。

10.4.3.3 评估者行为元素

10.4.3.3.1 ASE_OBJ.1.1E

评估者应确认所提供的信息满足证据的内容和形式的的所有要求。

10.4.4 ASE_OBJ.2 安全目的

依赖关系:ASE_SPD.1 安全问题定义。

10.4.4.1 开发者行为元素

10.4.4.1.1 ASE_OBJ.2.1D

开发者应提供安全目的的陈述。

10.4.4.1.2 ASE_OBJ.2.2D

开发者应提供安全目的的基本原理。

10.4.4.2 内容和形式元素

10.4.4.2.1 ASE_OBJ.2.1C

安全目的的陈述应描述 TOE 的安全目的和运行环境安全目的。

10.4.4.2.2 ASE_OBJ.2.2C

安全目的基本原理应追溯到 TOE 的每一个安全目的,以便于能追溯到安全目的所对抗的威胁及安全目的实施的组织安全策略。

10.4.4.2.3 ASE_OBJ.2.3C

安全目的基本原理应追溯到运行环境的每一个安全目的,以便于能追溯到安全目的所对抗的威胁、安全目的实施的组织安全策略和安全目的支持的假设。

10.4.4.2.4 ASE_OBJ.2.4C

安全目的基本原理应证实安全目的能抵抗所有威胁。

10.4.4.2.5 ASE_OBJ.2.5C

安全目的基本原理应证实安全目的执行所有组织安全策略。

10.4.4.2.6 ASE_OBJ.2.6C

安全目的基本原理应证实运行环境安全目的支持所有的假设。

10.4.4.3 评估者行为元素

10.4.4.3.1 ASE_OBJ.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.5 扩展组件定义(ASE_ECD)

10.5.1 目的

扩展安全要求不是基于 ISO/IEC 15408 第 2 部分或第 3 部分的组件要求,而是基于 ST 作者定义的扩展组件。

对扩展组件定义的评估需要确定这些组件是明确的、没有歧义的并且是必要的,如:他们可能没有使用 ISO/IEC 15408 第 2 部分或第 3 部分已存在的组件进行清晰的描述。

10.5.2 ASE_ECD.1 扩展组件定义

依赖关系:无。

10.5.2.1 11.5.2.1 开发者行为元素

10.5.2.1.1 ASE_ECD.1.1D

开发者应提供安全要求的陈述。

10.5.2.1.2 ASE_ECD.1.2D

开发者应提供扩展组件的定义。

10.5.2.2 内容和形式元素

10.5.2.2.1 ASE_ECD.1.1C

安全要求陈述应标识所有扩展的安全要求。

10.5.2.2.2 ASE_ECD.1.2C

扩展组件定义应为每一个扩展的安全要求定义一个扩展的组件。

10.5.2.2.3 ASE_ECD.1.3C

扩展组件定义应描述每个扩展的组件与已有组件、族和类的关联性。

10.5.2.2.4 ASE_ECD.1.4C

扩展组件定义应使用已有的组件、族、类和方法学作为陈述的模型。

10.5.2.2.5 ASE_ECD.1.5C

扩展组件应由可测量的和客观的元素组成,以便于证实这些元素之间的符合性或不符合性。

10.5.2.3 评估者行为元素

10.5.2.3.1 ASE_ECD.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.5.2.3.2 ASE_ECD.1.2E

评估者应确认扩展组件不能利用已经存在的组件明确的表达。

10.6 安全要求(ASE_REQ)

10.6.1 目的

安全功能要求(SFR)对 TOE 预期安全行为进行了清晰、无歧义且定义准确的描述。安全保障要求(SAR)对 TOE 获得保障而采取的预期活动进行了清晰、无歧义且规范的描述。

对安全要求的评估要求确保这些安全要求是清晰的、无歧义的并且是准确定义的。

10.6.2 组件分级

本族中的组件是基于安全要求是否被陈述而进行分级的。

10.6.3 ASE_REQ.1 陈述性的安全要求

依赖关系:ASE_ECD.1 扩展组件定义。

10.6.3.1 开发者行为元素

10.6.3.1.1 ASE_REQ.1.1D

开发者应提供安全要求的陈述。

10.6.3.1.2 ASE_REQ.1.2D

开发者应提供安全要求的基本原理。

10.6.3.2 内容和形式元素

10.6.3.2.1 ASE_REQ.1.1C

安全要求的陈述应描述安全功能要求和安全保障要求。

10.6.3.2.2 ASE_REQ.1.2C

应对安全功能要求和安全保障要求中使用的所有主体、客体、操作、安全属性、外部实体及其他术语进行定义。

10.6.3.2.3 ASE_REQ.1.3C

安全要求的陈述应对安全要求的所有操作进行标识。

10.6.3.2.4 ASE_REQ.1.4C

所有操作应被正确执行。

10.6.3.2.5 ASE_REQ.1.5C

应满足安全要求间的依赖关系,或者安全要求基本原理应论证不需要满足某个依赖关系。

10.6.3.2.6 ASE_REQ.1.6C

安全要求的陈述应是内在一致的。

10.6.3.3 评估者行为元素

10.6.3.3.1 ASE_REQ.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.6.4 ASE_REQ.2 推导出的安全要求

依赖关系:ASE_OBJ.2 安全目的;
ASE_ECD.1 扩展组件定义。

10.6.4.1 开发者行为元素

10.6.4.1.1 ASE_REQ.2.1D

开发者应提供安全要求的陈述。

10.6.4.1.2 ASE_REQ.2.2D

开发者应提供安全要求的基本原理。

10.6.4.2 内容和形式元素

10.6.4.2.1 ASE_REQ.2.1C

安全要求的陈述应描述安全功能要求和安全保障要求。

10.6.4.2.2 ASE_REQ.2.2C

应对安全功能要求和安全保障要求中使用的所有主体、客体、操作、安全属性、外部实体及其他术语

进行定义。

10.6.4.2.3 ASE_REQ.2.3C

安全要求的陈述应对安全要求的所有操作进行标识。

10.6.4.2.4 ASE_REQ.2.4C

所有操作应被正确地执行。

10.6.4.2.5 ASE_REQ.2.5C

应满足安全要求间的依赖关系,或者安全要求基本原理应论证不需要满足某个依赖关系。

10.6.4.2.6 ASE_REQ.2.6C

安全要求基本原理应描述每一个安全功能要求可追溯至对应的 TOE 安全目的。

10.6.4.2.7 ASE_REQ.2.7C

安全要求基本原理应证实安全功能要求可满足所有的 TOE 安全目的。

10.6.4.2.8 ASE_REQ.2.8C

安全要求基本原理应说明选择安全保障要求的理由。

10.6.4.2.9 ASE_REQ.2.9C

安全要求的陈述应是内在一致的。

10.6.4.3 评估者行为元素

10.6.4.3.1 ASE_REQ.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.7 TOE 概要规范(ASE_TSS)

10.7.1 目的

TOE 概要规范能够使评估者和潜在消费者对 TOE 是如何实现的有一个全面的理解。

TOE 概要规范的评估对于确定 TOE 是如何实现以下方面的描述是否充分是必要的:

- 满足 TOE 的安全功能要求;
- 保护 TOE 自身防止被干扰、逻辑篡改和被旁路。

而且,对于确定 TOE 概要规范和 TOE 的其他叙述性描述是否一致也是必要的。

10.7.2 组件分级

本族中的组件是基于 TOE 概要规范仅需描述 TOE 如何满足安全功能要求,或者 TOE 概要规范也需要描述 TOE 是怎样保护其自身,以防止被逻辑篡改和被旁路而进行分级的。这些附加的描述适用于对 TOE 安全架构有特殊要求的特定情形。

10.7.3 ASE_TSS.1 TOE 概要规范

依赖关系:ASE_INT.1 ST 引言;

ASE_REQ.1 陈述性的安全要求；
ADV_FSP.1 基本功能规范。

10.7.3.1 开发者行为元素

10.7.3.1.1 ASE_TSS.1.1D

开发者应提供 TOE 概要规范。

10.7.3.2 内容和形式元素

10.7.3.2.1 ASE_TSS.1.1C

TOE 概要规范应描述 TOE 是如何满足每一项安全功能要求的。

10.7.3.3 评估者行为元素

10.7.3.3.1 ASE_TSS.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.7.3.3.2 ASE_TSS.1.2E

评估者应确认 TOE 概要规范与 TOE 概述、TOE 描述是一致的。

10.7.4 ASE_TSS.2 具有结构设计概述的 TOE 概要规范

依赖关系:ASE_INT.1 ST 引言；
ASE_REQ.1 陈述性的安全要求；
ADV_ARC.1 安全架构描述。

10.7.4.1 开发者行为元素

10.7.4.1.1 ASE_TSS.2.1D

开发者应提供 TOE 概要规范。

10.7.4.2 内容和形式元素

10.7.4.2.1 ASE_TSS.2.1C

TOE 概要规范应描述 TOE 是如何满足每一项安全功能要求的。

10.7.4.2.2 ASE_TSS.2.2C

TOE 概要规范应描述 TOE 是如何保护自身以防止被干扰和被逻辑篡改。

10.7.4.2.3 ASE_TSS.2.3C

TOE 概要规范应描述 TOE 是如何保护自身以防止被旁路。

10.7.4.3 评估者行为元素

10.7.4.3.1 ASE_TSS.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

10.7.4.3.2 ASE_TSS.2.2E

评估者应确认 TOE 概要规范与 TOE 概述、TOE 描述是一致的。

11 ADV 类:开发

开发类的安全要求提供了 TOE 的相关信息。从这些信息中所获得的知识可作为执行 TOE 脆弱性分析和测试的基础,正如 AVA 和 ATE 类中所描述的那样。

开发类共包括六个族,这些族在不同级别和不同抽象形式上组织和表示 TOE 安全功能。这些族包含以下要求:

- 安全功能要求设计和实现在不同抽象级别下的描述要求(ADV_FSP、ADV_TDS、ADV_IMP);
- 域分离、TSF 自保护和安全功能的不可旁路性等面向架构特性的描述要求(ADV_ARC);
- 安全策略模型以及安全策略模型和功能规范之间对应性映射要求(ADV_SPM);
- TSF 内部结构的要求,包括诸如模块化、层次化以及复杂度最小化等各个方面(ADV_INT)。

当对 TOE 安全功能进行文档化描述时,需要证实两个属性。第一个属性是安全功能实现正确,也就是说它能按照规定执行。第二个属性是不能以破坏或者旁路安全功能的方式使用 TOE,这是比较难以证实的。这两种属性需要不同程度的分析方法,因此 ADV 类中的族是按照支持这些不同方法组织起来的。功能规范族(ADV_FSP)、TOE 设计族(ADV_TDS)、实现表示族(ADV_IMP)和安全策略模型族(ADV_SPM)处理第一个属性:安全功能规范。安全架构族(ADV_ARC)和 TSF 内部族(ADV_INT)处理第二个属性:TOE 设计规范证实了安全功能不能被破坏或旁路。需要注意的是这两个属性都必须实现:这些属性满足的越充分,TOE 就越可信。这些族中的组件被设计成随着组件层次的增加能获得的保障也更多。

保障族的一个范型即是设计分解中的第一个属性为对象。最高级别上,是 TSF 接口的功能规范(向 TSF 请求服务以及做出响应),将 TSF 分解成较小的部件(由期望的保障级和 TOE 的复杂度决定),描述 TSF 怎样实现它的功能(与保障级别相当的详细程度),并且说明 TSF 是如何实现的。也可能给出安全行为的形式化模型。分解的所有级别用于确定所有其他级别的完整性和精确性,以保证这些级别是互相支撑的。关于各种 TSF 表示的要求都被分散到不同的族中,以允许 PP/ST 的作者指明需要哪一个 TSF 表示。选择的级别将表明所期望/获得的保障。

图 10 表明了 ADV 类的各种 TSF 表示间的关系,也表明了它与其他类之间的关系。如图 10 所示,APE 类和 ASE 类定义了安全功能要求和 TOE 安全目的之间对应性要求。ASE 类也定义了安全目的和安全功能要求之间对应性要求,并描述了用来解释 TOE 是怎样满足安全功能要求的 TOE 概要规范。实际上,ALC_CMC.5.2E 子活动内容就是包括验证 ATE 和 AVA 类测试过的 TSF 实际上都被 ADV 描述过。

ADV 类中定义了图 10 所示的所有其他对应性要求。安全策略模型族(ADV_SPM)定义了已选安全功能要求的形式化模型的相关要求,并且提供功能规范和形式化模型的对应性。与 TSF 描述相关的保障族(即功能规范(ADV_FSP)、TOE 设计(ADV_TDS)和实现表示(ADV_IMP)等族)定义了安全功能描述与安全功能要求相关联的一些要求。所有的分解必须与其他分解正确对应(即相互支持);在组件的最后一个 C 元素中,开发者给出追溯关系。通过分析每一级别的分解来获得关于该要素的保障,当进行特定级别的分解分析时可参照其他级别的分解(以递归的方式);作为第二个 E 元素的一部分,评估者应验证此一致性。对这些级别的分解的理解形成了功能测试和穿透性测试的基础。

图 10 中没有标出 ADV_INT 族,因为它与 TSF 的内部结构相关,并且仅与 TSF 描述的细化过程间接相关。在图中也没有标出 ADV_ARC 族,因为它与 TSF 的结构可靠性相关,与 TSF 描述无关。ADV_INT 和 ADV_ARC 都与 TOE 不能回避或破坏的安全功能的属性分析相关。

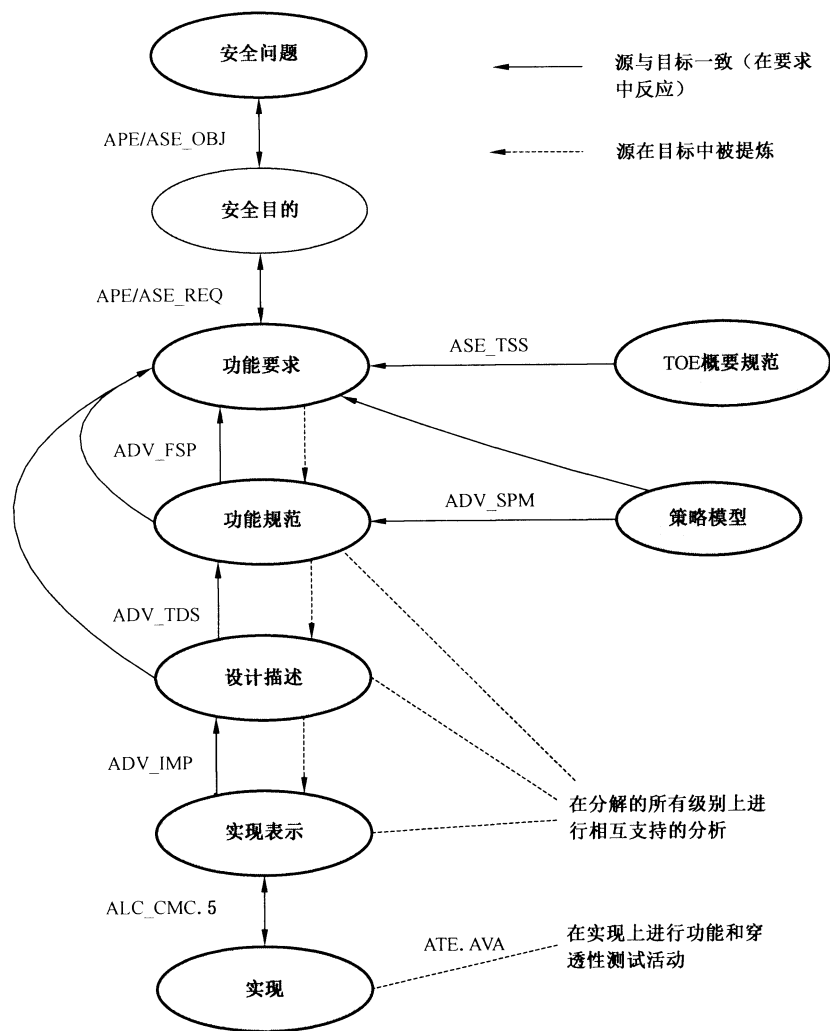


图 10 ADV 各族之间及与其他族之间的关系

TOE 安全功能(TSF)包括 SFR-执行所必须依赖的 TOE 所有部分。TSF 既包括一些直接执行安全功能要求的功能及间接执行安全功能要求但会间接影响安全功能要求,违背安全功能要求的功能。这包括在启动时被调用以将 TSF 置于初始安全状态的 TOE 相关组成部分。

在 ADV 族组件的开发中用到了几个重要的概念。这里仅简单介绍了这些概念,在族的操作说明中对他们做了更详细的解释。

一个重要的观点是,获取的信息越多,安全功能在 1)正确实现,2)不被破坏,3)不可旁路性方面所获得的保障就越多。获取信息的手段可以是验证文档的正确性以及文档之间的一致性,也可以通过提供信息确保测试活动(功能测试和穿透性测试)是全面、充分的。信息的多少体现在族中各组件的层次上。一般说来,组件的所处的层次基于所提供的信息量的多少(这点将在后面进行分析)。

虽然不是所有 TOE,但一般情况下 TSF 十分复杂,TSF 的某些部分比其他部分值得做更严格的检测。但是对于如何确定那些部分往往具有主观性,因此术语和组件被定义成随着保障级别的增加,确定 TSF 的哪些部分需要详细测试的职责由开发者转换为评估者。为了便于描述此概念,特引入以下术语。应当注意的是在类的各族中,当描述安全功能要求相关的 TOE 部分时,往往要用到这些术语(即,包含于功能规范(ADV_FSP),TOE 设计(ADV_TDS)和表示实现(ADV_IMP)族中的各元素和工作单元)。而对于被应用于其他族中的一般概念(比其他部分更感兴趣的 TOE 组成部分),为了获得必需的保障,描述的规则会不同。

TSF 的所有部分都与安全相关,这意味着它们必须保护被描述为与域分离、不可旁路性相关的 TOE 安全性要求。安全相关性一方面是指部分 TSF 执行安全要求的程度。既然在执行安全要求中,TOE 的不同部分发挥着不同的作用(或者根本没有明显的作用),这就造成了安全功能要求相关性的连续统一体;该连续统一体的末端被称为强制实施 SFR-执行(SFR-enforcing)的 TOE 部分。这样的部分在 TOE 中扮演着实现安全功能要求的指挥角色。这些安全功能要求参照 ST 中安全功能要求的全部功能性。应当注意的是,SFR-执行部分所发挥功能性作用的大小是很难量化表达的。例如,在强制访问控制(DAC)机制的实现中,SFR-执行在狭义上可能是实际上由几行代码实现了依据客体特性对主体特性的检查。在广义上可能是包含这几行代码的软件实体(例如 C 函数)。再广义的观点将包括对 C 函数的调用,因为他们负责决定执行属性检查后返回值的情况。再广义一些的观点是包括 C 函数的调用树(或者该开发语言使用的等同的设计)上的任何代码(例如用首次匹配算法实现的分类访问控制列表实体的一类函数)。某些情况下,有些部分并没有强制要求执行安全策略,而是更多的发挥着支持的作用,这部分定义为 SFR 支撑(SFR-supporting)部分。

SFR-支撑功能的一个特点是它可通过无误的操作保持安全功能要求的正确执行。SFR-执行功能可以依赖 SFR-支撑功能,但是这种依赖通常体现在功能层面上。例如,内存管理、缓冲区管理等。按安全相关性进行连续性划分的下面一层是称为 SFR-无关(SFR-non-interfering)的功能。这种功能在实现安全功能要求方面不起作用,由于其执行环境的原因,这种功能也可能是 TSF 的一部分。例如,操作系统中运行在一个特权硬件模式下的代码就是 TSF 的一部分,因为它一旦被侵害(被恶意代码替代),就有可能因为运行在特权硬件模式而妨碍安全功能要求的正确运行。一些在内核模式中用于提升运算速度的数学浮点运算操作就是 SFR-无关功能的具体例子。

在域分离、自保护和不可旁路的特性基础之上,架构族(安全架构(ADV_ARC))用于对 TOE 提出要求并进行分析。这些特性与安全功能要求相关,如果没有这些特性,将会导致安全功能要求实现机制出现问题。由于本质和分析要求的根本不同,与这些特性相关的功能性和设计不在上面描述连续统一体中进行考虑,而是单独处理。

安全功能要求的实现(包括 SFR-执行和 SFR-支撑功能)与相对基础的 TOE 的基本安全特性的实现(包括初始化、自保护和不可旁路性所涉及的部分)在分析方面的不同之处在于,SFR 相关的功能性在一定程度上是直接可见并且相对而言易于测试的,而后者则需要在一个更宽泛的功能性集合中变换角度分析。而且,对这些特性分析的深度将依赖 TOE 的设计而不同。ADV 中有一个单独的族(安全架构 ADV_ARC)用来描述对初始化、自保护和不可旁路性等要求的分析,同时其他族用于对 SFR-支撑的功能性的分析。

即使需要不同级别下的抽象描述,也没有将每一个 TSF 描述为一个独立文档的绝对必要。确实有时一个文档需要多种 TSF 描述,因为每一种 TSF 描述的信息都是需要的,而不仅仅是最终的文档形式。当多种 TSF 描述被融合到一篇文档中的情况下,开发者应当指出文档的哪一部分符合哪些要求。

这个类规定了三种类型的规范方式:非形式化、半形式化和形式化。功能规范和 TOE 设计文档只以非形式化或者半形式化方式进行描述。半形式化方式描述的文档与非形式化文档相比降低了文档的歧义性。除半形式化的描述之外,可能也需要形式化的描述;用多种方式对 TSF 进行描述可以增加对 TSF 完全准确描述的保障。

非形式化规范就是像白话文一样用自然语言来书写。在这里使用自然语言,意味着以任何普通口头语言(如汉语、西班牙语、德语、法语、英语、荷兰语)来沟通。非形式化规范除了使用语言本身需要的常规约定(如文法和句法)外,不受任何标记性的或特殊的限制。虽然没有标记限制,非形式化规范也要求为上下文中的术语进行定义,除非作为常规用法该术语已被认可。

半形式化和非形式化文档的区别仅仅在于格式上或者表达方式上:半形式化的符号系统包括明确的术语表、标准化的描述格式等。半形式化规范用标准的描述模板来编写。如果用自然语言编写,描述应始终使用术语。描述也可以使用更结构化的语言或者图表(例如数据流图、状态转换图、实体关系图、

数据结构图以及过程或程序结构图)的形式。无论使用图表还是自然语言,在描述中必须使用一套约定。术语表对需要以准确固定的方式使用的词汇进行了明确标识;同样,使用标准化的格式意味着已特别注意到了该如何以最清晰的方式系统地准备文档。应注意 TSF 中存在根本性差异的部分可能有不同的半形式化符号约定和描述风格(只要不同的“半形式化符号”的数量不是很多);这种情况仍然遵照半形式化描述的观念。

形式化规范就是用基于公认的数学概念的符号系统来书写,并且通常附有支持性的解释(非形式化)语句。这些数学概念被用来定义符号的语法和语义,以及支持逻辑推理的证明规则。支持形式化符号的语法和语义规则应该定义如何明确地识别其结构并确定其含义。这里必须有证据表明不可能有产生矛盾之处,并且所有用于符号支持的规则都必须被定义或者引用。

图 11 给出了该类中包含的族以及族内组件的层次结构。

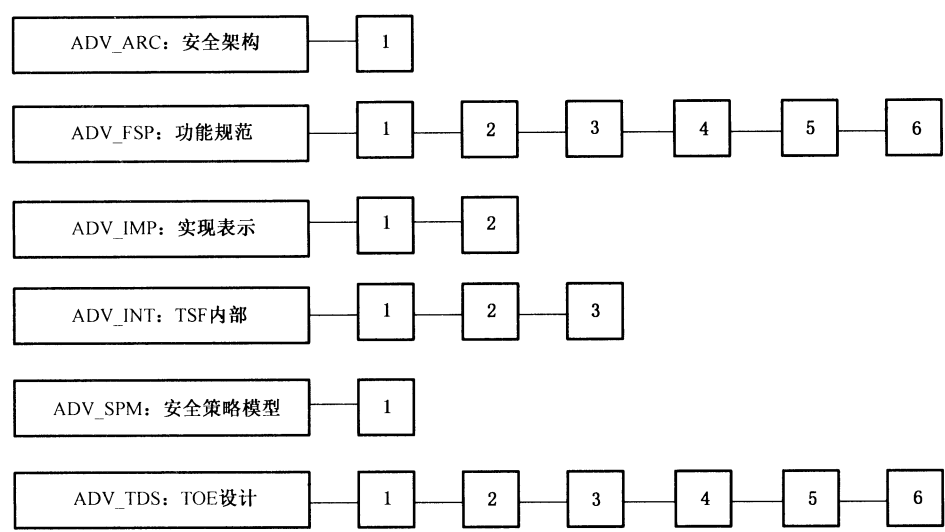


图 11 ADV:开发类分解

11.1 安全架构(ADV_ARC)

11.1.1 目的

本族的目的是让开发者提供对 TSF 安全架构的描述。允许额外提供的其他 TSF 证据对这些信息进行分析,这些信息将确保 TSF 达到期望的属性。安全架构描述支持隐含的声明,即通过检测 TSF 所能获得的 TOE 安全分析;如果没有合理的架构,则必须检查 TOE 全部的功能特性。

11.1.2 组件分级

本族只包含一个组件。

11.1.3 应用注释

自保护、域分离和不可旁路与 ISO/IEC 15408-2 安全功能要求描述的安全功能特性是有区别的,因为在 TSF 中自保护和不可旁路基本上没有直接可观察的接口。确切地说,他们是通过 TOE 和 TSF 设计而获得的 TSF 属性,并且通过正确实现该设计而得以执行。

本族中的方法是开发者设计和提供具有上述特性的 TSF,并且提供这些描述 TSF 特性的证据(以文档的形式)。提供的这种解释与 TOE 的设计文档中的 TOE 的 SFR-执行元素的描述有相同的详细级别。评估者有责任考察证据,再加上交付 TOE 和 TSF 的其他证据,以确定这些特性得到实现。

实施安全功能要求(在功能规范(ADV_FSP)和 TOE 设计(ADV_TDS)中)的安全功能规范不一定会描述用于实施自我保护和不可旁路(如内存管理机制)的机制。因此,需要提供资料来保障这些正在实现的要求与体现在 ADV_FSP 和 ADV_TDS 中的 TSF 的分解设计相比更好。但这并不是说这个组件所要求的安全架构描述不能参考或者使用分解设计材料,而是说分解设计文档中的大量细节描述不能被直接用于安全架构描述文档。

对架构可靠性的描述可以认为是开发者的脆弱性分析,其中论证了为什么 TSF 是可靠的并且满足所有安全功能要求。可靠性是通过特定的安全机制实现的,而且这些将作为测试深度(ATE_DPT)要求的一部分进行测试;其中可靠性只有通过架构才能得以实现,并且其行为作为 AVA(脆弱性评估要求)的一部分来进行测试。

本族由安全架构描述要求组成,描述了自保护、域分离和不可旁路的原理,包括由用于 TSF 初始化的那部分 TOE 所支持的原理的描述。

在附录 A.1,ADV_ARC:安全架构的补充材料中,可以找到安全架构的自保护、域分离和不可旁路的特性的补充说明。

11.1.4 ADV_ARC.1 安全架构描述

依赖关系:ADV_FSP.1 基本功能规范;
ADV_TDS.1 基础设计。

11.1.4.1 开发者行为元素

11.1.4.1.1 ADV_ARC.1.1D

开发者应设计并实现 TOE,确保 TSF 的安全特性不可旁路。

11.1.4.1.2 ADV_ARC.1.2D

开发者应设计并实现 TSF,以防止不可信主体的破坏。

11.1.4.1.3 ADV_ARC.1.3D

开发者应提供 TSF 安全架构的描述。

11.1.4.2 内容和形式元素

11.1.4.2.1 ADV_ARC.1.1C

安全架构的描述应与在 TOE 设计文档中对 SFR-执行的抽象描述的级别一致。

11.1.4.2.2 ADV_ARC.1.2C

安全架构的描述应描述与安全功能要求一致的 TSF 安全域。

11.1.4.2.3 ADV_ARC.1.3C

安全架构的描述应描述 TSF 初始化过程为何是安全的。

11.1.4.2.4 ADV_ARC.1.4C

安全架构的描述应证实 TSF 可防止被破坏。

11.1.4.2.5 ADV_ARC.1.5C

安全架构的描述应证实 TSF 可防止 SFR-执行的功能被旁路。

11.1.4.3 评估者行为元素

11.1.4.3.1 ADV_ARC.1.1E

评估者应确认提供的信息符合证据的内容和形式要求。

11.2 功能规范(ADV_FSP)

11.2.1 目的

本族提出了针对功能规范的要求,该规范描述了 TSF 接口(TSFI)。TSFI 包括所有的通过外部实体(或者位于 TSF 之外 TOE 内部的主体)向 TSF 提供数据、接收来自于 TSF 的数据并且调用 TSF 的服务的方法。它并不描述 TSF 如何处理那些服务请求,也不描述当 TSF 调用运行环境的服务时的通信;TOE 设计族(ADV_TDS)和依赖模块的依赖性族(ACO_REL)将分别描述这些内容。

本族通过使评估者理解 TSF 如何满足安全功能要求提供直接的保障。它也作为其他保障族和保障类的输入,提供间接的保障:

- ADV_ARC,对 TSFI 的描述可以用来更好地理解 TSF 如何抵御侵害(自保护或域分离的破坏)以及/被旁路;
- ATE,对 TSFI 的描述是开发者和评估者测试的重要输入;
- AVA,通过对 TSFI 的描述用来查找脆弱性。

11.2.2 组件分级

本族中的组件是基于所要求的对 TSFI 描述的详细程度和形式化程度而分级的。

11.2.3 应用注释

一旦确定了 TSFI(参见 A.2.1,确定 TSFI 的指导以及确定 TSFI 的例子),就对其进行描述。在低级的组件中,开发者将他们的文档(评估者将他们的分析)更多地关注于 TOE 的安全相关方面。基于所获得的服务定义了三类 TSFI,安全功能要求声明为:

- 如果通过一个接口可获得的服务能够被追溯到 TSF 的一个安全功能要求,则该接口被称为 SFR-执行。注意该接口可能有多种服务和结果,其中一些可能用于 SFR-执行,另一些可能不是。
- 如果是 SFR-执行功能所依赖的,且只需正确运行以保持 TOE 安全策略的那些服务接口(或通过接口相关的服务),则称其为 SFR-支撑。
- SFR-执行功能不依赖的服务接口称为 SFR-无关。

应注意 SFR-支撑和 SFR-无关接口必须没有 SFR-执行的服务或者结果。相反,SFR-执行接口可能有 SFR-支撑服务(例如,设置系统时钟的能力可能是一个接口的 SFR-执行服务,但是如果同一个接口用作显示系统日期,那么该服务可能只是 SFR-支撑)。纯 SFR-支撑接口的例子是既被用户使用又被为用户利益而运行的一部分 TSF 使用的系统调用接口。

TSFI 的信息获得的越多,就越能确保对这些接口的分类和分析的正确性。在最低级别中,评估者为了以有效的方式进行评估确定,SFR-无关接口需要的信息是最不必要的。在更高的级别中,由于获得了更多的信息,评估者对设计就越有信心。

命名这些接口(SFR-执行、SFR-支撑和 SFR-无关)以及(在更低级别的保障组件上)为每种接口规定不同要求的目的是为了近似地给出应重点分析之处以及分析的证据。如果开发者的 TSF 接口文档按 SFR-执行接口要求的详细程度描述了所有接口(也就是说,如果文档超越了要求),那么就开发者就不需要创建新的证据来满足要求。同样,因为这种接口命名仅仅是从要求的角度区分接口类型的一种

方法,所以开发者没有必要仅仅为了标识接口为 SFR-执行、SFR-支撑还是 SFR-无关而更新证据。这样命名方法的主要目的是让开发者在没有成熟的开发方法(和相关的人工产品,如详细接口和设计文档)的情况下就只提供必要的证据,而无需付出过多的代价。

本族中每个组件的最后一个 C 元素给出了安全功能要求与功能规范的直接联系,也就是说,指出了每个安全功能要求是通过哪些接口调用的。当 ST 中包含残余信息保护(FDP_RIP)这样的功能要求时,由于其功能未在 TSFI 中陈述、功能规范和/或用于识别安全功能要求的追溯应能标识这些安全功能要求;在细化分解过程中这些安全功能要求作用重大,将其包含在功能规范中有助于确保在这个过程中不会被遗漏。

11.2.3.1 接口细则

本要求定义了 TSFI 的细节的集合。本要求的目的是按照接口的目的、使用方法、参数、参数描述和错误信息来详细描述接口(以不同的详细程度)。

详细说明接口的意图(例如 GUI 命令、接收网络数据包、提供打印输出等)是对接口的一般目的进行高层的描述。

接口的使用方法描述如何使用接口。这种描述应当建立在接口的各种相互作用之上。例如,如果接口是个 UNIX shell 命令,那么 ls、mv 和 cp 将与该接口有相互作用。对于每个相互作用,使用方法描述相互作用是什么,既有该接口的行为(例如程序员调用 API,Windows 用户更改注册表的设置等),也有其他接口的行为(例如生成一条审计记录)。

参数是接口的输入输出,它控制着接口的行为。例如提供给 API 的参数;一个给定网络协议的数据包的不同字段;Windows 注册表中的个别的键值;通过芯片上一组管脚的信号;用于设置 ls 的标志等。参数被标识为一个表明它们是什么的简单的列表。

参数描述讲述这个参数的意图。例如,接口 foo(i)的可接受的参数描述可以是“参数 i 是一个表明当前登录到本系统的用户数量的整数”。像“参数 i 是一个整数”这样的描述就不可接受。

接口行为的描述讲述了该接口是做什么的。这比目的描述更详细,“目的”揭示为什么可以使用它,而“行为”揭示接口所做的每一件事。这些行为可能与安全功能要求相关或无关。如果接口行为与 SFR-无关可概括描述,意味着描述仅解释它确实不是安全功能要求相关的。

错误消息的描述可确定错误产生的条件,如消息是什么,以及错误代码的含义。TSF 生成一条错误消息,用于表示遇到的问题或者某些程度的不规律。本族要求涉及不同类型的错误消息:

- “直接”错误消息是通过特定 TSFI 调用的安全相关的响应。
- “间接”错误不能依赖于特定的 TSFI 调用,因为它是系统环境的情况(例如资源耗尽、连接中断等)。不是安全相关的错误消息也被认为是“间接”的。
- “残余”错误是所有其他错误,如那些代码中可能出现的错误。例如,使用用于检查逻辑上不会发生的情况(例如一系列“case”语句之后的最后一条“else”)的条件检查代码,将会生成一个笼统的错误消息;在运行的 TOE 中,不应看到这些错误消息。
- 在 A.2.3 中给出了功能规范的一个范例。

11.2.3.2 本族的组件

通过增强接口规范的完备性和准确性而提高保障,这就要求开发者提供本族不同等级的组件细节的文档。

ADV_FSP.1 基本功能规范,唯一的文档要求是刻画所有 TSFI,并对 SFR-执行和 SFR-支撑类型的 TSFI 进行高层描述。为了确保 TSF 的“重要”方面在 TSFI 中已被正确描述,要求开发者提供目的、使用方法、SFR-执行和 SFR-支撑的 TSFI 的参数。

ADV_FSP.2 安全执行功能规范,要求开发者提供所有 TSFI 的目的、使用方法、参数和参数描述。

另外,对于 SFR-执行 TSFI,开发者必须描述 SFR-执行行为和直接错误消息。

ADV_FSP.3 带完整摘要的功能规范,除了 ADV_FSP.2 要求的信息之外,开发者必须提供足够的关于 SFR-支撑和 SFR-无关行为的信息,用以表明它们不是 SFR-执行的。此外,开发者必须文档化由调用 SFR-执行的 TSFI 产生的所有直接错误消息。

ADV_FSP.4 完备的功能规范,所有 TSFI(无论是 SFR-执行、SFR-支撑还是 SFR-无关)必须以同等程度来描述,包括所有的直接错误消息。

ADV_FSP.5 附加错误信息的完备的半形式化功能规范,TSFI 描述也包括不是由 TSFI 调用引起的错误消息。

ADV_FSP.6 附加形式化描述的完备的半形式化功能规范,除了 ADV_FSP.5 要求的消息外,还包含对所有残余错误消息的描述。开发者也必须提供对 TSFI 的形式化描述。这为分析 TSFI 提供了另一个角度,可以暴露出内在矛盾或者不完善的规范。

11.2.4 ADV_FSP.1 基本功能规范

依赖关系:无。

11.2.4.1 开发者行为元素

11.2.4.1.1 ADV_FSP.1.1D

开发者应提供一个功能规范。

11.2.4.1.2 ADV_FSP.1.2D

开发者应提供功能规范到安全功能要求的追溯关系。

11.2.4.2 内容和形式元素

11.2.4.2.1 ADV_FSP.1.1C

功能规范应描述每个 SFR-执行和 SFR-支撑的 TSFI 的目的和使用方法。

11.2.4.2.2 ADV_FSP.1.2C

功能规范应识别每个 SFR-执行和 SFR-支撑的 TSFI 相关的所有参数。

11.2.4.2.3 ADV_FSP.1.3C

功能规范应提供暗含的 SFR-无关的接口分类的基本原理。

11.2.4.2.4 ADV_FSP.1.4C

功能规范应证实安全功能要求到 TSFI 的追溯。

11.2.4.3 评估者行为元素

11.2.4.3.1 ADV_FSP.1.1E

评估者应确认所提供的信息满足证据的内容和形式的有关要求。

11.2.4.3.2 ADV_FSP.1.2E

评估者应确定功能规范是安全功能要求的一个准确且完备的实例化。

11.2.5 ADV_FSP.2 安全执行功能规范

依赖关系:ADV_TDS.1 基础设计。

11.2.5.1 开发者行为元素

11.2.5.1.1 ADV_FSP.2.1D

开发者应提供一个功能规范。

11.2.5.1.2 ADV_FSP.2.2D

开发者应提供功能规范到安全功能要求的追溯关系。

11.2.5.2 证据的内容和形式元素

11.2.5.2.1 ADV_FSP.2.1C

功能规范应完整地描述 TSF。

11.2.5.2.2 ADV_FSP.2.2C

功能规范应描述所有的 TSFI 的目的和使用方法。

11.2.5.2.3 ADV_FSP.2.3C

功能规范应识别和描述每个 TSFI 相关的所有参数。

11.2.5.2.4 ADV_FSP.2.4C

对于每个 SFR-执行 TSFI,功能规范应描述 TSFI 相关的 SFR-执行行为。

11.2.5.2.5 ADV_FSP.2.5C

对于 SFR-执行 TSFI,功能规范应描述由 SFR-执行行为相关处理而引起的直接错误消息。

11.2.5.2.6 ADV_FSP.2.6C

功能规范应证实安全功能要求到 TSFI 的追溯。

11.2.5.3 评估者行为元素

11.2.5.3.1 ADV_FSP.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

11.2.5.3.2 ADV_FSP.2.2E

评估者应确定功能规范是安全功能要求的一个准确且完备的实例化。

11.2.6 ADV_FSP.3 带完整摘要的功能规范

依赖关系:ADV_TDS.1 基础设计。

11.2.6.1 开发者行为元素

11.2.6.1.1 ADV_FSP.3.1D

开发者应提供一个功能规范。

11.2.6.1.2 ADV_FSP.3.2D

开发者应提供功能规范到安全功能要求的追溯。

11.2.6.2 证据的内容和形式元素

11.2.6.2.1 ADV_FSP.3.1C

功能规范应完全描述 TSF。

11.2.6.2.2 ADV_FSP.3.2C

功能规范应描述所有的 TSFI 的目的和使用方法。

11.2.6.2.3 ADV_FSP.3.3C

功能规范应识别和描述每个 TSFI 相关的所有参数。

11.2.6.2.4 ADV_FSP.3.4C

对于每个 SFR-执行 TSFI,功能规范应描述 TSFI 相关的 SFR-执行行为。

11.2.6.2.5 ADV_FSP.3.5C

对于每个 SFR-执行 TSFI,功能规范应描述与 TSFI 的调用相关的安全实施行为和异常而引起的直接错误消息。

11.2.6.2.6 ADV_FSP.3.6C

功能规范需总结与每个 TSFI 相关的 SFR-支撑和 SFR-无关的行为。

11.2.6.2.7 ADV_FSP.3.7C

功能规范应证实安全功能要求到 TSFI 的追溯。

11.2.6.3 评估者行为元素

11.2.6.3.1 ADV_FSP.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

11.2.6.3.2 ADV_FSP.3.2E

评估者应确定功能规范是安全功能要求的一个准确且完备的实例化。

11.2.7 ADV_FSP.4 完备的功能规范

依赖关系:ADV_TDS.1 基础设计。

11.2.7.1 开发者行为元素

11.2.7.1.1 ADV_FSP.4.1D

开发者应提供一个功能规范。

11.2.7.1.2 ADV_FSP.4.2D

开发者应提供功能规范到安全功能要求的追溯。

11.2.7.2 证据的内容和形式元素

11.2.7.2.1 ADV_FSP.4.1C

功能规范应完全描述 TSF。

11.2.7.2.2 ADV_FSP.4.2C

功能规范应描述所有的 TSFI 的目的和使用方法。

11.2.7.2.3 ADV_FSP.4.3C

功能规范应识别和描述每个 TSFI 相关的所有参数。

11.2.7.2.4 ADV_FSP.4.4C

对于每个 SFR-执行 TSFI,功能规范应描述 TSFI 相关的**所有**行为。

11.2.7.2.5 ADV_FSP.4.5C

功能规范应描述可能由每个 TSFI 的调用而引起的所有直接错误消息。

11.2.7.2.6 ADV_FSP.4.6C

功能规范应证实安全功能要求到 TSFI 的追溯。

11.2.7.3 评估者行为元素

11.2.7.3.1 ADV_FSP.4.1E

评估者应**确认**所提供的信息满足证据的内容和形式的所有要求。

11.2.7.3.2 ADV_FSP.4.2E

评估者应**确定**功能规范是安全功能要求的一个准确且完备的实例化。

11.2.8 ADV_FSP.5 附加错误信息的完备的半形式化功能规范

依赖关系:ADV_TDS.1 基础设计;
ADV_IMP.1 TSF 实现表示。

11.2.8.1 开发者行为元素

11.2.8.1.1 ADV_FSP.5.1D

开发者应提供一个功能规范。

11.2.8.1.2 ADV_FSP.5.2D

开发者应提供功能规范到安全功能要求的追溯。

11.2.8.2 证据的内容和形式元素

11.2.8.2.1 ADV_FSP.5.1C

功能规范应完全描述 TSF。

11.2.8.2.2 ADV_FSP.5.2C

功能规范应用半形式化方式描述 TSFI。

11.2.8.2.3 ADV_FSP.5.3C

功能规范应描述所有的 TSFI 的目的和使用方法。

11.2.8.2.4 ADV_FSP.5.4C

功能规范应识别和描述每个 TSFI 相关的所有参数。

11.2.8.2.5 ADV_FSP.5.5C

功能规范应描述每个 TSFI 相关的所有行为。

11.2.8.2.6 ADV_FSP.5.6C

功能规范应描述可能由每个 TSFI 的调用引起的所有直接错误消息。

11.2.8.2.7 ADV_FSP.5.7C

功能规范应描述不是由 TSFI 调用而引起的所有错误消息。

11.2.8.2.8 ADV_FSP.5.8C

功能规范应为每个包含在 TSF 实现中但不是由 TSFI 调用而引起的错误消息提供基本原理。

11.2.8.2.9 ADV_FSP.5.9C

功能规范应证实安全功能要求到 TSFI 的追溯。

11.2.8.3 评估者行为元素

11.2.8.3.1 ADV_FSP.5.1E

评估者**应确认**所提供的信息满足证据的内容和形式的所有要求。

11.2.8.3.2 ADV_FSP.5.2E

评估者**应确定**功能规范是安全功能要求的一个准确且完备的实例化。

11.2.9 ADV_FSP.6 附加形式化描述的完备的半形式化功能规范

依赖关系: ADV_TDS.1 基础设计;
ADV_IMP.1 TSF 实现表示。

11.2.9.1 开发者行为元素

11.2.9.1.1 ADV_FSP.6.1D

开发者应提供一个功能规范。

11.2.9.1.2 ADV_FSP.6.2D

开发者应提供 TSF 功能规范的形式化描述。

11.2.9.1.3 ADV_FSP.6.3D

开发者应提供功能规范到安全功能要求的追溯。

11.2.9.2 证据的内容和形式元素

11.2.9.2.1 ADV_FSP.6.1C

功能规范应完全描述 TSF。

11.2.9.2.2 ADV_FSP.6.2C

功能规范应使用形式化方式描述 TSFI。

11.2.9.2.3 ADV_FSP.6.3C

功能规范应描述所有的 TSFI 的目的和使用方法。

11.2.9.2.4 ADV_FSP.6.4C

功能规范应识别和描述每个 TSFI 相关的所有参数。

11.2.9.2.5 ADV_FSP.6.5C

功能规范应描述每个 TSFI 相关的所有行为。

11.2.9.2.6 ADV_FSP.6.6C

功能规范应描述可能由每个 TSFI 的调用而引起的所有直接错误消息。

11.2.9.2.7 ADV_FSP.6.7C

功能规范应描述包含在 TSF 实现表示中的所有错误消息。

11.2.9.2.8 ADV_FSP.6.8C

功能规范应为每个包含在 TSF 实现中,但未在功能规范中描述的错误消息提供基本原理,以论证这些错误消息为何与 TSFI 无关。

11.2.9.2.9 ADV_FSP.6.9C

TSF 功能规范的形式化描述应使用形式化方式来描述 TSF,并 I 通过适当的非形式化解释性文字来支持。

11.2.9.2.10 ADV_FSP.6.10C

功能规范应证实安全功能要求到 TSFI 的追溯。

11.2.9.3 评估者行为元素

11.2.9.3.1 ADV_FSP.6.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

11.2.9.3.2 ADV_FSP.6.2E

评估者应确定功能规范是安全功能要求的一个准确且完备的实例化。

11.3 实现表示(ADV_IMP)

11.3.1 目的

实现表示(ADV_IMP)族的功能是让开发者以评估者能够分析的形式来制定 TOE 的实现表示(并且在高级别中实现 TOE)。在分析其他族(例如,分析 TOE 设计)的活动中用实现表示来证实 TOE 符合其设计,以及为其他部分的评估(例如查找脆弱性)分析提供基础。实现表示应详细说明 TSF 的内部工作,可以包括软件代码、固件代码、硬件图表和/或 IC 硬件设计代码或者设计数据等。

11.3.2 组件分级

本族中的组件是基于 TOE 设计实现的数量而分级的。

11.3.3 应用注释

用来构造实际硬件的源代码/硬件代码和/或 IC 硬件设计代码/设计数据都是实现表示的部分实例。要注意的是虽然实现表示必须是对评估者可用的,但是这并不意味着评估者需要控制那些实现表示,这点很重要。例如,开发者可以要求评估者在开发者选择的场所复查实现表示。

全部实现表示都应可用,以确保不会由于缺少信息而简化分析活动。但是,这并不意味着执行分析活动时要检查所有的实现表示,而这也几乎是不可行的;同时,与以定向抽样的方式查看实现表示相比,即便提供了全部实现表示也不太可能会提升 TOE 的保障级别。实现表示可用于分析其他的设计分解过程(例如功能规范、TOE 设计),也可用于在确认高层设计中描述的安全功能在 TOE 似乎得到了实现这一方面建立信心。在某些形式的实现表示中的某些常规方法使得我们可能很难或不可能仅仅通过实现表示来确定实际的编译结果或运行时解释执行的情况。例如,C 语言编译器的编译指令会引起编译器排除或包含全部代码。因此,提供的这样的“额外”信息或相关工具(脚本、编译器等)对于确定实现表示是很重要的。

在实现表示和 TOE 设计描述之间建立映射的目的是为了帮助评估者做分析。当 TOE 设计的分析与实现表示内容一致时,TOE 的内部工作就更容易理解。映射将作为对实现表示的索引。在低级别的组件中,只将实现表示的一个子集映射到 TOE 设计描述上。由于不确定实现表示的哪一部分需要这种映射,开发者可以预先选择映射全部的实现表示,或等待了解评估者要求映射哪部分实现表示。

开发者采用一种适合转换为真正实现的方式来处理实现表示。例如,开发者的工作可能会用到包含源代码的文件,这些源代码最终编译为 TSF 的一部分。开发者以自身习惯使用的方式来提供实现表示,这样评估者就可以使用自动化技术进行分析。这样也提升了评估者的信心,即接受检查的实现表示正是在实际的 TSF 实现中使用的(这与以文字处理文档等形式提交的表达方式不同)。应当注意的是开发者也可能使用其他形式的实现表示;这些形式也需要提供。总体目标是为评估者提供信息以最大限度地提高评估者的分析效率。

某些形式的实现表示可能需要额外的信息,因为它们对理解和分析构成了严重障碍。例如包含了外壳的源代码或者以其他方式进行了扰乱的代码,都会阻止理解和/或分析。这些形式的实现表示典型地是由 TOE 开发者在实现表示的一个版本上运行加壳或扰乱程序而产生的。虽然加壳的表示是编译过的,可能与原始的、未隐藏的表示相比更接近实现(在结构方面),但是提供这样的扰乱代码可能会使针对实现表示的分析任务花费更多的时间。当创建了这种形式的表示时,需要在组件中描述使用的加壳工具/算法的细节,以致能获得未加壳的表示。这些额外的信息可用于让评估者获得信心以确定隐藏程序不会危及任何安全功能。

11.3.4 ADV_IMP.1 TSF 实现表示

依赖关系:ADV_TDS.3 基础模块设计;
ALC_TAT.1 明确定义的开发工具。

11.3.4.1 开发者行为元素

11.3.4.1.1 ADV_IMP.1.1D

开发者应为全部 TSF 提供实现表示。

11.3.4.1.2 ADV_IMP.1.2D

开发者应提供 TOE 设计描述与实现表示实例之间的映射。

11.3.4.2 内容和形式元素

11.3.4.2.1 ADV_IMP.1.1C

实现表示应按详细级别定义 TSF,且详细程度达到无须进一步设计就能生成 TSF 的程度。

11.3.4.2.2 ADV_IMP.1.2C

实现表示应以开发人员使用的形式提供。

11.3.4.2.3 ADV_IMP.1.3C

TOE 设计描述与实现表示实例之间的映射应能证实它们的一致性。

11.3.4.3 评估者行为元素

11.3.4.3.1 ADV_IMP.1.1E

对于选取的实现表示实例,评估者应确认提供的信息满足证据的内容和形式的所有要求。

11.3.5 ADV_IMP.2 TSF 实现表示完全映射

依赖关系:ADV_TDS.3 基础模块设计;
ALC_TAT.1 明确定义的开发工具;
ALC_CMC.5 高级支持。

11.3.5.1 开发者行为元素

11.3.5.1.1 ADV_IMP.2.1D

开发者应为全部 TSF 提供实现表示。

11.3.5.1.2 ADV_IMP.2.2D

开发者应提供 TOE 设计描述与全部实现表示之间的映射。

11.3.5.2 证据的内容和形式元素

11.3.5.2.1 ADV_IMP.2.1C

实现表示应按详细级别定义 TSF,且详细程度达到无须进一步设计就能生成 TSF 的程度。

11.3.5.2.2 ADV_IMP.2.2C

实现表示应以开发人员使用的形式提供。

11.3.5.2.3 ADV_IMP.2.3C

TOE 设计描述与全部实现表示之间的映射应证实它们的一致性。

11.3.5.3 评估者行为元素

11.3.5.3.1 ADV_IMP.2.1E

评估者应确认提供的信息满足证据的内容和形式的所有要求。

11.4 TSF 内部(ADV_INT)

11.4.1 目的

本族负责评估 TSF 的内部结构。内部结构合理的 TSF 容易实现并且可能导致脆弱性的缺陷也较少;因为无缺陷引入,也更加容易维护。

11.4.2 组件分级

本族中的组件是基于结构的数目和最小化复杂度的需求而分级的。ADV_INT.1 结构合理的 TSF 内部子集仅要求选中的 TSF 部分内部结构合理。在 EAL 中不包含该组件,因为本组件被认为只在特殊环境中使用(例如发起者对密码模块特别关心,而且该模块独立于 TSF 的其他模块)并且不会大范围应用。

在下一级别,对结构合理的内部的要求是置于整个 TSF 之上的。最后,在最高级别的组件中引入的复杂度的要最小。

11.4.3 应用注释

当这些要求应用到 TSF 内部结构时,通常会帮助开发者和评估者理解 TSF,并且会对设计和评估测试程序提供依据。而且,提高 TSF 的可理解性帮助开发者节省维护的工作量。

本族的要求在相当抽象级别上给出。TOE 的多样性导致不可能对任何方面都规定出比“结构合理”或“最小复杂度”更明确的指标。对结构和复杂度的判断可从 TOE 中使用的特定技术导出。例如,一个软件产品如果能体现软件工程学科提到的规格要求,那么它可能就是结构合理的。本族组件需要指出衡量特征是否结构合理和过度复杂的标准。

11.4.4 ADV_INT.1 结构合理的 TSF 内部子集

依赖关系:ADV_IMP.1 TSF 实现表示;
ADV_TDS.3 基础模块设计;
ALC_TAT.1 明确定义的开发工具。

11.4.4.1 目的

本组件的目的是提供一种方法用以保证 TSF 特定部分是结构合理的。其用意是用合理的工程原则设计实现整个 TSF,但是只在一个特定的子集上进行分析。

11.4.4.2 应用注释

本组件要求 PP 或 ST 的作者用 TSF 子集进行赋值。这个子集可以根据 TSF 的内部情况在任何

抽象层面上进行标识。例如：

- 在 TOE 设计中标识的 TSF 结构性元素（例如“开发者应设计和实现审计子系统，以使得 TSF 内部结构合理。”）
- 实现（例如“开发者应设计和实现 encrypt.c 和 decrypt.c 文件，以使其内部结构合理。”或“开发者应设计和实现“IC 芯片 6227”，以使其内部结构合理。”）

仅仅参考声明的安全功能要求可能不容易达到目的，因为这并没有指出需要分析的具体目标。（例如“开发者应设计和实现 TSF 中像在 FPR_ANO.2 中定义的那样提供匿名的一部分，以使其内部结构合理。”）。

本组件适用范围有限，主要适用于当潜在恶意的用户/主体对 TSFI 的访问权限有限或严格受控的情况，或存在其他保护措施（如域分离）能将已选 TSF 子集从对未选子集的攻击中屏蔽出来的情况。（如，独立于 TSF 其他部分的密码功能就是结构合理的）

11.4.4.3 开发者行为元素

11.4.4.3.1 ADV_INT.1.1D

开发者应设计和实现[赋值:TSF 子集]，使内部结构合理。

11.4.4.3.2 ADV_INT.1.2D

开发者应提供内部描述和论证过程。

11.4.4.4 证据的内容和形式元素

11.4.4.4.1 ADV_INT.1.1C

论证过程应解释用来判断“结构合理”的含义的特性。

11.4.4.4.2 ADV_INT.1.2C

TSF 内部描述应证实指定的 TSF 子集结构合理。

11.4.4.5 评估者行为元素

11.4.4.5.1 ADV_INT.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

11.4.4.5.2 ADV_INT.1.2E

评估者应执行指定的 TSF 子集内部分析。

11.4.5 ADV_INT.2 内部结构合理

- 依赖关系:ADV_IMP.1 TSF 实现表示；
- ADV_TDS.3 基础模块设计；
- ALC_TAT.1 明确定义的开发工具。

11.4.5.1 目的

本组件的目的是提供一种要求 TSF 结构合理的手段。意在于使用合理的工程原理设计和实现整个 TSF。

11.4.5.2 应用注释

期望用来自于 TOE 中的具体技术充分判断结构。此组件要求确定标准用以衡量结构是合理的。

11.4.5.3 开发者行为元素

11.4.5.3.1 ADV_INT.2.1D

开发者应设计和实现**整个 TSF**,使其内部结构合理。

11.4.5.3.2 ADV_INT.2.2D

开发者应提供内部描述和论证过程。

11.4.5.4 内容和形式元素

11.4.5.4.1 ADV_INT.2.1C

论证过程应**描述**用于判定“结构合理”的含意的特性。

11.4.5.4.2 ADV_INT.2.2C

TSF 内部描述应证实指定的**整个 TSF** 结构合理。

11.4.5.5 评估者行为元素

11.4.5.5.1 ADV_INT.2.1E

评估者**应确认**所提供的信息满足证据的内容和形式的所有要求。

11.4.5.5.2 ADV_INT.2.2E

评估者**应执行**指定的 TSF 子集内部分析。

11.4.6 ADV_INT.3 内部复杂度最小化

依赖关系:ADV_IMP.1 TSF 实现表示;
ADV_TDS.3 基础模块设计;
ALC_TAT.1 明确定义的开发工具。

11.4.6.1 目的

本组件的目的是提供一种要求 TSF 结构合理且复杂度最小化的方法。意在于使用合理的工程原理设计和实现整个 TSF。

11.4.6.2 应用注释

对于结构和复杂度充分性的判断期望源自 TOE 中使用的具体技术。此组件要求确定标准用以衡量复杂度是合理的。

11.4.6.3 开发者行为元素

11.4.6.3.1 ADV_INT.3.1D

开发者应设计**和实现整个 TSF**,使内部结构合理。

11.4.6.3.2 ADV_INT.3.2D

开发者应提供内部描述和论证过程。

11.4.6.4 内容和形式元素

11.4.6.4.1 ADV_INT.3.1C

论证过程应解释描述用于判定“结构合理”及复杂性的含义的特性。

11.4.6.4.2 ADV_INT.3.2C

TSF 内部描述应证实指定的整个 TSF 结构合理且不过于复杂。

11.4.6.5 评估者行为元素

11.4.6.5.1 ADV_INT.3.1E

评估者应**确认**所提供的信息满足证据的内容和形式的所有要求。

11.4.6.5.2 ADV_INT.3.2E

评估者应**执行**整个 TSF 的内部分析。

11.5 安全策略模型(ADV_SPM)

11.5.1 目的

本族目的是通过建立一个形式化的 TSF 安全策略模型,以及功能规范与安全策略模型之间的对应关系来提供额外的保障。为了保持内部一致性,安全策略模型期望能通过数学证明的方式来建立形式化的安全规则。

11.5.2 组件分级

本族只包含一个组件。

11.5.3 应用注释

TOE 的缺陷来源于对安全要求的错误理解或对安全要求的实现缺陷。对安全要求进行充分地定义可能很难实现,这是因为给出的定义必须足够精确,以防止在 TOE 实现过程中出现不期望的结果或细微的缺陷。在整个设计、实现和审查过程中,模型化的安全要求可用于指导准确的设计和实施过程,从而为 TOE 满足安全要求模型这一目标提供更多的保障。如果模型采用形式化语言来描述,且安全要求通过形式证明来验证,那么这个模型和由此产生的指导的准确性将得到显著改善。

形式化安全策略模型的建立可用于帮助识别和消除歧义的、不一致的、矛盾的或无法实施的安全策略元素。TOE 完成后,形式化模型可帮助评估者判断开发者对实现的安全功能的理解程度,以及安全要求和 TOE 的设计之间是否一致。一个无歧义的证明可用于建立对模型的信任。

形式化的安全模型是对重要的安全特征及其与 TOE 行为之间的关系的一种准确的形式化描述;它给出了一系列的规则和惯例,以调节 TSF 对系统资源的管理、保护和控制。该模型包括一系列限制条件和属性以指定信息和计算资源应如何防止被用于违背安全功能要求,同时还给出了一系列有说服力的工程实践论据,以说明这些限制条件和属性是如何在安全功能要求的执行过程中发挥重要作用的。模型不仅包含了描述安全功能的形式化内容,也包含了辅助性的文字来解释模型以及模型成立的条件。

TSF 安全行为的建模是综合其外部行为(TSF 与 TOE 的其他部分及其运行环境的交互)和内部行为来进行的。

TOE 的安全策略模型是通过分析 ST 中提出的安全要求,从其实现中以非形式的方法抽象出来的。如果 TOE 的工作原理(也称为“不变量”)是由其特征来驱动的,这个非形式化的抽象过程就能成功。非形式化的论证易于犯错,特别是涉及众多的主体、客体和操作之间的关系时。形式化模型的目的在于提升实施过程的严密性。为了最小化不安全状态产生的风险,安全策略模型的规则和性质应被映射为某个形式化模型中的相应属性和特征,此后安全属性可在这个严谨的模型下通过定理和形式化证明来获得。

虽然“形式化安全策略模型”是在学术界使用的术语,ISO/IEC 15408 没有对“安全”进行固定的定义,这里安全等同于对安全功能要求的任意声明。因此,这里的形式化安全策略模型仅仅是已声明的安全功能要求集合的形式化表示。

历来安全策略只与访问控制策略相关,这些访问控制策略或者是基于标记的(强制访问控制),或者是基于用户的(自由访问控制)。然而正如在 PP 或 ST 中描述的,安全策略不仅限于访问控制,还有审计策略、标识策略、鉴别策略、加密策略、管理策略和由 TOE 执行的任何其他安全策略。为此,ADV_SPM.1.1D 包含了赋值要求,以明确这些已采用形式化方法建模后的策略。

11.5.4 ADV_SPM.1 形式化 TOE 安全策略模型

依赖关系:ADV_FSP.4 完备的功能规范。

11.5.4.1 开发者行为元素

11.5.4.1.1 ADV_SPM.1.1D

开发者应提供一个形式化的安全策略模型[赋值:被形式化模型化的策略列表]。

11.5.4.1.2 ADV_SPM.1.2D

对于形式化安全策略模型覆盖的每个策略,该模型应标识构成这一策略的安全功能要求声明的有关部分。

11.5.4.1.3 ADV_SPM.1.3D

开发者应提供该模型与形式化功能规范的对应性的形式化证明。

11.5.4.1.4 ADV_SPM.1.4D

开发者应提供该模型与功能规范的对应性的论证。

11.5.4.2 内容和形式元素

11.5.4.2.1 ADV_SPM.1.1C

该模型应是形式化的,必要时辅以解释性的文字,并且标识模型化的 TSF 安全策略。

11.5.4.2.2 ADV_SPM.1.2C

对于所有被模型化的策略,模型定义该 TOE 的安全,提供该 TOE 不能达到非安全状态的形式化证明。

11.5.4.2.3 ADV_SPM.1.3C

该模型与功能规范的一致性应采用正确的形式化级别进行论述。

11.5.4.2.4 ADV_SPM.1.4C

该对应性应表明功能规范相对该模型是一致的和完备的。

11.5.4.2.5 ADV_SPM.1.5C

该对应性论证应表明功能规范中描述的接口相对于 ADV_SPM.1.1D 中赋值的策略是一致的和完备的。

11.5.4.3 评估者行为元素

11.5.4.3.1 ADV_SPM.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

11.6 TOE 设计(ADV_TDS)

11.6.1 目的

TOE 的设计描述提供了与 TSF 描述的上下文,以及对 TSF 的详尽描述。随着保障要求的增加,在说明中提供的详细程度也随之增加。随着 TSF 规模和复杂性的增加,适合多级分解。设计要求的目的是提供信息(与给定的保障级别相称的),以便确定是否实现了安全功能要求。

11.6.2 组件分级

本族中组件的分级是建立在所需的 TSF 信息量和设计描述要求的形式化程度的基础之上。

11.6.3 应用注释

设计文档的目的是为确定 TSF 边界以及描述 TSF 如何实现安全功能要求提供充足的信息。设计文档的数量和结构将取决于 TOE 的复杂性和安全功能要求的数量;通常,实现了大量的安全功能要求的非常复杂的 TOE 将比只实现了少量安全功能要求的非常简单的 TOE 需要更多的设计文档。非常复杂的 TOE 将受益(所提供的保障条款)于在描述设计中的不同层次的分解,而很简单的 TOE 不要求高层和低层的实现描述。

本族使用了两个层次的分解:子系统和模块。模块是功能最具体的描述:它是实现的描述。如果没有更多的设计决定,开发者应能够实现模块描述的部分 TOE。子系统是 TOE 设计的描述,它有助于提供高层次的描述,TOE 的一部分是做什么,以及如何做。因此,子系统可进一步分为低层次的子系统或分为模块。为了充分有效的描述 TOE 是如何工作的,非常复杂的 TOE 可能需要几个层次的子系统。相反,很简单的 TOE 可能并不需要子系统层次的描述,模块可以清楚地描述 TOE 是如何工作的。

设计文档通常采用的描述方法是,随着保障级别的增加,描述的重点由概括的(子系统级)转换为更详细的(模块级)。在某些情况下,当 TOE 很简单足以进行模块级别的描述时,虽然保障级要求子系统级别的描述,但单独提供一个模块级别的描述也是合适的。然而,对于复杂的 TOE,情况与此不同,若没有一个子系统级别的描述,即便对模块细节进行了大量的描述也让人难以理解。

这种方法遵循一般范型,提供 TSF 实现的额外细节将导致更多的保障,以确保正确实现安全功能要求,在测试(ATE:测试)中提供能够用于证实这些的信息。

在本族的要求中,术语接口用作通信(两个子系统或模块之间的)的手段。它描述了如何调用通信,这是类似于 TSFI(见功能规范(ADV_FSP))的细节。术语相互作用用于识别通信的目的,它识别两个子系统或模块进行通信的原因。

11.6.3.1 子系统和模块细节

定义子系统和模块细节的要求：

- a) 子系统和模块的简单列表；
- b) 子系统和模块可以按照“SFR-执行”、“SFR-支撑”或“SFR-无关”进行分类(隐含或明确)；这些术语与功能规范(ADV_FSP)中的用法相同；
- c) 子系统的行为是它要做什么，也可分为 SFR-执行、SFR-支撑或 SFR-无关三类。对子系统行为进行的分类不会超出子系统本身与安全功能要求的相关性。例如，SFR-执行子系统能有 SFR-执行的行为，也能有 SFR-支撑的或 SFR-无关的行为；
- d) 子系统的行为概述是它执行的动作的概述(如“TCP 子系统将 IP 数据报组装成可靠的字节流的”)；
- e) 子系统的行为描述是它所做的一切的解释。该描述应到可以很容易地确定行为是否与安全功能要求的实施有任何相关性的详细程度；
- f) 子系统或模块之间的相互作用的描述标识子系统或模块通信的原因，和传递信息的特征。它不需要以接口规范那样的详细程度来定义信息。例如，“子系统 X 从内存管理器请求内存，内存管理器返回分配的内存地址”；
- g) 接口描述提供了模块之间的相互作用是如何实现的细节。而不是描述模块通信的原因或通信的目的(即交互作用的描述)，接口描述说明通信是如何实现的细节，包括了消息、信号、内部进程通信等方面的结构和内容；
- h) 目的描述模块如何提供它们的功能。它提供足够的细节，不需要进一步的设计决策。实现模块的实现表示和模块目的之间的对应关系，应该是显而易见的；
- i) 无论元素中标识什么都描述模块。

在 A.4 ADV_TDS:子系统和模块中，进一步非常详尽地解释了子系统和模块以及 SFR-执行。

11.6.4 ADV_TDS.1 基础设计

依赖关系：ADV_FSP.2 安全执行功能规范。

11.6.4.1 开发者行为元素

11.6.4.1.1 ADV_TDS.1.1D

开发者应提供 TOE 的设计。

11.6.4.1.2 ADV_TDS.1.2D

开发者应提供从功能规范的 TSFI 到 TOE 设计中获取到的最低层分解的映射。

11.6.4.2 内容和形式元素

11.6.4.2.1 ADV_TDS.1.1C

设计应根据子系统描述 TOE 的结构。

11.6.4.2.2 ADV_TDS.1.2C

设计应标识 TSF 的所有子系统。

11.6.4.2.3 ADV_TDS.1.3C

设计应对每一个 SFR-支撑或 SFR-无关的 TSF 子系统的行为进行足够详细的描述，以确定它不是

SFR-执行。

11.6.4.2.4 ADV_TDS.1.4C

设计应概括 SFR-执行子系统的 SFR-执行行为。

11.6.4.2.5 ADV_TDS.1.5C

设计应描述 TSF 的 SFR-执行子系统间的相互作用和 TSF 的 SFR-执行子系统与其他 TSF 子系统间的相互作用。

11.6.4.2.6 ADV_TDS.1.6C

映射关系应证实 TOE 设计中描述的所有行为能够映射到调用它的 TSFI。

11.6.4.3 评估者行为元素

11.6.4.3.1 ADV_TDS.1.1E

评估者应确认提供的信息满足证据的内容与形式的所有要求。

11.6.4.3.2 ADV_TDS.1.2E

评估者应确定设计是所有安全功能要求的正确且完备的实例。

11.6.5 ADV_TDS.2 结构化设计

依赖关系:ADV_FSP.3 带完整摘要的功能规范。

11.6.5.1 开发者行为元素

11.6.5.1.1 ADV_TDS.2.1D

开发者应提供 TOE 的设计。

11.6.5.1.2 ADV_TDS.2.2D

开发者应提供从功能规范的 TSFI 到 TOE 设计中获取到的最低层分解的映射。

11.6.5.2 内容和形式元素

11.6.5.2.1 ADV_TDS.2.1C

设计应根据子系统描述 TOE 的结构。

11.6.5.2.2 ADV_TDS.2.2C

设计应标识 TSF 的所有子系统。

11.6.5.2.3 ADV_TDS.2.3C

设计应对每一个 TSF 的 SFR-无关子系统的行为进行足够详细的描述,以确定它是 SFR-无关。

11.6.5.2.4 ADV_TDS.2.4C

设计应描述 SFR-执行子系统的 SFR-执行行为。

11.6.5.2.5 ADV_TDS.2.5C

设计应概括 SFR-执行子系统的 **SFR-支撑**和 **SFR-无关**行为。

11.6.5.2.6 ADV_TDS.2.6C

设计应概括 **SFR-支撑**子系统的行为。

11.6.5.2.7 ADV_TDS.2.7C

设计应描述 TSF 所有子系统间的相互作用。

11.6.5.2.8 ADV_TDS.2.8C

映射关系应证实 TOE 设计中描述的所有行为能够映射到调用它的 TSFI。

11.6.5.3 评估者行为元素

11.6.5.3.1 ADV_TDS.2.1E

评估者应确认提供的信息满足证据的内容与形式的所有要求。

11.6.5.3.2 ADV_TDS.2.2E

评估者应确定设计是所有安全功能要求的正确且完全的实例。

11.6.6 ADV_TDS.3 基础模块设计

依赖关系:ADV_FSP.4 完备的功能规范。

11.6.6.1 开发者行为元素

11.6.6.1.1 ADV_TDS.3.1D

开发者应提供 TOE 的设计。

11.6.6.1.2 ADV_TDS.3.2D

开发者应提供从功能规范的 TSFI 到 TOE 设计中获取到的最低层分解的映射。

11.6.6.2 内容和形式元素

11.6.6.2.1 ADV_TDS.3.1C

设计应根据子系统描述 TOE 的结构。

11.6.6.2.2 ADV_TDS.3.2C

设计应根据模块描述 TSF。

11.6.6.2.3 ADV_TDS.3.3C

设计应标识 TSF 的所有子系统。

11.6.6.2.4 ADV_TDS.3.4C

设计应描述每一个 TSF 子系统。

11.6.6.2.5 ADV_TDS.3.5C

设计应描述 TSF 所有子系统间的相互作用。

11.6.6.2.6 ADV_TDS.3.6C

设计应提供 TSF 子系统到 TSF 模块间的映射关系。

11.6.6.2.7 ADV_TDS.3.7C

设计应描述每一个 SFR-执行模块,包括它的目的及与其他模块间的相互作用。

11.6.6.2.8 ADV_TDS.3.8C

设计应描述每一个 SFR-执行模块,包括它的安全功能要求相关接口、其他接口的返回值、与其他模块间的相互作用及调用的接口。

11.6.6.2.9 ADV_TDS.3.9C

设计应描述每一个 SFR-支撑或 SFR-无关模块,包括它的目的及与其他模块间的相互作用。

11.6.6.2.10 ADV_TDS.3.10C

映射关系应论证 OE 设计中描述的所有行为能够映射到调用它的 TSFI。

11.6.6.3 评估者行为元素

11.6.6.3.1 ADV_TDS.3.1E

评估者应确认提供的信息满足证据的内容与形式的所有要求。

11.6.6.3.2 ADV_TDS.3.2E

评估者应确定设计是所有安全功能要求的正确且完全的实例。

11.6.7 ADV_TDS.4 半形式化模块设计

依赖关系:ADV_FSP.5 附加错误信息的完备的半形式化功能规范。

11.6.7.1 开发者行为元素

11.6.7.1.1 ADV_TDS.4.1D

开发者应提供 TOE 的设计。

11.6.7.1.2 ADV_TDS.4.2D

开发者应提供从功能规范的 TSFI 到 TOE 设计中获取到的最低层分解的映射。

11.6.7.2 内容和形式元素

11.6.7.2.1 ADV_TDS.4.1C

设计应根据子系统描述 TOE 的结构。

11.6.7.2.2 ADV_TDS.4.2C

设计应根据模块描述 TSF,以 **SFR-执行**、**SFR-支撑**或 **SFR-无关**标出每一个模块。

11.6.7.2.3 ADV_TDS.4.3C

设计应标识 TSF 的所有子系统。

11.6.7.2.4 ADV_TDS.4.4C

设计应提供每一个 TSF 子系统的半形式化描述,适当时配以非形式化的、解释性的描述。

11.6.7.2.5 ADV_TDS.4.5C

设计应描述 TSF 所有子系统间的相互作用。

11.6.7.2.6 ADV_TDS.4.6C

设计应提供 TSF 子系统到 TSF 模块间的映射关系。

11.6.7.2.7 ADV_TDS.4.7C

设计应描述每一个 SFR-执行和 **SFR-支撑**模块,包括它的目的及与其他模块间的相互作用。

11.6.7.2.8 ADV_TDS.4.8C

设计应描述每一个 SFR-执行和 **SFR-支撑**模块,包括它的安全功能要求相关接口、其他接口的返回值、与其他模块间的相互作用及调用的接口。

11.6.7.2.9 ADV_TDS.4.9C

设计应描述每一个 SFR-支撑和 SFR-无关模块,包括它的目的及与其他模块间的相互作用。

11.6.7.2.10 ADV_TDS.4.10C

映射关系应证实 TOE 设计中描述的所有行为能够映射到调用它的 TSFI。

11.6.7.3 评估者行为元素

11.6.7.3.1 ADV_TDS.4.1E

评估者应确认提供的信息满足证据的内容与形式的所有要求。

11.6.7.3.2 ADV_TDS.4.2E

评估者应确定设计是所有安全功能要求的正确且完全的实例。

11.6.8 ADV_TDS.5 完全半形式化模块设计

依赖关系:ADV_FSP.5 附加错误信息的完备的半形式化功能规范。

11.6.8.1 开发者行为元素

11.6.8.1.1 ADV_TDS.5.1D

开发者应提供 TOE 的设计。

11.6.8.1.2 ADV_TDS.5.2D

开发者应提供从功能规范的 TSFI 到 TOE 设计中获取到的最低层分解的映射。

11.6.8.2 内容和形式元素

11.6.8.2.1 ADV_TDS.5.1C

设计应根据子系统描述 TOE 的结构。

11.6.8.2.2 ADV_TDS.5.2C

设计应从模块的角度描述 TSF,把每个模块指定为 SFR-执行、SFR-支撑或 SFR-无关。

11.6.8.2.3 ADV_TDS.5.3C

设计应标识 TSF 的所有子系统。

11.6.8.2.4 ADV_TDS.5.4C

设计应提供每一个 TSF 子系统的半形式化描述,适当时配以非形式化的、解释性的描述。

11.6.8.2.5 ADV_TDS.5.5C

设计应描述 TSF 所有子系统间的相互作用。

11.6.8.2.6 ADV_TDS.5.6C

设计应提供 TSF 子系统到 TSF 模块间的映射关系。

11.6.8.2.7 ADV_TDS.5.7C

设计应为每一个模块提供一个半形式化描述,包括它的目的、相互作用、接口、其他接口的返回值、被其他模块调用的接口,适当时配以非形式化的、解释性的描述。

11.6.8.2.8 ADV_TDS.5.8C

映射关系应证实 TOE 设计中描述的所有行为能够映射到调用它的 TSFI。

11.6.8.3 评估者行为元素

11.6.8.3.1 ADV_TDS.5.1E

评估者应确认提供的信息满足证据的内容与形式的所有要求。

11.6.8.3.2 ADV_TDS.5.2E

评估者应确定设计是所有安全功能要求的正确且完备的实例。

11.6.9 ADV_TDS.6 带形式化高层设计表示的完全半形式化模块设计

依赖关系:ADV_FSP.6 附加形式化描述的完备的半形式化功能规范。

11.6.9.1 开发者行为元素

11.6.9.1.1 ADV_TDS.6.1D

开发者应提供 TOE 的设计。

11.6.9.1.2 ADV_TDS.6.2D

开发者应提供一个从功能规范的 TSFI 到 TOE 设计所描述的最低层分解的映射。

11.6.9.1.3 ADV_TDS.6.3D

开发者应提供 TSF 子系统的形式化说明。

11.6.9.1.4 ADV_TDS.6.4D

开发者应提供 TSF 子系统和功能规范的形式化说明间的一致性论证。

11.6.9.2 内容和形式元素

11.6.9.2.1 ADV_TDS.6.1C

设计应根据子系统描述 TOE 的结构。

11.6.9.2.2 ADV_TDS.6.2C

设计应从模块的角度描述 TSF,把每个模块指定为 SFR-执行、SFR-支撑或 SFR-无关。

11.6.9.2.3 ADV_TDS.6.3C

设计应标识 TSF 的所有子系统。

11.6.9.2.4 ADV_TDS.6.4C

设计应提供每一个 TSF 子系统的半形式化描述,适当时配以非形式化的、解释性的描述。

11.6.9.2.5 ADV_TDS.6.5C

设计应描述 TSF 所有子系统间的相互作用。

11.6.9.2.6 ADV_TDS.6.6C

设计应提供 TSF 子系统到 TSF 模块间的映射关系。

11.6.9.2.7 ADV_TDS.6.7C

设计应以半形式化方式描述每一个模块,包括它的目的、相互作用、接口、其他接口的返回值、被其他模块调用的接口,适当时用非形式化的、解释性的文字进行支撑。

11.6.9.2.8 ADV_TDS.6.8C

TSF 子系统的形式化说明应以形式化方式描述 TSF,适当时配以非形式化的、解释性的描述。

11.6.9.2.9 ADV_TDS.6.9C

映射关系应证实 TOE 设计中描述的所有行为能够映射到调用它的 TSFI。

11.6.9.2.10 ADV_TDS.6.10C

TSF 子系统和功能规范的形式化说明间的一致性证明应证实 TOE 设计中描述的所有行为都是调用它的 TSFI 的正确且完备的提炼。

11.6.9.3 评估者行为元素

11.6.9.3.1 ADV_TDS.6.1E

评估者应确认提供的信息满足证据的内容与形式的所有要求。

11.6.9.3.2 ADV_TDS.6.2E

评估者应确定设计是所有安全功能要求的正确且完备的实例。

12 AGD 类:指导性文档

指导性文档类为所有用户角色提供了指南文档的要求。为了安全的准备和操作 TOE,有必要描述所有有关 TOE 安全操作方面的内容,这个类同时也描述了无意的错误配置和操作 TOE 的情况。

在多数情况下,按照下面的方式提供指南是合适的:依据 TOE 准备和 TOE 操作分别提供文档,或者依据不同的用户角色分别提供文档,如最终用户、管理员用户和运用软件或硬件接口进行应用开发者等。

指导性文档类细分为两个族,分别是关于预备用户指南(使交付的 TOE 达到与 ST 中描述的一致性的操作环境评估配置状态)和操作用户指南(在 TOE 处于评估配置状态下的操作)。

图 12 给出了本类中的族及族中各组件之间的层次关系。

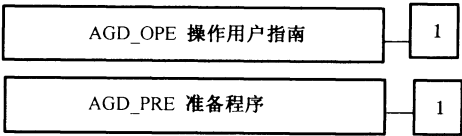


图 12 AGD:指导性文档类分解

12.1 操作用户指南(AGD_OPE)

12.1.1 目的

操作用户指南是一种书面材料,计划被 TOE 评估配置中提到的所有类型的用户使用,这包括:最终用户、保证 TOE 以正确方式和最高安全方式运行的维护和管理人员以及其他使用 TOE 外部接口的用户(如程序员)。操作用户指南描述 TSF 所提供的安全功能,提供说明和指南(包括警告),以帮助理解 TSF 及其安全使用所必须的关键信息和动作。容易误解的或者不合理的指南不应出现在指导性文档中,而运行的所有模式的安全程序则应该包括在其中,以便于检测不安全状态。

操作用户指南为非恶意用户、管理员、应用提供者及其他运行 TOE 外部接口的人理解 TOE 的安全操作并且有目的地使用它提供一定的信心。用户指南的评估包括研究能否以一种 TOE 用户认为是安全而实际上是不安全的方式使用 TOE。目的是要减小在运行过程中,那些可能使安全功能不能执行、无效或者不能成功激活安全功能的人为疏忽或者其他错误,最终导致没有检测到的安全状态存在的风险。

12.1.2 组件分级

本族中仅包含一个组件。

12.1.3 应用注释

或许存在 TOE 标识的不同用户角色或用户组能和 TSF 进行交互,在操作用户指南中应考虑这些

用户角色和用户组。他们也许被大致按照管理员用户和非管理员用户分组,或者按照 TOE 的接收、验收、安装和维护、应用开发、修订、审计、日常管理和最终用户进行更加明确的分组。一种角色能包含大量权限或者只包含有一个权限。

AGD_OPE.1.1C 用户操作 TOE 期间与 PP/ST 中描述的运行环境的安全问题定义和安全目的有关的任何警示信息需包含在指南中。

AGD_OPE.1.3C 中涉及的安全值的概念与用户支配的安全参数有关。此时指南需提供这类参数的安全或不安全的设置。

AGD_OPE.1.4C 要求用户指南描述对所有安全相关事件的适当反应。虽然许多安全相关事件是功能执行的结果,但并不总是这样(如审计日志已满、检测到一个入侵)。此外,一个安全相关事件可以作为一系列特定功能的结果,相反地,几个安全相关事件也可能被一个功能触发。

AGD_OPE.1.7C 要求用户指南是清晰和合理的,容易误解的或者不合理的指南也许会让 TOE 的用户相信 TOE 是安全的,而实际上它不是。

容易造成误解的指南实例,如:一条指令可以用多于一种方式进行理解,其中一种也许会导致一种不安全的状态。

不合理的指南实例:按照复杂的程序交付且希望用户遵从该指南是不合理的。

12.1.4 AGD_OPE.1 操作用户指南

依赖关系:ADV_FSP.1 基本功能规范。

12.1.4.1 开发者行为元素

12.1.4.1.1 AGD_OPE.1.1D

开发者应提供操作用户指南。

12.1.4.2 内容和形式元素

12.1.4.2.1 AGD_OPE.1.1C

操作用户指南应对每一种用户角色进行描述,在安全处理环境中应被控制的用户可访问的功能和特权,包含适当的警示信息。

12.1.4.2.2 AGD_OPE.1.2C

操作用户指南应对每一种用户角色进行描述,怎样以安全的方式使用 TOE 提供的可用接口。

12.1.4.2.3 AGD_OPE.1.3C

操作用户指南应对每一种用户角色进行描述,可用功能和接口,尤其是受用户控制的所有安全参数,适当时应指明安全值。

12.1.4.2.4 AGD_OPE.1.4C

操作用户指南应对每一种用户角色明确说明,与需要执行的用户可访问功能有关的每一种安全相关事件,包括改变 TSF 所控制实体的安全特性。

12.1.4.2.5 AGD_OPE.1.5C

操作用户指南应标识 TOE 运行的所有可能状态(包括操作导致的失败或者操作性错误)。它们与维持安全运行之间的因果关系和联系。

12.1.4.2.6 AGD_OPE.1.6C

操作用户指南应对每一种用户角色进行描述,为了充分实现 ST 中描述的运行环境安全目的所必须执行的安全策略。

12.1.4.2.7 AGD_OPE.1.7C

操作用户指南应是明确和合理的。

12.1.4.3 评估者行为元素

12.1.4.3.1 AGD_OPE.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

12.2 准备程序(AGD_PRE)

12.2.1 目的

准备程序用于保证 TOE 以开发者预期的安全方式被接收和安装。要求为实现从 TOE 交付到使它进入初始运行环境的安全过渡做准备。TOE 是否以一种不安全的方式进行配置或者安装,而这种方式是 TOE 用户认为是安全的。

12.2.2 组件分级

本族仅包含一个组件。

12.2.3 应用注释

满足这些要求的应用依赖多方面而变化,例如 TOE 是否是以运行状态交付的,或者是否在 TOE 拥有者场所安装了 TOE。

准备程序包含的第一个程序是消费者按照开发者交付程序安全地接收 TOE。如果开发者没有定义交付程序,安全验收必须得到其他的保证。

TOE 的安装使它的运行环境状态与 ST 中提供的运行环境安全目的一致。

也有可能不需要安装,例如:智能卡,这种情况下要求和分析安装程序是不合适的。

由于准备程序不经常使用,而且可能只使用一次,本保障族的要求和操作用户指南(AGD_OPE)保障族的要求分开陈述。

12.2.4 AGD_PRE.1 准备程序

依赖关系:无。

12.2.4.1 开发者行为元素

12.2.4.1.1 AGD_PRE.1.1D

开发者应提供 TOE,包括它的准备程序。

12.2.4.2 内容和形式元素

12.2.4.2.1 AGD_PRE.1.1C

准备程序应描述与开发者交付程序相一致的安全接收所交付 TOE 必需的所有步骤。

12.2.4.2.2 AGD_PRE.1.2C

准备程序应描述安全安装 TOE 以及安全准备与 ST 中描述的运行环境安全目的一致运行环境必需的所有步骤。

12.2.4.3 评估者行为元素

12.2.4.3.1 AGD_PRE.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

12.2.4.3.2 AGD_PRE.1.2E

评估者应运用准备程序确认 TOE 运行能被安全的准备。

13 ALC 类：生命周期支持

生命周期支持是在开发和维护期间，建立规则和对 TOE 的细化过程进行控制。如果安全分析和证据的产生均能够作为开发和维护活动的主要组成部分有规律地进行，那么将增强对 TOE 安全要求和 TOE 之间一致性的信心。

在产品生命周期内，是按照 TOE 由开发者负责或者用户负责来区分生命周期阶段，而不是按照 TOE 处于开发环境或用户环境区分的。将 TOE 移交给用户的时间点，也就是从 ALC“生命周期支持”类到 AGD“指导性文档”类的时间点。

ALC 生命周期类由七个族组成。ALC_LCD“生命周期定义”族是 TOE 生命周期的高级描述，ALC_CMC“CM 能力”族要求对配置项的管理进行详细描述，ALC_CMS“CM 范围”族要求要求以定义的方式管理一个最小的配置项集合。ALC_DVS“开发安全”族与开发者物理的、程序的、人员的以及其他的安全措施有关，ALC_TAT“工具和技术”族关于开发者使用的开发工具和实现标准，ALC_FLR“缺陷纠正”族负责安全缺陷的处理。ALC_DEL“交付”族定义将 TOE 移交给消费者使用的程序，在 TOE 开发过程中产生的交付过程称为传输，是由本类其他族中定义的集成和接受程序来处理的。

在本类中，开发和相关的术语（开发者、开发）是指从一般意义上组成的开发和生产，而生产特指由实现表示变成最终 TOE 的转变过程。

图 13 给出了本类中的族及族中各组件之间的层次关系。

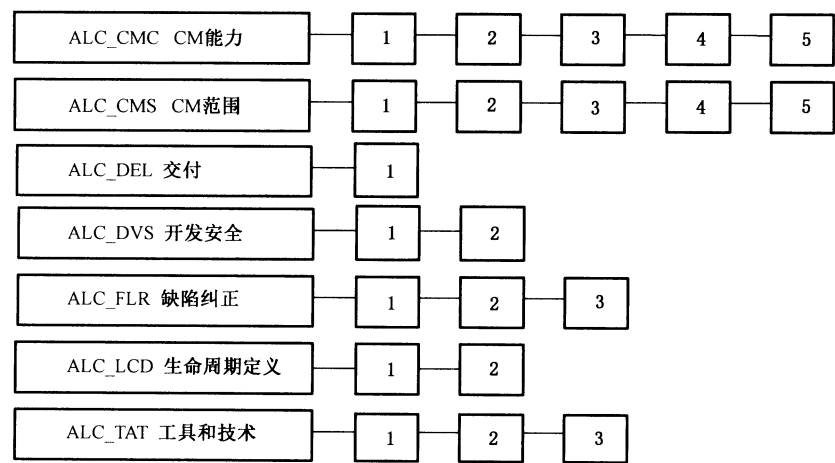


图 13 生命周期支持类分解

13.1 CM 能力(ALC_CMC)

13.1.1 目的

配置管理(CM)是一种用于增强 TOE 满足安全功能要求的保障方法。CM 通过要求在 TOE 和相关信息的细化和修改过程中遵守一定的规章制度和采取一定的控制措施,以达到目的。CM 系统通过提供一种跟踪任何变化的方法和确保所有修改都是经过了授权的,并且确保它们所控制的 TOE 部分的完整性。

本族的目的是要求开发者的 CM 系统具有一定的能力。这些旨在减少配置项意外或未经授权修改发生的可能性。从早期设计阶段到后继维护的整个过程,CM 系统都应确保 TOE 的完整性。

引入自动 CM 工具的目的是为了提高 CM 系统的效率。虽然自动 CM 系统和手动 CM 系统都能被旁路或忽视,或者不能充分证明其可以阻止未授权的修改,至少自动的系统不易受人为错误或者疏忽的影响。

本族的目的包括以下几个方面:

- a) 确保 TOE 在发送给消费者之前是正确的和完备的;
- b) 确保在评估过程中没有遗漏配置项;
- c) 防止对 TOE 配置项进行未经授权地修改、增加或删除。

13.1.2 组件分级

本族中的组件是基于 CM 系统能力、开发者提供的 CM 文档范围以及证据进行分级的。

13.1.3 应用注释

虽然期望 CM 从早期设计阶段和其后的阶段都能适用,但本族要求至少在评估结束前部署并使用 CM。

当 TOE 是一个产品的子集时,本族要求仅适用于 TOE 配置项,而不是整个产品。

如果开发者已经将 CM 系统分隔成不同的生命周期阶段(例如开发,生产和/或最终产品),则文档应包含上述的所有阶段。评估时应将分隔的 CM 系统作为标准中所陈述的完整 CM 系统的组成部分。

类似的,如果 TOE 的组成部分是由不同开发者或者在不同场所生产的,使用在不同地点的 CM 系统应该作为标准中所陈述的完整 CM 系统的组成部分。在这种情况下,完整性方面也需要考虑。

本族的部分元素涉及配置项。上述元素利用配置项列表中的所有配置项识别 CM 要求,但配置列表的内容由开发者自行完善。通过标识那些必须包含在配置列表中并因此受 CM 控制的特定项,ALC_CMS“CM 范围”可用于限定开发者自行处理的能力。

ALC_CMC.2.3C 要求 CM 系统应唯一地标识所有配置项,也要求给修改后的配置项分配一个新的唯一标识符。

ALC_CMC.3.8C 要求证据应证实 CM 系统的运行与 CM 计划是一致的。举例如下:屏幕快照、CM 系统输出的审计迹之类的文档,或开发者提供的 CM 系统详细论证。评估者有责任确定这些证据是否足以表明 CM 系统的运行与 CM 计划是一致的。

ALC_CMC.4.5C 要求 CM 系统应提供一种自动化的方式支持 TOE 的生产。这就要求 CM 系统应提供一种自动化的方式来帮助确定在生成 TOE 时使用了正确的配置项。

ALC_CMC.5.10C 要求 CM 系统提供一种自动化的方式来确定 TOE 和它以前版本之间的变化情况。如果 TOE 不存在以前的版本,开发者仍需要提供一种自动化的方式来确定 TOE 和将来版本之间的变化。

13.1.4 ALC_CMC.1 TOE 标识

依赖关系:ALC_CMS.1 TOE CM 覆盖。

13.1.4.1 目的

需要一个唯一的参照号,以确保 TOE 实例在被评估时不会产生歧义。用该参照号为 TOE 做标注,以确保 TOE 用户能唯一确认所使用的 TOE 实例。

13.1.4.2 开发者行为元素

13.1.4.2.1 ALC_CMC.1.1D

开发者应提供 TOE 及其参照号。

13.1.4.3 内容和形式元素

13.1.4.3.1 ALC_CMC.1.1C

应给 TOE 标注唯一参照号。

13.1.4.4 评估者行为元素

13.1.4.4.1 ALC_CMC.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.1.5 ALC_CMC.2 CM 系统的使用

依赖关系:ALC_CMS.1 TOE CM 覆盖。

13.1.5.1 目的

需要一个唯一的参照号,以确保 TOE 实例在被评估时不会产生歧义。用该参照号为 TOE 做标注,以确保 TOE 用户能唯一确认所使用的 TOE 实例。

配置项的唯一标识可以使我们对 TOE 的组成有更清晰的理解,从而有助于确定哪些配置项满足 TOE 评估要求。

CM 系统的使用增强了对配置项以受控方式进行维护的保障。

13.1.5.2 开发者行为元素

13.1.5.2.1 ALC_CMC.2.1D

开发者应提供 TOE 及其参照号。

13.1.5.2.2 ALC_CMC.2.2D

开发者应提供 CM 文档。

13.1.5.2.3 ALC_CMC.2.3D

开发者应使用 CM 系统。

13.1.5.3 内容和形式元素

13.1.5.3.1 ALC_CMC.2.1C

应给 TOE 标注唯一参照号。

13.1.5.3.2 ALC_CMC.2.2C

CM 文档应描述用于唯一标识配置项的方法。

13.1.5.3.3 ALC_CMC.2.3C

CM 系统应唯一标识所有配置项。

13.1.5.4 评估者行为元素

13.1.5.4.1 ALC_CMC.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.1.6 ALC_CMC.3 授权控制

依赖关系:ALC_CMS.1 TOE CM 覆盖;
ALC_DVS.1 安全措施标识;
ALC_LCD.1 开发者定义的生命周期模型。

13.1.6.1 目的

需要一个唯一的参照号,以确保 TOE 实例在被评估时不会产生歧义。用该参照号为 TOE 做标注,以确保 TOE 用户能唯一确认所使用的 TOE 实例。

配置项的唯一标识可以使我们对 TOE 的组成有更清晰的理解,从而有助于确定哪些配置项满足 TOE 评估要求。

CM 系统的使用增强了对配置项以受控方式进行维护的保障。

提供控制机制以确保不会对 TOE 进行未授权地修改(“CM 访问控制”),并确保 CM 系统的正确功能和使用,有助于保持 TOE 的完整性。

13.1.6.2 开发者行为元素

13.1.6.2.1 ALC_CMC.3.1D

开发者应提供 TOE 及其参照号。

13.1.6.2.2 ALC_CMC.3.2D

开发者应提供 CM 文档。

13.1.6.2.3 ALC_CMC.3.3D

开发者应使用 CM 系统。

13.1.6.3 内容和形式元素

13.1.6.3.1 ALC_CMC.3.1C

应给 TOE 标注唯一参照号。

13.1.6.3.2 ALC_CMC.3.2C

CM 文档应描述用于唯一标识配置项的方法。

13.1.6.3.3 ALC_CMC.3.3C

CM 系统应唯一标识所有配置项。

13.1.6.3.4 ALC_CMC.3.4C

CM 系统应提供措施使得只能对配置项进行授权变更。

13.1.6.3.5 ALC_CMC.3.5C

CM 文档应包括一个 CM 计划。

13.1.6.3.6 ALC_CMC.3.6C

CM 计划应描述 CM 系统是如何应用于 TOE 的开发过程。

13.1.6.3.7 ALC_CMC.3.7C

证据应证实所有配置项都正在 CM 系统下进行维护。

13.1.6.3.8 ALC_CMC.3.8C

证据应证实 CM 系统的运行与 CM 计划是一致的。

13.1.6.4 评估者行为元素

13.1.6.4.1 ALC_CMC.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.1.7 ALC_CMC.4 生产支持和接受程序及其自动化

依赖关系:ALC_CMS.1 TOE CM 覆盖;

ALC_DVS.1 安全措施标识;

ALC_LCD.1 开发者定义的生命周期模型。

13.1.7.1 目的

需要一个唯一的参照号,以确保 TOE 实例在被评估时不会产生混淆。用该参照号为 TOE 做标注,以确保 TOE 用户能唯一确认所使用的 TOE 实例。

配置项的唯一标识可以使我们对 TOE 的组成有更清晰的理解,从而有助于确定哪些配置项满足 TOE 评估要求。

CM 系统的使用增强了对配置项以受控方式进行维护的保障。

提供控制机制以确保不会对 TOE 进行未授权地修改(“CM 访问控制”),并确保 CM 系统的正确功能和使用,有助于保持 TOE 的完整性。

接受程序的目的是确保 TOE 的各组成部分的质量,并且确保配置项的任何建立或修改都是已授权的。接受程序是 TOE 集成过程和生命周期管理的一个基本元素。

在开发环境中,配置项是复杂的,没有自动化工具的支持难以控制各种变化。特别是这些自动化的

工具必须能够应对开发期间出现的各种变化,并且保证这些变化都是已授权的。本组件的目的是确保通过自动化手段控制配置项。如果 TOE 是由多个开发者开发的,需要对 TOE 进行集成,自动化工具的使用才是足够的。

生产支持程序有助于保证从一个可管理的配置项集合以授权的方式正确生成 TOE,尤其是当涉及不同的开发者,而且必须执行集成过程的时候。

13.1.7.2 开发者行为元素

13.1.7.2.1 ALC_CMC.4.1D

开发者应提供 TOE 及其参照号。

13.1.7.2.2 ALC_CMC.4.2D

开发者应提供 CM 文档。

13.1.7.2.3 ALC_CMC.4.3D

开发者应使用 CM 系统。

13.1.7.3 内容和形式元素

13.1.7.3.1 ALC_CMC.4.1C

应给 TOE 标记唯一参照号。

13.1.7.3.2 ALC_CMC.4.2C

CM 文档应描述用于唯一标识配置项的方法。

13.1.7.3.3 ALC_CMC.4.3C

CM 系统应唯一标识所有配置项。

13.1.7.3.4 ALC_CMC.4.4C

CM 系统应提供自动化的措施使得只能对配置项进行授权变更。

13.1.7.3.5 ALC_CMC.4.5C

CM 系统应以自动化的方式支持 TOE 的生产。

13.1.7.3.6 ALC_CMC.4.6C

CM 文档应包括 CM 计划。

13.1.7.3.7 ALC_CMC.4.7C

CM 计划应描述 CM 系统是如何应用于 TOE 的开发的。

13.1.7.3.8 ALC_CMC.4.8C

CM 计划应描述用来接受修改过的或新创建的作为 TOE 组成部分的配置项的程序。

13.1.7.3.9 ALC_CMC.4.9C

证据应证实所有配置项都正在 CM 系统下进行维护。

13.1.7.3.10 ALC_CMC.4.10C

证据应证实 CM 系统的运行与 CM 计划是一致的。

13.1.7.4 评估者行为元素

13.1.7.4.1 ALC_CMC.4.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.1.8 ALC_CMC.5 高级支持

依赖关系:ALC_CMS.1 TOE CM 覆盖;
ALC_DVS.2 充分的安全措施;
ALC_LCD.1 开发者定义的生命周期模型。

13.1.8.1 目的

需要一个唯一的参照号,以确保 TOE 实例在被评估时不会产生混淆。用该参照号为 TOE 做标注,以确保 TOE 用户能唯一确认所使用的 TOE 实例。

配置项的唯一标识可以使我们对 TOE 的组成有更清晰的理解,从而有助于确定哪些配置项满足 TOE 评估要求。

CM 系统的使用增强了对配置项以受控方式进行维护的保障。

提供控制机制以确保不会对 TOE 进行未经授权地修改(“CM 访问控制”),并确保 CM 系统的正确功能和使用,有助于保证 TOE 的完整性。

接受程序的目的是确保 TOE 的各组成部分的质量,并且确保配置项的任何建立或修改都是已授权的。接受程序是 TOE 集成过程和生命周期管理的一个基本元素。

在开发环境中,配置项是复杂的,没有自动化工具的支持难以控制各种变化。特别是这些自动化的工具必须能够应付开发期间出现的多种变化,并且保证这些变化都是已授权的。本组件的目的是确保通过自动化手段控制配置项。如果 TOE 是由多个开发者开发的,需要对 TOE 进行集成,自动化工具的使用才是足够的。

生产支持程序有助于保证从一个可管理的配置项集合以授权的方式正确生成 TOE,尤其是当涉及不同的开发者,而且必须执行集成过程的时候。

要求 CM 系统能够标识用于生成 TOE 的实现表示的版本,有助于确保这些材料的完整性是受适当的技术、物理和程序上的安全措施保护的。

提供一种自动化的方式确定 TOE 版本间的变化,并标识哪些配置项会受其他配置项修改的影响,这有助于确定 TOE 后续版本间变化的影响。如此依次进行能提供一些有价值的信息,以确定 TOE 变更后所有配置项是否还相互一致。

13.1.8.2 开发者行为元素

13.1.8.2.1 ALC_CMC.5.1D

开发者应提供 TOE 及其参照号。

13.1.8.2.2 ALC_CMC.5.2D

开发者应提供 CM 文档。

13.1.8.2.3 ALC_CMC.5.3D

开发者应使用 CM 系统。

13.1.8.3 内容和形式元素

13.1.8.3.1 ALC_CMC.5.1C

应给 TOE 标记唯一参照号。

13.1.8.3.2 ALC_CMC.5.2C

CM 文档应描述用于唯一标识配置项的方法。

13.1.8.3.3 ALC_CMC.5.3C

CM 文档应论证接受程序对所有配置项的变更都提供了充分且适当的复查。

13.1.8.3.4 ALC_CMC.5.4C

CM 系统应唯一标识所有配置项。

13.1.8.3.5 ALC_CMC.5.5C

CM 系统应提供自动化的措施使得只能对配置项进行授权改变。

13.1.8.3.6 ALC_CMC.5.6C

CM 系统应以自动化的方式支持 TOE 的生产。

13.1.8.3.7 ALC_CMC.5.7C

CM 系统应确保负责将某个配置项接受到 CM 中的人不是开发此配置项的人。

13.1.8.3.8 ALC_CMC.5.8C

CM 系统应标识组成 TSF 的配置项。

13.1.8.3.9 ALC_CMC.5.9C

CM 系统应以自动化的方式支持 TOE 所有变化的审计,审计迹中要包括源发者、日期和时间等信息。

13.1.8.3.10 ALC_CMC.5.10C

CM 系统应提供自动化的方式标识受已给定配置项的变化影响的所有其他配置项。

13.1.8.3.11 ALC_CMC.5.11C

CM 系统应能标识用于生成 TOE 的实现表示的版本。

13.1.8.3.12 ALC_CMC.5.12C

CM 文档应包括 CM 计划。

13.1.8.3.13 ALC_CMC.5.13C

CM 计划应描述 CM 系统是如何应用于 TOE 开发过程。

13.1.8.3.14 ALC_CMC.5.14C

CM 计划应描述用来接受修改过的或新创建的作为 TOE 组成部分的配置项的程序。

13.1.8.3.15 ALC_CMC.5.15C

证据应证实所有配置项都正在 CM 系统下进行维护。

13.1.8.3.16 ALC_CMC.5.16C

证据应证实 CM 系统的运行与 CM 计划是一致的。

13.1.8.4 评估者行为元素

13.1.8.4.1 ALC_CMC.5.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.1.8.4.2 ALC_CMC.5.1E

评估者应确定开发者提供用于测试活动的 TOE 的生产支持程序的应用。

13.2 CM 范围(ALC_CMS)

13.2.1 目的

本族的目的是标识出纳入配置管理系统的配置项,并满足 ALC_CMC“CM 能力”族提出的 CM 要求。将配置管理应用到这些额外项目为 TOE 完整性维护提供了额外保障。

13.2.2 组件分级

本族中的组件是基于下列配置项进行分级的:TOE 及安全保障要求的评估证据、TOE 的组成部分、实现表示、安全缺陷、开发工具和相关信息。

13.2.3 应用注释

虽然 ALC_CMS“CM 范围”强制要求配置项列表并且该列表中的每一项都受 CM 控制,ALC_CMC“CM 能力”将配置项列表的具体内容留给开发者自行处理。ALC_CMS“CM 范围”通过识别那些必须包含到配置项列表中因而需满足 ALC_CMC“CM 能力”提出的 CM 要求的项目,限定开发者自行处理的能力。

13.2.4 ALC_CMS.1 TOE CM 覆盖

依赖关系:无。

13.2.4.1 目的

CM 系统只能控制处于 CM 下配置项(即在控制项列表中标识的配置项)的改变。将 TOE 本身和 ST 中其他的安全保障要求的评估证据置于 CM 之下,可以确保它们的修改是在一个带正确授权的受控方式下进行的。

13.2.4.2 应用注释

ALC_CMS.1.1C 要求 TOE 本身和 ST 中其他的安全保障要求的评估证据必须包含在配置项列表

中,因而需要满足 ALC_CMC“CM 能力”提出的 CM 要求。

13.2.4.3 开发者行为元素

13.2.4.3.1 ALC_CMS.1.1D

开发者应提供 TOE 配置项列表。

13.2.4.4 内容和形式元素

13.2.4.4.1 ALC_CMS.1.1C

配置项列表应包括:TOE 本身和安全保障要求的评估证据。

13.2.4.4.2 ALC_CMS.1.2C

配置项列表应唯一标识配置项。

13.2.4.5 评估者行为元素

13.2.4.5.1 ALC_CMS.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.2.5 ALC_CMS.2 部分 TOE CM 覆盖

依赖关系:无。

13.2.5.1 目的

CM 系统只能控制处于 CM 下配置项(即在控制项列表中标识的配置项)的改变。将 TOE 本身、TOE 组成部分和其他安全保障要求所需的评估证据置于 CM 之下,可以确保它们的修改是在一个带正确授权的受控方式下进行的。

13.2.5.2 应用注释

ALC_CMS.2.1C 要求 TOE 的组成部分(交付给消费者的所有部分,例如硬件部分或可执行文件)必须包含在配置项列表中,因而需要满足 ALC_CMC“CM 能力”提出的 CM 要求。

ALC_CMS.2.3C 要求配置项列表要简要说明每一个 TSF 相关配置项的开发者。这里的“开发者”不是指某个人,而是负责开发该配置项的组织。

13.2.5.3 开发者行为元素

13.2.5.3.1 ALC_CMS.2.1D

开发者应提供 TOE 配置项列表。

13.2.5.4 内容和形式元素

13.2.5.4.1 ALC_CMS.2.1C

配置项列表应包括:TOE 本身、安全保障要求的评估证据和 TOE 的组成部分。

13.2.5.4.2 ALC_CMS.2.2C

配置项列表应唯一标识配置项。

13.2.5.4.3 ALC_CMS.2.3C

对于每一个 TSF 相关的配置项,配置项列表应简要说明该配置项的开发者。

13.2.5.5 评估者行为元素

13.2.5.5.1 ALC_CMS.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.2.6 ALC_CMS.3 实现表示 CM 覆盖

依赖关系:无。

13.2.6.1 目的

CM 系统只能控制处于 CM 下配置项(即在控制项列表中标识的配置项)的改变。将 TOE 本身、TOE 组成部分、TOE 实现表示和其他安全保障要求所需的评估证据置于 CM 之下,可以确保它们的修改是在一个带正确授权的受控方式下进行的。

13.2.6.2 应用注释

ALC_CMS.3.1C 要求 TOE 的实现表示必须包含在配置项列表中,因而需要满足 ALC_CMC“CM 能力”提出的 CM 要求。

13.2.6.3 开发者行为元素

13.2.6.3.1 ALC_CMS.3.1D

开发者应提供 TOE 配置项列表。

13.2.6.4 内容和形式元素

13.2.6.4.1 ALC_CMS.3.1C

配置项列表应包括:TOE 本身、安全保障要求的评估证据、TOE 的组成部分和实现表示。

13.2.6.4.2 ALC_CMS.3.2C

配置项列表应唯一标识配置项。

13.2.6.4.3 ALC_CMS.3.3C

对于每一个 TSF 相关的配置项,配置项列表应简要说明该配置项的开发者。

13.2.6.5 评估者行为元素

13.2.6.5.1 ALC_CMS.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.2.7 ALC_CMS.4 问题跟踪 CM 覆盖

依赖关系:无。

13.2.7.1 目的

CM 系统只能控制处于 CM 下配置项(即在控制项列表中标识的配置项)的改变。将 TOE 本身、TOE 的组成部分、TOE 的实现表示和其他安全保障要求所需的评估证据置于 CM 之下,可以确保它们的修改是在一个带正确授权的受控方式下进行的。

将安全缺陷置于 CM 之下,确保不会丢失或者遗忘安全缺陷报告,并且允许开发者跟踪安全缺陷,找到解决方案。

13.2.7.2 应用注释

ALC_CMS.4.1C 要求安全缺陷必须包含在配置项列表中,因而需要满足 ALC_CMC“CM 能力”提出的 CM 要求。这将要求维护有关以前的安全缺陷及其解决方案的信息和当前安全缺陷的详细信息。

13.2.7.3 开发者行为元素

13.2.7.3.1 ALC_CMS.4.1D

开发者应提供 TOE 配置项列表。

13.2.7.4 内容和形式元素

13.2.7.4.1 ALC_CMS.4.1C

配置项列表应包括:TOE 本身、安全保障要求的评估证据、TOE 的组成部分、实现表示 **和安全缺陷报告及其解决状态**。

13.2.7.4.2 ALC_CMS.4.2C

配置项列表应唯一标识配置项。

13.2.7.4.3 ALC_CMS.4.3C

对于每一个 TSF 相关的配置项,配置项列表应简要说明该配置项的开发者。

13.2.7.5 评估者行为元素

13.2.7.5.1 ALC_CMS.4.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.2.8 ALC_CMS.5 开发工具 CM 覆盖

依赖关系:无。

13.2.8.1 目的

CM 系统只能控制处于 CM 下配置项(即在控制项列表中标识的配置项)的改变。将 TOE 本身、TOE 的组成部分、TOE 的实现表示和其他安全保障要求所需的评估证据置于 CM 之下,可以确保它们的修改是在一个带正确授权的受控方式下进行的。

将安全缺陷置于 CM 之下,确保不会丢失或者遗忘安全缺陷报告,并且允许开发者跟踪安全缺陷,找到其解决方案。

开发工具对于确保产生 TOE 的高质量版本十分重要。因此,对这些工具的修改进行控制是很重

要的。

13.2.8.2 应用注释

ALC_CMS.5.1 要求开发工具及其相关信息必须包含在配置项列表中,因而需要满足 ALC_CMC “CM 能力”提出的 CM 要求。编程语言和编译工具是开发工具的例子。关于 TOE 生成的选项(例如编译选项、生成选项和构建选项)是开发工具相关信息的一个例子。

13.2.8.3 开发者行为元素

13.2.8.3.1 ALC_CMS.5.1D

开发者应提供 TOE 配置项列表。

13.2.8.4 内容和形式元素

13.2.8.4.1 ALC_CMS.5.1C

配置项列表应包括:TOE 本身、安全保障要求的评估证据、TOE 的组成部分、实现表示、安全缺陷报告及其解决状态、开发工具及其相关信息。

13.2.8.4.2 ALC_CMS.5.2C

配置项列表应唯一标识配置项。

13.2.8.4.3 ALC_CMS.5.3C

对于每一个 TSF 相关的配置项,配置项列表应简要说明该配置项的开发者。

13.2.8.5 评估者行为元素

13.2.8.5.1 ALC_CMS.5.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.3 交付(ALC_DEL)

13.3.1 目的

本族关注的是如何将完成的 TOE 从开发环境安全传递到负责接受的用户手中。

交付要求需要详细说明系统控制、分发工具和程序等必要措施,确保在向用户分发 TOE 期间 TOE 的安全性得到维护。对于 TOE 的一个有效分发,用于分发 TOE 的程序负责处理 PP/ST 中标识的在交付过程中与 TOE 安全性相关的客观事实。

13.3.2 组件分级

本族只包含一个组件。一个增强的保护级别是通过要求交付程序与 AVA_VAN“脆弱性分析”族中假设的潜在攻击对应而建立的。

13.3.3 应用注释

在本族中不考虑从子承包商到开发者或不同开发地点之间的传递,这些在 ALC_DVS“开发安全”族中考虑。

将 TOE 移交由用户负责后,而不一定要将 TOE 送达到用户地点,即标志着交付阶段的结束。

可用的准备程序应考虑以下几个方面：

- a) 确保消费者所接收到的 TOE 正好与被评估的 TOE 版本一致；
- b) 避免/检测对 TOE 现行版本的任何篡改；
- c) 防止提交一个错误的 TOE 版本；
- d) 避免让消费者了解不必要的 TOE 分发知识：可能存在潜在攻击者不知道交付时间和如何交付的情形；
- e) 避免/检测 TOE 在交付过程中被中途截取；和
- f) 避免 TOE 在分发时被延迟或中止。

交付程序应包括上述方面暗含的可接受行为。如果提交，这些暗含的行为的一致性描述在 AGD_PRE“准备程序”族中检查。

13.3.4 ALC_DEL.1 交付程序

依赖关系：无。

13.3.4.1 开发者行为元素

13.3.4.1.1 ALC_DEL.1.1D

开发者应将把 TOE 或其部分交付给消费者的程序文档化。

13.3.4.1.2 ALC_DEL.1.2D

开发者应使用交付程序。

13.3.4.2 内容和形式元素

13.3.4.2.1 ALC_DEL.1.1C

交付文档应描述，在向消费者分发 TOE 版本时，用以维护安全性所必需的所有程序。

13.3.4.3 评估者行为元素

13.3.4.3.1 ALC_DEL.1.1E

评估者应确认所提供的信息满足证据的内容和形式的要求。

13.4 开发安全(ALC_DVS)

13.4.1 目的

开发安全与物理的、程序的、人员的以及其他的为保护 TOE 或者其部分而在开发环境中采用的安全措施有关。包括开发场地的物理安全和任何用于选择开发人员的程序。

13.4.2 组件分级

本族中的组件是基于所要求的安全措施的充分性是否需要论证而分级的。

13.4.3 应用注释

本族涉及用于消除或减少开发场地存在的威胁的一些措施。

评估者为了评估开发安全证据应该检查开发场地。包括 TOE 开发和生产涉及的子承包商场地。任何不检查的决定必须征得评估机构的同意。

尽管开发安全涉及 TOE 的维护并且因而同时涉及评估完成后的相关方面,但 ALC_DVS“开发安全”要求仅说明那些在评估时实施的开发安全措施。此外,ALC_DVS“开发安全”不包含任何评估完成之后发起者有意使用的开发安全措施的相关要求。

应该承认,在 TOE 的开发环境中机密性并不总是涉及 TOE 保护的问题。“必需的”一词的使用,是考虑到要选择适当的防护措施。

13.4.4 ALC_DVS.1 安全措施标识

依赖关系:无。

13.4.4.1 开发者行为元素

13.4.4.1.1 ALC_DVS.1.1D

开发者应提供开发安全文档。

13.4.4.2 内容和形式元素

13.4.4.2.1 ALC_DVS.1.1C

开发安全文档应描述在 TOE 的开发环境中,保护 TOE 设计和实现的机密性和完整性所必需的所有物理的、程序的、人员的及其他方面的安全措施。

13.4.4.3 评估者行为元素

13.4.4.3.1 ALC_DVS.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.4.4.3.2 ALC_DVS.1.2E

评估者应确认安全措施正在被使用。

13.4.5 ALC_DVS.2 充分的安全措施

依赖关系:无。

13.4.5.1 开发者行为元素

13.4.5.1.1 ALC_DVS.2.1D

开发者应提供开发安全文档。

13.4.5.2 内容和形式元素

13.4.5.2.1 ALC_DVS.2.1C

开发安全文档应描述在 TOE 的开发环境中,保护 TOE 设计和实现的机密性和完整性所必需的所有物理的、程序的、人员的及其他方面的安全措施。

13.4.5.2.2 ALC_DVS.2.2C

开发安全文档应论证安全措施提供了必需的保护级别以维护 TOE 的机密性和完整性。

13.4.5.3 评估者行为元素

13.4.5.3.1 ALC_DVS.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.4.5.3.2 ALC_DVS.2.2E

评估者应确认安全措施正在被使用。

13.5 缺陷纠正(ALC_FLR)

13.5.1 目的

缺陷纠正要求开发者跟踪并纠正已发现的安全缺陷。尽管在 TOE 评估时不能确定将来是否遵从缺陷纠正程序,但评估开发者用于跟踪、纠正缺陷,并发布缺陷信息和纠正措施而采取的策略和程序是可行的。

13.5.2 组件分级

本族中的组件是基于缺陷纠正程序的范围不断扩大和缺陷纠正策略的严格性不断加深而分级的。

13.5.3 应用注释

本族通过要求 TOE 的开发者跟踪并纠正 TOE 的缺陷,保障将来对 TOE 进行维护和支持。除此以外,还包括一些关于缺陷纠正措施发布的要求。可是,本族不强行实施超出当前评估以外的评估要求。

认为 TOE 用户是用户组织中负责接收并实现安全缺陷修复的焦点。TOE 用户不必是一个个人用户,可以是一个负责处理安全缺陷的组织代表。术语“TOE 用户”的使用认可不同组织拥有不同的处理缺陷报告的程序,这些程序可以由一个个人用户实施,也可以由一个中心管理机构实施。

缺陷纠正程序应当描述用于处理所有遇到的缺陷类型的方法。这些缺陷可以是开发者、TOE 用户或其他熟悉 TOE 的团队报告的。某些缺陷可能不能立即被修复。某些情况下,缺陷可能不能被修复,必须采取其他的措施(例如程序上的)来补救。所提供的文档应当包括为运行场所提供缺陷修复的程序,以及当修复被推迟(在过渡期间应采取的措施)或缺陷已不可能修补时提供缺陷信息的程序。

尽管原先评估的一些信息可能仍然适用,但在 TOE 发布之后对其进行变更,则该 TOE 即是未经评估的。本族中所使用的术语“TOE 发布版本”是指一个已经过认证的 TOE 的发布版本,对它的变更都已经被接受。

13.5.4 ALC_FLR.1 基本的缺陷纠正

依赖关系:无。

13.5.4.1 开发者行为元素

13.5.4.1.1 ALC_FLR.1.1D

开发者应文档化缺陷纠正程序并提交给 TOE 开发者。

13.5.4.2 内容和形式元素

13.5.4.2.1 ALC_FLR.1.1C

缺陷纠正程序文档应描述用于跟踪每一个 TOE 发布版本中所有已报告安全缺陷的程序。

13.5.4.2.2 ALC_FLR.1.2C

缺陷纠正程序应要求描述所提供的每个安全缺陷的性质和影响,以及缺陷修正的情况。

13.5.4.2.3 ALC_FLR.1.3C

缺陷纠正程序应要求标识对每一个安全缺陷所采取的修正动作。

13.5.4.2.4 ALC_FLR.1.4C

缺陷纠正程序文档应描述用于提供缺陷信息、修正物和修正动作指南给 TOE 用户的方法。

13.5.4.3 评估者行为元素

13.5.4.3.1 ALC_FLR.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.5.5 ALC_FLR.2 缺陷报告程序

依赖关系:无。

13.5.5.1 目的

为了让开发者能够对来自 TOE 用户的安全缺陷报告采取适当的动作,并且知道该向谁发送修正补丁,TOE 用户需要了解如何将安全缺陷报告递交给开发者。开发者将缺陷纠正指南提供给 TOE 用户,确保 TOE 用户知道这一重要信息。

13.5.5.2 开发者行为元素

13.5.5.2.1 ALC_FLR.2.1D

开发者应文档化缺陷纠正程序并提交给 TOE 开发者。

13.5.5.2.2 ALC_FLR.2.2D

开发者应建立一个程序来接受和处理所有的安全缺陷报告并要求修正这些缺陷。

13.5.5.2.3 ALC_FLR.2.3D

开发者应提供缺陷纠正指南给 TOE 用户。

13.5.5.3 内容和形式元素

13.5.5.3.1 ALC_FLR.2.1C

缺陷纠正程序文档应描述用于跟踪每一个 TOE 发布版本中所有已报告安全缺陷的程序。

13.5.5.3.2 ALC_FLR.2.2C

缺陷纠正程序应要求描述所提供的每个安全缺陷的性质和影响,以及缺陷修正的情况。

13.5.5.3.3 ALC_FLR.2.3C

缺陷纠正程序应要求标识对每一个安全缺陷所采取的修正动作。

13.5.5.3.4 ALC_FLR.2.4C

缺陷纠正程序文档应描述用于提供缺陷信息、修正和修正动作指南给 TOE 用户的方法。

13.5.5.3.5 ALC_FLR.2.5C

缺陷纠正程序应描述开发者接收 TOE 用户报告并询问 TOE 中可疑安全缺陷的手段。

13.5.5.3.6 ALC_FLR.2.6C

处理所报告的安全缺陷的程序应确保任何已报告的缺陷都已被纠正,并且纠正程序已发布给 TOE 用户。

13.5.5.3.7 ALC_FLR.2.7C

处理所报告的安全缺陷的程序应提供预防措施,确保对这些安全缺陷的任何修正都不会引入任何新的缺陷。

13.5.5.3.8 ALC_FLR.2.8C

缺陷纠正指南应描述 TOE 用户将 TOE 中任何可疑安全缺陷报告给开发者的手段。

13.5.5.4 评估者行为元素

13.5.5.4.1 ALC_FLR.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.5.6 ALC_FLR.3 系统的缺陷纠正

依赖关系:无。

13.5.6.1 目的

为了让开发者能够对来自 TOE 用户的安全缺陷报告采取适当的动作,并且知道该向谁发送修正补丁,TOE 用户需要了解如何将安全缺陷报告递交给开发者,以及他们如何向开发者注册,以便能接收到修正补丁。开发者将缺陷纠正指南提供给 TOE 用户,确保 TOE 用户知道这一重要信息。

13.5.6.2 开发者行为元素

13.5.6.2.1 ALC_FLR.3.1D

开发者应文档化缺陷纠正程序并提交给 TOE 开发者。

13.5.6.2.2 ALC_FLR.3.2D

开发者应建立一个程序来接受和处理所有的安全缺陷报告并要求修正这些缺陷。

13.5.6.2.3 ALC_FLR.3.3D

开发者应提供缺陷纠正指南给 TOE 用户。

13.5.6.3 内容和形式元素

13.5.6.3.1 ALC_FLR.3.1C

缺陷纠正程序文档应描述用于跟踪每一个 TOE 发布版本中所有已报告安全缺陷的程序。

13.5.6.3.2 ALC_FLR.3.2C

缺陷纠正程序应要求描述所提供的每个安全缺陷的性质和影响,以及缺陷修正的情况。

13.5.6.3.3 ALC_FLR.3.3C

缺陷纠正程序应要求标识对每一个安全缺陷所采取的修正动作。

13.5.6.3.4 ALC_FLR.3.4C

缺陷纠正程序文档应描述用于提供缺陷信息、修正和修正动作指南给 TOE 用户的方法。

13.5.6.3.5 ALC_FLR.3.5C

缺陷纠正程序应描述开发者接收 TOE 用户报告并询问 TOE 中可疑安全缺陷的手段。

13.5.6.3.6 ALC_FLR.3.6C

缺陷纠正程序应包含一个程序,要求及时反应并且自动分发安全缺陷报告和相应的修正给可能受到安全缺陷影响的注册用户。

13.5.6.3.7 ALC_FLR.3.7C

处理所报告的安全缺陷的程序应确保任何已报告的缺陷都已被纠正,并且纠正程序已发布给 TOE 用户。

13.5.6.3.8 ALC_FLR.3.8C

处理所报告的安全缺陷的程序应提供预防措施,确保对这些安全缺陷的任何修正都不会引入任何新的缺陷。

13.5.6.3.9 ALC_FLR.3.9C

缺陷纠正指南应描述 TOE 用户将 TOE 中任何可疑安全缺陷报告给开发者的手段。

13.5.6.3.10 ALC_FLR.3.10C

缺陷纠正指南应描述 TOE 用户向开发者注册以便有资格接收安全缺陷报告和修正的手段。

13.5.6.3.11 ALC_FLR.3.11C

缺陷纠正指南应标识特定的联系点,以便完全地报告和询问涉及 TOE 的安全问题。

13.5.6.4 评估者行为元素

13.5.6.4.1 ALC_FLR.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.6 生命周期定义(ALC_LCD)

13.6.1 目的

TOE 的开发和维护缺乏控制将导致 TOE 不能满足它的所有安全要求。因此,尽早在 TOE 生命周期内建立 TOE 开发和维护模型是很重要的。

使用 TOE 开发和维护模型并不能确保 TOE 满足它的所有安全功能要求。可能选择的模型是不胜任的或不适当的,因此无助于提高 TOE 的质量。使用一个已得到一组专家(例如技术专家、标准化组织)认可的生命周期模型,将提高开发和维护模型使 TOE 满足它的所有安全功能要求的机会。含有一些定量评估方法的生命周期模型的使用能进一步保障 TOE 开发过程的整体质量。

13.6.2 组件分级

本族中的组件是基于关于生命周期模型的可测性和对该模型的遵从性的要求不断增加而分级的。

13.6.3 应用注释

生命周期模型包括用于开发和维护 TOE 的程序、工具和技术。可被这样一个模型涵盖的过程包括设计方法、审查程序、项目管理控制、变化控制程序、测试方法和接受程序。一个有效的生命周期模型将在一个分配职责并监控进度的总体管理框架内,处理开发和维护过程的以上各个方面。

不同类型的接受情况由 ISO/IEC 15408 的不同族处理:子承包商部分交付的验收应由 ALC_LCD“生命周期定义”族处理,接受后到内部的运输由 ALC_DVS“开发安全”族处理,接受部分纳入 CM 系统在 ALC_CMC“CM 能力”中处理,交付 TOE 给消费者的验收在 ALC_DEL“交付”族中处理。前三种验收可以交叉使用。

虽然生命周期定义涉及 TOE 的维护,且同时涉及评估完成后的相关方面,但 TOE 的评估需要通过评估时所提供的 TOE 生命周期信息加以分析来增强保障。

如果生命周期模型能够充分降低 TOE 不满足它的所有安全要求的危险性,那么该生命周期模型就为 TOE 的开发和维护提供了必要的控制。

一个可测量的生命周期模型是一个为了测量产品的开发特性而使用了管理产品的定量评估方法(算术参数和/或量度)的模型。典型的量度是源代码复杂性量度、缺陷率(每定量代码的错误数)或者平均失败次数。为了安全性评估这些量度都是相关的,可用于降低缺陷率以提高质量,并因此最终增加了 TOE 的安全保障。

生命周期模型,一方面应考虑已存在的标准的生命周期模型(像瀑布模型),另一方面考虑标准的量度(像错误率),可以是二者结合。ISO/IEC 15408 不要求生命周期完全地遵从某一个标准去定义这两方面。

13.6.4 ALC_LCD.1 开发者定义的生命周期模型

依赖关系:无。

13.6.4.1 开发者行为元素

13.6.4.1.1 ALC_LCD.1.1D

开发者应建立一个生命周期模型,用于 TOE 的开发和维护。

13.6.4.1.2 ALC_LCD.1.2D

开发者应提供生命周期定义文档。

13.6.4.2 内容和形式元素

13.6.4.2.1 ALC_LCD.1.1C

生命周期定义文档应描述用于开发和维护 TOE 的模型。

13.6.4.2.2 ALC_LCD.1.2C

生命周期模型应为 TOE 的开发和维护提供必要的控制。

13.6.4.3 评估者行为元素

13.6.4.3.1 ALC_LCD.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.6.5 ALC_LCD.2 可测量的生命周期模型

依赖关系:无。

13.6.5.1 开发者行为元素

13.6.5.1.1 ALC_LCD.2.1D

开发者应基于一个可测量的生命周期模型,建立一个生命周期模型,用于 TOE 的开发和维护。

13.6.5.1.2 ALC_LCD.2.2D

开发者应提供生命周期定义文档。

13.6.5.1.3 ALC_LCD.2.3D

开发者应使用可测量生命周期模型来测量 TOE 的开发。

13.6.5.1.4 ALC_LCD.2.4D

开发者应提供生命周期输出文档。

13.6.5.2 内容和形式元素

13.6.5.2.1 ALC_LCD.2.1C

生命周期定义文档应描述用于开发和维护 TOE 的模型,包括用于测量 TOE 的质量及/或其开发的算术参数和/或量度的一些细节。

13.6.5.2.2 ALC_LCD.2.2C

生命周期模型应为 TOE 的开发和维护提供必要的控制。

13.6.5.2.3 ALC_LCD.2.3C

生命周期输出文档应提供运用可测量的生命周期模型测量 TOE 开发的结果。

13.6.5.3 评估者行为元素

13.6.5.3.1 ALC_LCD.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.7 工具和技术(ALC_TAT)

13.7.1 目的

本族主要是选择用于开发、分析和实现 TOE 的工具。它包括一些要求,以防止将定义存在问题、

不稳定或者不正确的开发工具用于开发 TOE。这里包括但不限于,编程语言、文档、实现标准,以及 TOE 其他部分如支持运行库。

13.7.2 组件分级

本族中的组件是基于关于实现标准和实现依赖选项文档的描述和范围要求的不断增加而分级的。

13.7.3 应用注释

这里对明确定义的开发工具有一个要求,即它们是描述清晰且完整的工具。例如,编程语言和计算机辅助设计(CAD)系统是基于标准机构公开发布的一个标准,因此认为是明确定义的。个人制作的工具需要进一步调查研究以阐明是否被明确定义。

ALC_TAT.1.2C 中的要求尤其适用于编程语言,以便保证源代码中的所有语句都无歧义。

在 ALC_TAT.2 和 ALC_TAT.3 中,如果实现指南已经被专家组(例如学术专家、标准组织)认可了,即可以接受作为实现标准。实现标准通常是某个特定行业普遍公用的、容易接受的、普遍使用的,但开发者特定的实现指南也可以接受为标准,这里强调的是专业技能。

工具和技术区分开发者使用的实现标准(ALC_TAT.2.3D)和包括第三方软件、硬件或固件的“TOE 的所有部分”的实现标准(ALC_TAT.3.3D)。ALC_CMS“CM 范围”引入的配置项列表要求为每一个 TSF 相关配置项说明是否是由 TOE 开发者或第三方开发者生成的。

13.7.4 ALC_TAT.1 明确定义的开发工具

依赖关系:ADV_IMP.1 TSF 实现表示。

13.7.4.1 开发者行为元素

13.7.4.1.1 ALC_TAT.1.1D

开发者应标识用于开发 TOE 的每个工具。

13.7.4.1.2 ALC_TAT.1.2D

开发者应在文档中描述每个开发工具所选取的实现依赖选项。

13.7.4.2 内容和形式元素

13.7.4.2.1 ALC_TAT.1.1C

用于实现的每个开发工具都应明确定义的。

13.7.4.2.2 ALC_TAT.1.2C

每个开发工具的文档应无歧义地定义所有语句和实现用到的所有协定与命令的含义。

13.7.4.2.3 ALC_TAT.1.3C

每个开发工具的文档应无歧义地定义所有实现依赖选项的含义。

13.7.4.3 评估者行为元素

13.7.4.3.1 ALC_TAT.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.7.5 ALC_TAT.2 遵从实现标准

依赖关系:ADV_IMP.1 TSF 实现表示。

13.7.5.1 开发者行为元素

13.7.5.1.1 ALC_TAT.2.1D

开发者应标识用于开发 TOE 的每个工具。

13.7.5.1.2 ALC_TAT.2.2D

开发者应在文档中描述每个开发工具所选取的实现依赖选项。

13.7.5.1.3 ALC_TAT.2.3D

开发者应描述开发者所使用的实现标准。

13.7.5.2 内容和形式元素

13.7.5.2.1 ALC_TAT.2.1C

用于实现的每个开发工具都应是明确定义的。

13.7.5.2.2 ALC_TAT.2.2C

每个开发工具的文档应无歧义地定义所有语句的含义,以及实现用到的所有协定与指令。

13.7.5.2.3 ALC_TAT.2.3C

每个开发工具的文档应无歧义地定义所有实现依赖选项的含义。

13.7.5.3 评估者行为元素

13.7.5.3.1 ALC_TAT.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.7.5.3.2 ALC_TAT.2.2E

评估者应确认已经采用实现标准。

13.7.6 ALC_TAT.3 遵从实现标准—所有部分

依赖关系:ADV_IMP.1 TSF 实现表示。

13.7.6.1 开发者行为元素

13.7.6.1.1 ALC_TAT.3.1D

开发者应标识用于开发 TOE 的每个工具。

13.7.6.1.2 ALC_TAT.3.2D

开发者应在文档中描述每个开发工具所选取的实现依赖选项。

13.7.6.1.3 ALC_TAT.3.3D

开发者应描述开发者和任何 TOE 所有部分的第三方提供商所使用的实现标准。

13.7.6.2 内容和形式元素

13.7.6.2.1 ALC_TAT.3.1C

用于实现的每个开发工具都应是明确定义的。

13.7.6.2.2 ALC_TAT.3.2C

每个开发工具的文档应无歧义地定义所有语句和实现用到的所有协定与命令的含义。

13.7.6.2.3 ALC_TAT.3.3C

每个开发工具的文档应无歧义地定义所有实现依赖选项的含义。

13.7.6.3 评估者行为元素

13.7.6.3.1 ALC_TAT.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

13.7.6.3.2 ALC_TAT.3.2E

评估者应确认实现标准已被采用。

14 ATE 类:测试

“测试”类包括四个族:覆盖(ATE_COV),深度(ATE_DPT),独立测试(例如由评估者执行的功能测试)(ATE_IND),功能测试(ATE_FUN)。测试能为 TSF 是否符合前面所述(功能规范,TOE 设计及实现描述)提供保障。

本类的重点在于确认 TSF 依据其规范进行运作。本类不涉及穿透性测试,穿透性测试是基于对 TSF 的分析专门查找并标识 TSF 设计与实现过程中的脆弱性,在 AVA“脆弱性评估”类中作为脆弱性评估的一个方面单独处理。

“测试”类将测试分为开发者测试和评估者测试。覆盖(ATE_COV)和深度(ATE_DPT)族涉及整个开发者测试,前者说明功能规范已被严格测试,后者说明针对其他设计描述(安全结构,TOE 设计,实现说明)的测试是否被满足。

功能测试族(ATE_FUN)描述了开发者该如何执行测试并该怎样记录这些测试。独立测试族(ATE_IND)描述了评估者测试,即评估者应该重复开发者的所有测试还是部分测试,以及该执行多少测试。

图 14 给出了本类中包含的族以及族内组件的层次结构。

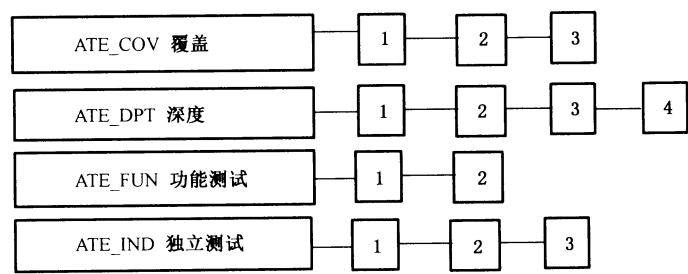


图 14 测试类分解

14.1 覆盖(ATE_COV)

14.1.1 目的

本族的目的证实已经按照功能规范对 TSF 进行了测试。这是通过检查开发者的对应性证据来实现的。

14.1.2 组件分级

本族中的组件是基于规范说明来分级的。

14.1.3 应用注释

14.1.4 ATE_COV.1 覆盖证据

依赖关系:ADV_FSP.2 安全执行功能规范;
ATE_FUN.1 功能测试。

14.1.4.1 目的

该组件的目的是确定已经对 TSF 的一些接口进行了测试。

14.1.4.2 应用注释

在该组件中开发者表明了测试文档中的测试如何与功能规范中的 TSF 接口对应。这可通过一个对应性陈述来实现,可以使用一个表格来说明。

14.1.4.3 开发者行为元素

14.1.4.3.1 ATE_COV.1.1D

开发者应提供测试覆盖的证据。

14.1.4.4 内容和形式元素

14.1.4.4.1 ATE_COV.1.1C

测试覆盖的证据应表明测试文档中的测试与功能规范中的 TSF 接口之间的对应性。

14.1.4.5 评估者行为元素

14.1.4.5.1 ATE_COV.1.1E

评估者应确认所提供的信息满足证据的所有内容和形式要求。

14.1.5 ATE_COV.2 覆盖分析

依赖关系:ADV_FSP.2 安全执行功能规范;
ATE_FUN.1 功能测试。

14.1.5.1 目的

本组件的目的是确认已经对所有的 TSF 接口都进行了测试。

14.1.5.2 应用注释

在该组件中开发者确认测试文档中的测试如何对应功能规范中的所有 TSF 接口。这可通过一个

对应性陈述来实现,可以使用一个表格来说明,同时开发者也需提供对测试覆盖的分析。

14.1.5.3 开发者行为元素

14.1.5.3.1 ATE_COV.2.1D

开发者应提供对测试覆盖的分析。

14.1.5.4 内容和形式元素

14.1.5.4.1 ATE_COV.2.1C

测试覆盖分析应证实测试文档中的测试与功能规范中 TSF 接口之间的对应性。

14.1.5.4.2 ATE_COV.2.2C

测试覆盖分析应证实已经对功能规范中的所有 TSF 接口都进行了测试。

14.1.5.5 评估者行为元素

14.1.5.5.1 ATE_COV.2.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.1.6 ATE_COV.3 严格覆盖分析

依赖关系:ADV_FSP.2 安全执行功能规范;
ATE_FUN.1 功能测试。

14.1.6.1 目的

本组件的目的是确认开发者已经对功能规范中的所有接口进行了彻底的测试。同时,本组件还要确认对所有 TSF 接口的所有参数已经进行了测试。

14.1.6.2 应用注释

在该组件中开发者必须表明测试文档中的测试如何对应功能规范中所有的 TSF 接口。这可通过一个对应性陈述来实现,可以使用一个表格来说明;另外,开发者必须证实针对所有 TSF 接口的所有参数的测试活动,包括边界测试(例如核查超过特定范围时产生的错误)和一致性测试(例如当把访问权赋给用户 A,不但核实 A 现在获得了访问权,而且要核实用户 B 不能突然地获得了访问权)。严格地讲,这种测试不是彻底的,因为不能对参数每一种可能的值都进行检查。

14.1.6.3 开发者行为元素

14.1.6.3.1 ATE_COV.3.1D

开发者应提供对测试覆盖的分析。

14.1.6.4 内容和形式元素

14.1.6.4.1 ATE_COV.3.1C

测试覆盖分析应证实测试文档中的测试与功能规范中 TSF 接口之间的对应性。

14.1.6.4.2 ATE_COV.3.2C

测试覆盖分析应证实已经对功能规范中的所有 TSF 接口进行了完全地测试。

14.1.6.5 评估者行为元素

14.1.6.5.1 ATE_COV.3.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.2 深度(ATE_DPT)

14.2.1 目的

本族中的组件涉及开发者做的 TSF 测试所能达到的详细程度。TSF 测试是根据从额外设计说明和描述中(TOE 设计,实现说明和安全结构描述等)导出的信息的深度不断增加而增加的。

本族的目的是降低在 TOE 开发中存在某些错误的风险。执行特定的内部接口测试不仅能保障 TSF 展现了所期望的外部安全行为,而且能保障这种行为来源于正确工作的内部功能。

14.2.2 组件分级

本族中的组件分级是基于 TSF 陈述所提供的细节不断增加,即从 TOE 设计到实现表示百分级的。这个分级反映了 ADV 类中所提出的 TSF 表示。

14.2.3 应用注释

TOE 设计描述了内部组件(例如子系统)、TSF 模块以及这些组件和模块之间的接口。TOE 设计的测试证据必须表明所述行为的内部接口已经被使用和执行。这可通过对 TSF 的外部接口测试,或者通过单独对 TOE 子系统、模块接口测试来实现,也可能是通过借用某个测试套件实现。如果内部接口的某些方面不能通过外部接口来测试,那么要么论证这些方面不需要被测试,要么应该直接对内部接口进行测试。在后者的情况下,TOE 设计需要充分详细以便于进行直接测试。

如果 TSF 的架构可靠性(在安全架构中(ADV_ARC))描述中引用了特定的机制,开发者所做的测试必须表明这个机制能和它描述的那样有效运行。

在该族的最高级别组件中,不仅要针对 TOE 设计进行测试,而且要对实现表示进行测试。

14.2.4 ATE_DPT.1 测试:基本设计

依赖关系:ADV_ARC.1 安全架构描述;
ADV_TDS.2 结构化设计;
ATE_FUN.1 功能测试。

14.2.4.1 目的

TSF 子系统描述提供了 TSF 内部工作的一个高层描述。在 TOE 子系统级别上进行测试,能保障 TSF 子系统的行为和交互与 TOE 设计、安全结构描述是一致的。

14.2.4.2 开发者行为元素

14.2.4.2.1 ATE_DPT.1.1D

开发者应提供测试深度分析。

14.2.4.3 内容和形式元素

14.2.4.3.1 ATE_DPT.1.1C

测试深度分析应证实测试文档中的测试与 TOE 设计中 TSF 子系统之间的对应性。

14.2.4.3.2 ATE_DPT.1.2C

测试深度分析应证实 TOE 设计中的所有 TSF 子系统都已经进行过测试。

14.2.4.4 评估者行为元素

14.2.4.4.1 ATE_DPT.1.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.2.5 ATE_DPT.2 测试:安全执行模块

依赖关系:ADV_ARC.1 安全架构描述;
ADV_TDS.3 基础模块设计;
ATE_FUN.1 功能测试。

14.2.5.1 目的

TSF 的子系统和模块描述提供了 TSF 内部工作的一个高层描述,同时提供了 SFR-执行模块的接口描述。在本 TOE 描述级别上进行测试,能保障 TSF 子系统和 SFR-执行模块的行为和交互,与 TOE 设计和安全架构描述中的描述是一致的。

14.2.5.2 开发者行为元素

14.2.5.2.1 ATE_DPT.2.1D

开发者应提供测试深度分析。

14.2.5.3 内容和形式元素

14.2.5.3.1 ATE_DPT.2.1C

深度测试分析应证实测试文档中的测试与 TOE 设计中的 TSF 子系统、SFR-执行模块之间的一致性。

14.2.5.3.2 ATE_DPT.2.2C

测试深度分析应证实 TOE 设计中的所有 TSF 子系统都已经进行过测试。

14.2.5.3.3 ATE_DPT.2.3C

测试深度分析应证实 TOE 设计中的 SFR-执行模块都已经进行过测试。

14.2.5.4 评估者行为元素

14.2.5.4.1 ATE_DPT.2.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.2.6 ATE_DPT.3 测试:模块设计

依赖关系:ADV_ARC.1 安全架构描述;
ADV_TDS.4 半形式化模块设计;
ADV_FUN.1 功能测试。

14.2.6.1 目的

TSF 的子系统和模块提供了 TSF 内部工作的一个高层描述,同时提供了模块接口的描述。在本 TOE 描述级别上进行测试,能保障 TSF 子系统和模块的行为和交互,与 TOE 设计和安全架构描述中是一致的。

14.2.6.2 开发者行为元素

14.2.6.2.1 ATE_DPT.3.1D

开发者应提供测试深度分析。

14.2.6.3 内容和形式元素

14.2.6.3.1 ATE_DPT.3.1C

深度测试分析应证实测试文档中的测试与 TOE 设计中的 TSF 子系统、模块之间的一致性。

14.2.6.3.2 ATE_DPT.3.2C

测试深度分析应证实 TOE 设计中的所有 TSF 子系统都已经进行过测试。

14.2.6.3.3 ATE_DPT.3.3C

测试深度分析应证实 TOE 设计中的**所有 TSF** 模块都已经进行过测试。

14.2.6.4 评估者行为元素

14.2.6.4.1 ATE_DPT.3.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.2.7 ATE_DPT.4 测试:实现表示

依赖关系:ADV_ARC.1 安全架构描述;
ADV_TDS.4 半形式化模块设计;
ADV_IMP.1 TSF 实现表示;
ADV_FUN.1 功能测试。

14.2.7.1 目的

TSF 的子系统和模块描述提供了 TSF 内部工作的一个高层描述,同时提供了模块接口的描述。在本 TOE 描述级别上进行测试,能保障 TSF 子系统和模块的行为和交互,与 TOE 设计、安全结构描述中的描述和实现表示是一致的。

14.2.7.2 开发者行为元素

14.2.7.2.1 ATE_DPT.4.1D

开发者应提供测试深度分析。

14.2.7.3 内容和形式元素

14.2.7.3.1 ATE_DPT.4.1C

深度测试分析应证实测试文档中的测试与 TOE 设计中的 TSF 子系统、模块之间的一致性。

14.2.7.3.2 ATE_DPT.4.2C

测试深度分析应证实 TOE 设计中的所有 TSF 子系统都已经进行过测试。

14.2.7.3.3 ATE_DPT.4.3C

测试深度分析应证实 TOE 设计中的所有模块都已经进行过测试。

14.2.7.3.4 ATE_DPT.4.4C

测试深度分析应证实 TSF 运行与其实现表示是完全一致的。

14.2.7.4 评估者行为元素

14.2.7.4.1 ATE_DPT.4.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.3 功能测试(ATE_FUN)

14.3.1 目的

开发者执行功能测试,用以保障测试文档中的测试被正确执行和记录。这些测试与 TSF 设计描述的对应性通过“覆盖”族(ATE_COV)和“深度”族(ATE_DPT)来实现。

本族有助于保障未知缺陷出现的可能性相对较小。

ATE_COV“覆盖”族、ATE_DPT“深度”族和 ATE_FUN“功能测试”族常组合起来定义由开发者提供的测试证据。评估者进行的独立功能测试由 ATE_IND“独立测试”族负责规范。

14.3.2 组件分级

本族包括两个组件,较高层组件要求分析顺序依赖性。

14.3.3 应用注释

执行测试的过程最好能提供测试程序和测试套的使用说明书,包括测试环境、测试条件、测试数据参数和值。测试过程还应该说明测试结果是如何从测试输入中导出的。

当某个特定测试的成功执行依赖于某个特定状态的存在时,顺序依赖性就十分重要了。例如:可能要求测试 B 必须紧接在测试 A 之后执行,因为 A 的成功执行是 B 成功执行的必要条件。这样,测试 B 的失败可能与顺序依赖性问题有关。在上面的例子中,测试 B 的失败可能是由于测试 C(而不是测试 A)紧接在它之前被执行而不是测试 A,也可能是与测试 A 的失败有关。

14.3.4 ATE_FUN.1 功能测试

依赖关系:ATE_COV.1 覆盖证据。

14.3.4.1 目的

本组件的目的是由开发者证实测试文档中的测试都被正确执行和记录。

14.3.4.2 开发者行为元素

14.3.4.2.1 ATE_FUN.1.1D

开发者应测试 TSF,并文档化测试结果。

14.3.4.2.2 ATE_FUN.1.2D

开发者应提供测试文档。

14.3.4.3 内容和形式元素

14.3.4.3.1 ATE_FUN.1.1C

测试文档应包括测试计划、预期的测试结果和实际的测试结果。

14.3.4.3.2 ATE_FUN.1.2C

测试计划应标识要执行的测试并描述执行每个测试的方案,这些方案应包括对于其他测试结果的任何顺序依赖性。

14.3.4.3.3 ATE_FUN.1.3C

预期的测试结果应指出测试成功执行后的预期输出。

14.3.4.3.4 ATE_FUN.1.4C

实际的测试结果应和预期的测试结果一致。

14.3.4.4 评估者行为元素

14.3.4.4.1 ATE_FUN.1.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.3.5 ATE_FUN.2 顺序的功能测试

依赖关系:ATE_COV.1 覆盖证据。

14.3.5.1 目的

本组件的目的是由开发者证实测试文档中的测试都被正确的执行和记录,同时保证测试是结构化的,例如避免反复论证正在测试的接口的正确性。

14.3.5.2 应用注释

尽管测试过程可以根据测试的顺序规定发起测试的前提条件,但可以不对排序进行基本原理证明。对于测试顺序的分析是确定测试充分性的一个重要因素,因为存在缺陷被测试的顺序掩盖的可能性。

14.3.5.3 开发者行为元素

14.3.5.3.1 ATE_FUN.2.1D

开发者应测试 TSF,并文档化测试结果。

14.3.5.3.2 ATE_FUN.2.2D

开发者应提供测试文档。

14.3.5.4 内容和形式元素

14.3.5.4.1 ATE_FUN.2.1C

测试文档应包括测试计划、预期的测试结果和实际的测试结果。

14.3.5.4.2 ATE_FUN.2.2C

测试计划应标识要执行的测试和描述执行每个测试的方案,这些方案应包括对于其他测试结果的任何顺序依赖性。

14.3.5.4.3 ATE_FUN.2.3C

预期的测试结果应指出测试成功执行后的预期输出。

14.3.5.4.4 ATE_FUN.2.4C

实际的测试结果应和预期的测试结果一致。

14.3.5.4.5 ATE_FUN.2.5C

测试文档应包含测试步骤顺序依赖性的一个分析。

14.3.5.5 评估者行为元素

14.3.5.5.1 ATE_FUN.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

14.4 独立测试(ATE_IND)

14.4.1 目的

本族的目的是以“功能测试”族(ATE_FUN)、“覆盖”族(ATE_COV)和“深度”族(ATE_DPT)取得的保障为基础,评估者核查开发者的测试,并执行一些额外的测试。

14.4.2 组件分级

本族的分级是基于开发者测试文档和测试支持的数量以及评估者测试的数量。

14.4.3 应用注释

本族涉及 TSF 独立测试所能达到的程度。独立测试可以采用重复开发者的功能测试(整个或部分)或拓宽开发者测试的范围或深度的形式进行。这些行为是互补的,并且对于每个 TOE 必须制定一个适当的组合计划,这种组合应当考虑测试结果的可用性和覆盖度,以及 TSF 的功能复杂度。

对开发者测试的抽样是为了确认开发者已经开展了他所计划的 TSF 测试程序,并正确记录了结果。抽样数量受开发者功能测试结果的详细程度和质量的影响。评估者也应考虑设计额外测试的范围,以及在这两方面努力所带来的帮助。在某些情况下,重复所有的开发者测试也许是可行且合理的,但是比较费力而且效率低下。所以本族最高级别的组件应谨慎使用。抽样将涉及测试结果的全部有效范围,包括那些满足 ATE_COV“覆盖”和 ATE_DPT“深度”要求的结果。

在评估过程中,TOE 的不同配置也是需要考虑的,评估者必须评估所提供结果的适应性,并相应地制定自己的测试计划。

TOE 测试的适宜性基于对 TOE 的访问,以及运行测试所需要的支持性文档和信息(包括任何测试软件和工具)。这种支持需求由对其他保障族的依赖关系来负责处理。

另外,对 TOE 测试的适宜性也可能基于其他考虑,例如开发者提交的 TOE 不是最终版本。

术语“接口”是指在功能规范和 TOE 设计中描述的接口,其中传递的调用参数在实现表示中标识。需要用到接口的精确集合是由“覆盖”族(ATE_COV)和“深度”族(ATE_DPT)的组件确定的。

对这些接口子集的引用旨在允许评估者设计一组适当的测试集,这与所实施评估的目的是一致的。

14.4.4 ATE_IND.1 独立测试—符合性

依赖关系:ADV_FSP.1 基础功能规范;
AGD_OPE.1 操作用户指南;
AGD_PRE.1 准备程序。

14.4.4.1 目的

本组件的目的是证实 TOE 的运行与它的设计描述和指导性文档是一致的。

14.4.4.2 应用注释

本组件不处理开发者测试结果的使用。当这些测试结果不可用时,或开发者的测试未经验证就接受时,这个组件是适用的。要求评估者设计并实施测试,以确认 TOE 的运行与设计实现,包括但不限于功能规范,是一致的。这样的方法是为了能通过有代表性的测试来取得正确运行的信心。而不是为了执行每一个可能的测试。为了此目的而计划的测试范围是一个方法论的问题,需要考虑一个特定 TOE 的环境以及其他测评活动的平衡。

14.4.4.3 开发者行为元素

14.4.4.3.1 ATE_IND.1.1D

开发者应提供用于测试的 TOE。

14.4.4.4 内容和形式元素

14.4.4.4.1 ATE_IND.1.1C

TOE 应适合测试。

14.4.4.5 评估者行为元素

14.4.4.5.1 ATE_IND.1.1E

评估者应确认所提供的信息满足证据内容和形式的所有要求。

14.4.4.5.2 ATE_IND.1.2E

评估者应测试 TSF 的一个子集,以确认 TSF 按照规定运行。

14.4.5 ATE_IND.2 独立测试—抽样

依赖关系:ADV_FSP.2 安全执行功能规范;
AGD_OPE.1 操作用户指南;
AGD_PRE.1 准备程序;
ATE_COV.1 覆盖证据;
ATE_FUN.1 功能测试。

14.4.5.1 目的

本组件的目的是证实 TOE 的运行与其设计实现和指导性文档是一致的。评估者的测试确认开发

者对功能规范中的接口进行了测试。

14.4.5.2 应用注释

目的是开发者应当为评估者提供必须的材料用于有效的重现开发者的测试。材料可包括机器可读的测试文档、测试程序等。

本组件包含这样一个要求：即评估者应拥有开发者提供的可用测试结果，以补充测试程序。评估者将重复开发者测试的一个样本，以获得对所得测试结果的信任。在建立上述的信任后，评估者将在开发者测试的基础上，通过执行额外的测试以另一种方式运行 TOE。在给定一个确定的资源水平，通过使用一个验证开发者测试结果的平台，相比仅使用开发者自己的测试而言，评估者能够确信 TOE 可在更大的范围内获得正确运行。在相信开发者已测试 TOE 后，评估者也将拥有更多的自由，适当时可以在文档检查或专业知识判断引起特别关心的某些领域集中测试。

14.4.5.3 开发者行为元素

14.4.5.3.1 ATE_IND.2.1D

开发者应提供用于测试的 TOE。

14.4.5.4 内容和形式元素

14.4.5.4.1 ATE_IND.2.1C

TOE 应适合测试。

14.4.5.4.2 ATE_IND.2.2C

开发者应提供一组与开发者 TSF 功能测试中同等的一系列资源。

14.4.5.5 评估者行为元素

14.4.5.5.1 ATE_IND.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

14.4.5.5.2 ATE_IND.2.2E

评估者应执行测试文档中的测试样本，以验证开发者的测试结果。

14.4.5.5.3 ATE_IND.2.3E

评估者应测试 TSF 的一个子集以确认 TSF 按照规定运行。

14.4.6 ATE_IND.3 独立测试—完全

- 依赖关系：ADV_FSP.4 完备的功能规范；
- AGD_OPE.1 操作用户指南；
- AGD_PRE.1 准备程序；
- ATE_COV.1 覆盖证据；
- ATE_FUN.1 功能测试。

14.4.6.1 目的

本组件的目的是证实 TOE 的运行与其设计实现和指导性文档是一致的。评估者测试包括重复所

有的开发者测试。

14.4.6.2 应用注释

目的是开发者应当为评估者提供必须的材料用于有效的重现开发者的测试，材料可包括机器可读的测试文档、测试程序等。

在本组件中，评估者必须重复所有的开发者测试，以作为测试程序的一个部分。如前面的组件一样，评估者也将会以不同于开发者的方式运行 TSF 以执行测试。在开发者测试非常详尽的情况下，评估者仅有极小的余地做到这样。

14.4.6.3 开发者行为元素

14.4.6.3.1 ATE_IND.3.1D

开发者应提供用于测试的 TOE。

14.4.6.4 内容和形式元素

14.4.6.4.1 ATE_IND.3.1C

TOE 应适合测试。

14.4.6.4.2 ATE_IND.3.2C

开发者应提供与开发者 TSF 功能测试中同等的一系列资源。

14.4.6.5 评估者行为元素

14.4.6.5.1 ATE_IND.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

14.4.6.5.2 ATE_IND.3.2E

评估者应执行测试文档中的**所有**测试，以验证开发者的测试结果。

14.4.6.5.3 ATE_IND.3.3E

评估者应测试 TSF 确认**全部** TSF 按照规定运行。

15 AVA 类：脆弱性评定

AVA：脆弱性评定类负责处理在 TOE 开发或者运行过程中引入可被利用脆弱性的可能性。

图 15 给出了本类中包含的族以及族内组件的层次结构：

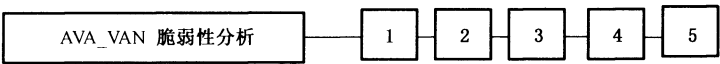


图 15 脆弱性评定类分解

15.1 应用注释

通常，脆弱性评定活动涉及 TOE 开发和运行中的各种脆弱性。开发脆弱性利用的是在开发过程中引入的 TOE 的一些属性，例如通过篡改、直接攻击或跟踪 TSF 等攻破 TSF 自我保护，或者通过跟

踪、直接攻击 TSF 等攻破 TSF 域分隔,或者通过绕过(旁路)TSF 攻破不可旁路性。运行脆弱性利用非技术措施中的脆弱点去破坏 TOE 安全功能要求,例如误用或错误配置。“误用”考察的是在 TOE 的管理员或用户被认为是安全时,TOE 是否会以一种不安全的方式被配置或使用。

开发脆弱性评定是由“脆弱性分析”保障族 AVA_VAN 来处理的。基本上所有的开发脆弱性都能在 AVA_VAN“脆弱性分析”族中得到考虑,这是基于如下事实:这个族允许对一类攻击场景可以使用各种非特定的评定方法学。这些非特定的评定方法学再结合针对 TSF 特定的方法学,能使隐蔽信道被考虑到(使用非正式的工程学测量方法或实际的测试方法能对一个信道的带宽进行估算)或者使用足够的资源以一种直接的攻击来克服(TSF 技术概念中隐含的基于概率或置换机制;能通过定量或统计分析确定功能行为的限制或者克服它们必须的措施)。

如果在 ST 中规定这样的安全目标:TOE 的一个用户不能观察另一个用户的活动,或者保证信息流不能被利用而得到非法的数据信号,那么在执行脆弱性分析中必须考虑隐蔽信道分析。这些通常反映在 ST 中包含的 ISO/IEC 15408-2 不可观察性(FPR_UNO)、ISO/IEC 15408-2 访问控制策略(FDP_ACC)中的多级访问控制策略的或者 ISO/IEC 15408-2 数据流控制策略(FDP_IFC)要求中。

15.2 脆弱性分析(AVA_VAN)

15.2.1 目的

脆弱性分析是一种评估方法,用来确定在评估 TOE 开发和预期运行期间潜在的脆弱性是否已被标识,或通过其他方法(例如缺陷假设或对基础安全机制安全行为的定量或统计分析)是否允许攻击者破坏安全功能要求。

脆弱性分析涉及攻击者能发现一些缺陷所造成的威胁,这些缺陷将允许对数据或者功能进行非授权的访问、允许能够干扰或改变 TSF、或者妨碍其他用户已授权能力。

15.2.2 组件分级

本族中的组件是基于评估者进行脆弱性分析时不断增加的严格性以及攻击者标识和利用潜在脆弱性所需攻击潜力要求的不断增加而分级的。

15.2.3 AVA_VAN.1 脆弱性调查

依赖关系:ADV_FSP.1 基础功能规范;
AGD_OPE.1 操作用户指南;
AGD_PRE.1 准备程序。

15.2.3.1 目的

评估者执行公共可用信息的一个脆弱性调查,以确定那些可能很容易被攻击者发现的潜在脆弱性。

评估者执行穿透性测试,以确认潜在的脆弱性在 TOE 运行环境中不能被利用。评估者在假定具有基本的攻击潜力的情况下执行穿透性测试。

15.2.3.2 开发者行为元素

15.2.3.2.1 AVA_VAN.1.1D

开发者应提供用于测试的 TOE。

15.2.3.3 内容和形式元素

15.2.3.3.1 AVA_VAN.1.1C

TOE 应适合测试。

15.2.3.4 评估者行为元素

15.2.3.4.1 AVA_VAN.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

15.2.3.4.2 AVA_VAN.1.2E

评估者应执行公共领域的调查以标识 TOE 的潜在脆弱性。

15.2.3.4.3 AVA_VAN.1.3E

评估者应基于已标识的潜在脆弱性实施穿透性测试,确定 TOE 能抵抗具有基本攻击潜力的攻击者的攻击。

15.2.4 AVA_VAN.2 脆弱性分析

依赖关系:ADV_ARC.1 安全架构描述;
ADV_FSP.2 安全执行功能规范;
ADV_TDS.1 基础设计;
AGD_OPE.1 操作用户指南;
AGD_PRE.1 准备程序。

15.2.4.1 目的

评估者执行脆弱性分析以确认潜在脆弱性的存在。

评估者执行穿透性测试,以确认潜在的脆弱性在 TOE 运行环境中不能被利用。评估者在假定具有基本的攻击潜力的情况下执行穿透性测试。

15.2.4.2 开发者行为元素

15.2.4.2.1 AVA_VAN.2.1D

开发者应提供用于测试的 TOE。

15.2.4.3 内容和形式元素

15.2.4.3.1 AVA_VAN.2.1C

TOE 应适合测试。

15.2.4.4 评估者行为元素

15.2.4.4.1 AVA_VAN.2.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

15.2.4.4.2 AVA_VAN.2.2E

评估者应执行公共领域的调查以标识 TOE 的潜在脆弱性。

15.2.4.4.3 AVA_VAN.2.3E

评估者应执行独立的 TOE 脆弱性分析去标识 TOE 潜在的脆弱性,在分析过程中使用指导性文档、

功能规范、TOE 设计和安全结构描述。

15.2.4.4.4 AVA_VAN.2.4E

评估者应基于已标识的潜在脆弱性实施穿透性测试,确定 TOE 能抵抗具有基本攻击潜力的攻击者的攻击。

15.2.5 AVA_VAN.3 关注点脆弱性分析

依赖关系:ADV_ARC.1 安全架构描述;
ADV_FSP.4 完备的功能规范;
ADV_TDS.3 基础模块设计;
ADV_IMP.1 TSF 实现表示;
AGD_OPE.1 操作用户指南;
AGD_PRE.1 准备程序;
ATE_DPT.1 测试:基本设计。

15.2.5.1 目的

评估者执行脆弱性分析以确认潜在脆弱性的存在。

评估者执行穿透性测试,以确认潜在的脆弱性在 TOE 运行环境中不能被利用。评估者在假定具有增强型基本攻击潜力的情况下执行穿透性测试。

15.2.5.2 开发者行为元素

15.2.5.2.1 AVA_VAN.3.1D

开发者应提供用于测试的 TOE。

15.2.5.3 内容和形式元素

15.2.5.3.1 AVA_VAN.3.3C

TOE 应适合测试。

15.2.5.4 评估者行为元素

15.2.5.4.1 AVA_VAN.3.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

15.2.5.4.2 AVA_VAN.3.2E

评估者应执行一个公共领域的调查以标识 TOE 的潜在脆弱性。

15.2.5.4.3 AVA_VAN.3.3E

评估者应针对 TOE 执行一个独立的脆弱性分析去标识 TOE 中潜在的脆弱性,在分析过程中使用指导性文档、功能规范、TOE 设计、安全结构描述和实现表示。

15.2.5.4.4 AVA_VAN.3.4E

评估者应基于已标识的潜在脆弱性实施穿透性测试,确定 TOE 能抵抗具有增强型基本攻击潜力

的攻击者的攻击。

15.2.6 AVA_VAN.4 系统的脆弱性分析

依赖关系:ADV_ARC.1 安全架构描述;
ADV_FSP.4 完备的功能规范;
ADV_TDS.3 基础模块设计;
ADV_IMP.1 TSF 实现表示;
AGD_OPE.1 操作用户指南;
AGD_PRE.1 准备程序;
ATE_DPT.1 测试:基本设计。

15.2.6.1 目的

评估者执行系统的脆弱性分析以确认潜在脆弱性的存在。

评估者执行穿透性测试,以确认潜在的脆弱性在 TOE 运行环境中不能被利用。评估者在假定具有中等攻击潜力的情况下执行穿透性测试。

15.2.6.2 开发者行为元素

15.2.6.2.1 AVA_VAN.4.1D

开发者应提供用于测试的 TOE。

15.2.6.3 内容和形式元素

15.2.6.3.1 AVA_VAN.4.1C

TOE 应适合测试。

15.2.6.4 评估者行为元素

15.2.6.4.1 AVA_VAN.4.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

15.2.6.4.2 AVA_VAN.4.2E

评估者应执行公共领域的调查以标识 TOE 的潜在脆弱性。

15.2.6.4.3 AVA_VAN.4.3E

评估者应针对 TOE 执行独立的、系统的脆弱性分析去标识 TOE 潜在的脆弱性,在分析过程中使用指导性文档、功能规范、TOE 设计、安全结构描述和实现表示。

15.2.6.4.4 AVA_VAN.4.4E

评估者应基于已标识的潜在脆弱性实施穿透性测试,确定 TOE 能抵抗具有中等攻击潜力的攻击者的攻击。

15.2.7 AVA_VAN.5 高级的系统的脆弱性分析

依赖关系:ADV_ARC.1 安全架构描述;

ADV_FSP.4 完备的功能规范；
ADV_TDS.3 基础模块设计；
ADV_IMP.1 TSF 实现表示；
AGD_OPE.1 操作用户指南；
AGD_PRE.1 准备程序；
ATE_DPT.1 测试：基本设计。

15.2.7.1 目的

评估者执行系统的脆弱性分析以确认潜在脆弱性的存在。

评估者执行穿透性测试，以确认潜在的脆弱性在 TOE 运行环境中不能被利用。评估者在假定具有高等攻击潜力的攻击情况下执行渗透性测试。

15.2.7.2 开发者行为元素

15.2.7.2.1 AVA_VAN.5.1D

开发者应提供用于测试的 TOE。

15.2.7.3 内容和形式元素

15.2.7.3.1 AVA_VAN.5.1C

TOE 应适合测试。

15.2.7.4 评估者行为元素

15.2.7.4.1 AVA_VAN.5.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

15.2.7.4.2 AVA_VAN.5.2E

评估者应执行公共领域的调查以标识 TOE 的潜在脆弱性。

15.2.7.4.3 AVA_VAN.5.3E

评估者应针对 TOE 执行独立的、**系统的**脆弱性分析去标识 TOE 潜在的脆弱性，在分析过程中使用指导性文档、功能规范、TOE 设计、安全结构描述和实现表示。

15.2.7.4.4 AVA_VAN.5.4E

评估者**应**基于已标识的潜在脆弱性**实施**穿透性测试，确定 TOE 能抵抗具有**高等**攻击潜力的攻击者的攻击。

16 ACO 类：组合

ACO“组合”类包括五个族。本族列出的保障要求，是为了提供确信一个组合 TOE 依靠过去评估过的软件、固件或硬件等组成部分所提供的安全功能是能够安全运行的信心。

“组合”是针对成功通过 ISO/IEC 15408 安全保障要求包（基本部件和依赖部件，参见附录 B）的评估的两个或更多的 IT 实体结合起来的**一个整体**，不包括对其中的 IT 实体进行的下一步的开发。额外

IT 实体(以前未经组件评估的实体)的开发也不包括。组合 TOE 形成了一种新的产品,它被安装并集成进任何特定环境中以满足环境目的。

“组合”没有为组件的评估提供可选的方法。ACO 类的“组合”为组合 TOE 整合者提供了一种方法,作为可选的方法被用于 ISO/IEC 15408 中指定的其他保障级,在由两个或更多的成功评估过的组件合成的 TOE 评估过程中获得信心,而不需要重新评估合成的 TSF。(在整个 ACO 类中用组合 TOE 整合者指代“开发者”,任何与基本或依赖部件相关的开发者都这样阐述。)

第 8 章和 6.3 定义的组合保障包是衡量组合 TOE 的一种保障等级。这种保障等级是除了 EAL 之外的要求,因为对于依据 EAL 评估过的组件,得到一个 EAL 保障的结果,组合 TOE 必须满足 EAL 中所有的安全保障要求项。尽管重复使用能构成组合 TOE 的评估结果,但是经常会有组件的其他方面必须放到组合 TOE 中考虑,如 B.3 中描述的那些。由于一个组合 TOE 的评估活动涉及不同的部分,通常不可能获得关于组件额外方面的所有必需的证据用于申请合适的 EAL。因此,定义组合保障包来解决评估组件的组合以及获得一个有意义结果相关的问题。这些在附录 B 中有进一步的探讨。

图 16 给出了 ACO 各族之间的关系和部件间的关系。在组合 TOE 中,通常存在一个部件依赖另外一个部件提供服务的情形。需要服务的部件称为依赖部件,提供相应服务的部件称为基本部件。这种相互作用和区别在附录 B 中有进一步的讨论。假设依赖部件的开发者是正在以某种形式(作为开发者、发起者或来自依赖部件评估组织的、提供必要评估证据的协助人员)支持组合 TOE 的评估。CAP 保障包包括的 ACO 组件不应被用于组合 TOE 评估的增加物,因为这无法为组件提供任何有意义的保障。

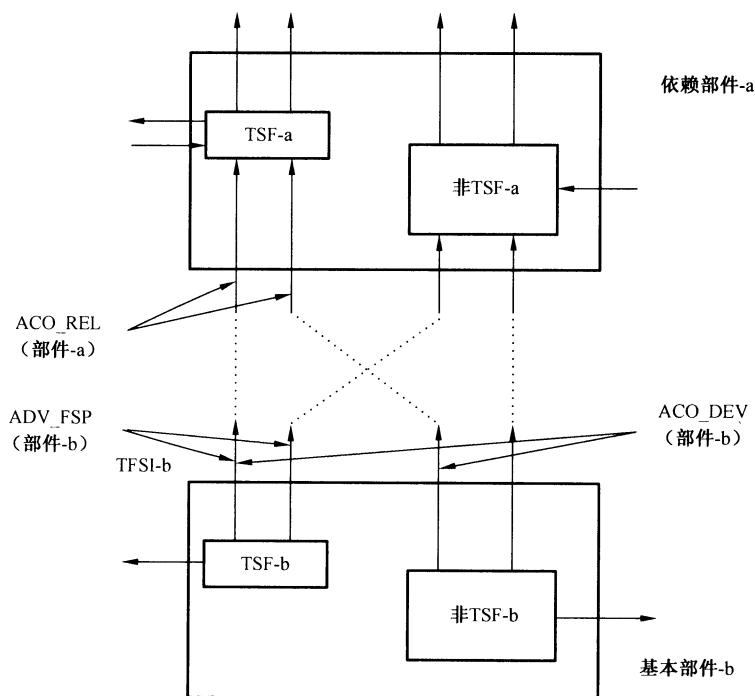


图 16 ACO 各族之间的关系和部件间的关系

在组合 TOE 评估过程中, ACO 类的族与 ADV、ATE、AVA 类是相似的交互方式, 因此, 那些类的要求规范在这里也是适用的。但有一些专门适用于组合 TOE 的特殊项。为了确定这些组件怎样作用, 并且确定不同部件评估的差别, ACO_REL 族明确标示出依赖部件对基本部件的依赖关系。按照依赖部件调用基本部件安全功能要求提供的服务接口, 详细说明对基本部件的这种依赖。基本部件提供的接口, 作为更上层的支持行为, 是基本部件对服务请求的响应, 这将在 ACO_DEV 族中分析。ACO_DEV 族是基于 ADV_TDS 族的, 从最简单的角度理解, 每个部件的 TSF 可看作组合 TOE 的一个子系统, 每个部件的额外部分可看作额外的子系统。因此, 在组合 TOE 评估中, 这些部件之间的接口被看作不同子系统之间的相互作用。

提供给 ACO_DEV 的接口和支持行为描述可能是不完备的。这将在 ACO_COR 实施期间确定的。ACO_COR 族接受 ACO_REL 族和 ACO_DEV 族的输出,确定这些组件是否正在用于已经评估过的配置项中,并标示出说明文档中不完整的地方,这些标示将作为组合 TOE 测试(ACO_CTT)和脆弱性分析(ACO_VUL)活动的输入。

执行测试是为了确定组合 TOE 能够表现出组合 TOE 的安全功能要求所确定的预期行为,并且在高层次上证实组合 TOE 的不同部件之间接口的兼容性。

组合 TOE 的脆弱性分析受各部件评估脆弱性分析结果的影响。组合 TOE 脆弱性分析要分析部件评估的所有残余脆弱性,确定残余脆弱性在组合 TOE 中不可用。从各自部件评估完成之后,也会执行一次与部件相关问题的公共可用信息的调查。

图 17 描述了 ACO 各族之间的关系。如图中所示,实线箭头表明一个族获得的认识和证据流入下一个评估活动,虚线箭头表示一个评估活动明确地追溯到组合 TOE 的安全功能要求。

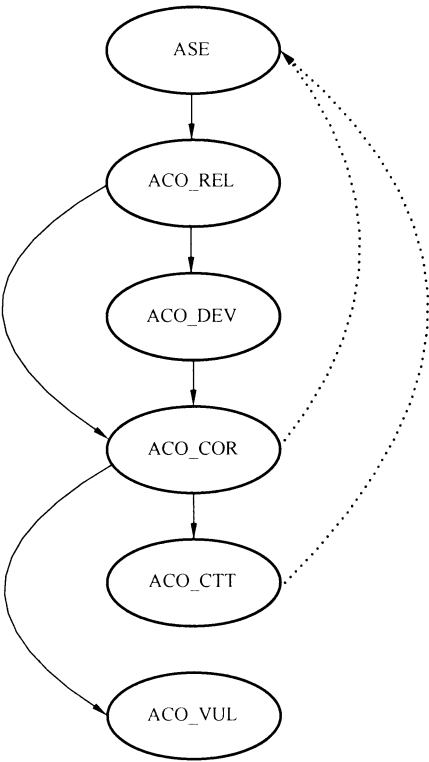


图 17 ACO 各族之间的关系

附录 B 中提供了组合 TOE 内部定义和相互作用的进一步阐述。

图 18 给出了本类中的族及族中各组件之间的层次关系。

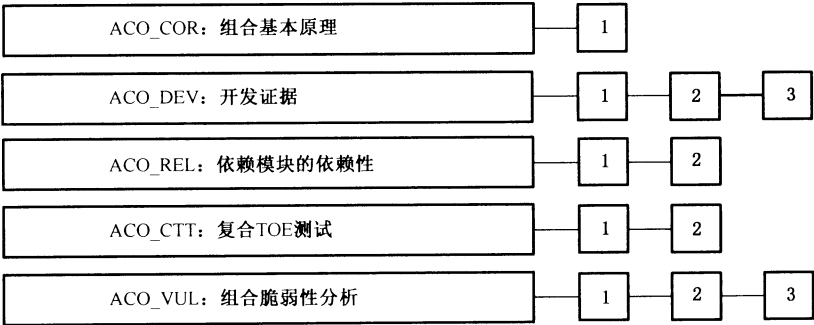


图 18 组合类分解

16.1 组合基本原理(ACO_COR)

16.1.1 目的

本族提出这样的要求:证实基本部件能为组合产品提供有效合适的保障级。

16.1.2 组件分级

本族仅包含一个组件。

16.1.3 ACO_COR.1 组合基本原理

依赖关系:ACO_DEV.1 功能描述;
ALC_CMC.1 TOE 标识;
ACO_REL.1 基本的依赖信息。

16.1.3.1 开发者行为元素

16.1.3.1.1 ACO_COR.1.1D

开发者应提供基本部件的组合基本原理。

16.1.3.2 内容和形式元素

16.1.3.2.1 ACO_COR.1.1C

组合基本原理应证实,当基本部件设定为支持依赖部件的 TSF 所必需的时候,基本部件的支持功能至少已获得依赖部件同样的保障级。

16.1.3.3 评估者行为元素

16.1.3.3.1 ACO_COR.1.1E

评估者应确认信息满足证据的内容和形式的的所有要求。

16.2 开发证据(ACO_DEV)

16.2.1 目的

本族陈述随着细节的逐级增加,对基本部件的规范说明的相关要求。这些信息是确认提供合适的安全功能支持依赖部件的要求所必需的(依赖信息中表示的)。

16.2.2 组件分级

本族组件是基于提供的接口以及实现细节的数量逐渐增加而分级的。

16.2.3 应用注释

基本部件的 TSF 经常在缺少潜在组合应用的依赖知识的情况下定义。基本部件的 TSF 被定义成包括执行基本部件安全功能要求依赖的所有部分,这也将包括实现基本部件安全功能要求必需的所有部分。

基本部件的功能规范将根据基本部件提供的允许外部实体调用 TSF 操作的接口来描述 TSFI。这包括允许通过与 TSF 操作交互来调用安全功能要求的用户接口,也包括允许外部 IT 实体调用 TSF 的

接口。

功能规范仅提供一份关于 TSF 提供了什么接口以及 TSF 功能被调用的方法的描述。因此,功能规范不需要提供外部实体和基本部件之间所有可能用到的接口的完整说明。不包括 TSF 对操作环境的要求/预期。依赖部件的 TSF 依赖于基本部件的描述在依赖部件的依赖性(ACO_REL)族中考虑,开发信息证据提供针对接口说明的一个反馈。

开发信息证据包括基本部件的一份规范说明。这可能是基本部件评估期间用于满足 ADV 要求的证据,或者是由基本部件的开发者或组合 TOE 开发者产生的另一种形式的证据。基本部件的规范说明用在开发证据期间(ACO_DEV),获得已提供合适的安全功能支持依赖部件要求的信心。这些证据要求细节的不断增加反映了组合 TOE 所要求的保障级别,期望能明显的反映出对这些部件运用保障包获得的不断增加的信心。评估者确定基本部件的这些描述和为依赖部件提供的依赖信息是否一致。

16.2.4 ACO_DEV.1 功能描述

依赖关系:ACO_REL.1 基本的依赖信息。

16.2.4.1 目的

依赖部件所依赖的基本部件接口的描述是必需的。这将被检查,以确定其是否与依赖部件依赖的接口信息描述一致。

16.2.4.2 开发者行为元素

16.2.4.2.1 ACO_DEV.1.1D

开发者应提供基本部件的开发信息。

16.2.4.3 内容和形式元素

16.2.4.3.1 ACO_DEV.1.1C

开发信息应描述组合 TOE 使用的基本部件的每个接口的目的。

16.2.4.3.2 ACO_DEV.1.2C

开发信息应说明用于组合 TOE 的基本部件和依赖部件之间的接口对应性,以支持依赖部件的 TSF。

16.2.4.4 评估者行为元素

16.2.4.4.1 ACO_DEV.1.1E

评估者应确认信息满足证据的内容和形式的所有要求。

16.2.4.4.2 ACO_DEV.1.2E

评估者应确定所提供的接口描述与为依赖部件提供的依赖信息是一致的。

16.2.5 ACO_DEV.2 基本的设计证据

依赖关系:ACO_REL.1 基本的依赖信息。

16.2.5.1 目的

依赖部件所依赖的基本部件接口的描述是必需的,通过检查该描述以确定其是否与依赖部件依赖

的接口信息描述一致。

另外,还要描述基本部件支持依赖部件 TSF 的安全行为。

16.2.5.2 开发者行为元素

16.2.5.2.1 ACO_DEV.2.1D

开发者应提供基本部件的开发信息。

16.2.5.3 内容和形式元素

16.2.5.3.1 ACO_DEV.2.1C

开发信息应描述组合 TOE 使用的基本部件的每个接口的目的及其使用方法。

16.2.5.3.2 ACO_DEV.2.2C

开发信息应提供基本部件支持依赖部件安全功能要求执行的行为的 高层描述。

16.2.5.3.3 ACO_DEV.2.3C

开发信息应说明用于组合 TOE 的基本部件和依赖部件之间的接口对应性,以支持依赖部件的 TSF。

16.2.5.4 评估者行为元素

16.2.5.4.1 ACO_DEV.2.1E

评估者应**确认**信息满足证据的内容和形式的所有要求。

16.2.5.4.2 ACO_DEV.2.2E

评估者应**确定**所提供的接口描述与为依赖部件提供的依赖信息是一致的。

16.2.6 ACO_DEV.3 详细的设计证据

依赖关系:ACO_REL.2 依赖信息。

16.2.6.1 目的

依赖部件所依赖的基本部件接口的描述是必需的。这将被检查,以确定其是否与依赖部件依赖的接口信息描述一致。

提供基本部件的结构的接口描述,以使评估者能确定是否是该接口构成了基本部件的 TSF 的一部分。

16.2.6.2 开发者行为元素

16.2.6.2.1 ACO_DEV.3.1D

开发者应提供基本部件的开发信息。

16.2.6.3 内容和形式元素

16.2.6.3.1 ACO_DEV.3.1C

开发信息应描述组合 TOE 使用的基本部件的每个接口的目的及其使用方法。

16.2.6.3.2 ACO_DEV.3.2C

开发信息应标识用于组合 TOE 的提供接口的基本部件子系统。

16.2.6.3.3 ACO_DEV.3.3C

开发信息应提供基本部件子系统支持依赖部件安全功能要求执行的行为的高层描述。

16.2.6.3.4 ACO_DEV.3.4C

开发信息应提供接口到基本部件子系统之间的映射。

16.2.6.3.5 ACO_DEV.3.5C

开发信息应说明用于组合 TOE 的基本部件和依赖部件之间的接口对应性,以支持依赖部件的 TSF。

16.2.6.4 评估者行为元素

16.2.6.4.1 ACO_DEV.3.1E

评估者应确认信息满足证据的内容和形式的所有要求。

16.2.6.4.2 ACO_DEV.3.2E

评估者应确定所提供的接口描述与为依赖部件提供的依赖信息是一致的。

16.3 依赖部件的依赖性(ACO_REL)

16.3.1 目的

本族的目的是提供描述依赖部件依赖基本部件的依赖性证据。这些信息对于负责组合该部件和其他已评估过的 IT 部件形成组合 TOE,并负责解释组合之后的安全属性的人员是有用的。

提供组合 TOE 的依赖部件和基本部件之间的接口描述,这些接口或许在它们各自部件评估期间没有得到分析,因为这些接口不是它们各自部件 TOE 的 TSFI。

16.3.2 组件分级

本族组件是根据依赖部件依赖基本部件的依赖性的描述的细节数量分级的。

16.3.3 应用注释

依赖部件的依赖性族(ACO_REL)分析部件间的相互作用,依赖部件依靠基本部件提供的支持其安全功能运行的服务。基本部件的这些服务接口,或许在基本部件评估期间没有经过分析,因为基本部件的这些服务在过去部件评估中被认为是与安全无关的,或者因为这种服务的本质用途(如,调整字体),再或者因为与 ISO/IEC 15408 相关的安全功能要求在基本部件的 ST 中没有声明(如,当在 ISO/IEC 15408-2 的 FIA:标识和鉴别安全功能要求中没有声明登录接口时)。在基本部件评估过程中,基本部件的这些接口经常视为功能接口,作为额外的安全接口(TSFI)在功能规范中加以考虑。

总的来说,功能规范中 TSFI 描述仅包括外部实体调用 TSF 接口以及这些调用的反馈。在这些部件评估期间没有明确考虑到的被 TSF 调用的接口,这些将在为了满足“依赖部件的依赖性”族(ACO_REL)要求提供的依赖信息中描述。

16.3.4 ACO_REL.1 基本的依赖信息

依赖组件:无。

16.3.4.1 开发者行为元素

16.3.4.1.1 ACO_REL.1.1D

开发者应提供依赖部件的依赖信息。

16.3.4.2 内容和形式元素

16.3.4.2.1 ACO_REL.1.1C

依赖信息应描述依赖部件 TSF 依赖的基本部件硬件、固件和/或软件的功能。

16.3.4.2.2 ACO_REL.1.2C

依赖信息应描述依赖部件 TSF 从基本部件请求服务的所有交互。

16.3.4.2.3 ACO_REL.1.3C

依赖信息应描述依赖 TSF 是如何保护自身,以避免被基本部件干扰和削弱。

16.3.4.3 评估者行为元素

16.3.4.3.1 ACO_REL.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

16.3.5 ACO_REL.2 依赖信息

依赖组件:无。

16.3.5.1 开发者行为元素

16.3.5.1.1 ACO_REL.2

开发者应提供依赖部件的依赖信息。

16.3.5.2 内容和形式元素

16.3.5.2.1 ACO_REL.2.1C

依赖信息应描述依赖部件 TSF 依赖的基本部件硬件、固件和/或软件的功能。

16.3.5.2.2 ACO_REL.2.2C

依赖信息应描述依赖部件 TSF 从基本部件请求服务的所有交互。

16.3.5.2.3 ACO_REL.2.3C

依赖信息应根据使用的接口和这些接口的返回值来描述每种交互。

16.3.5.2.4 ACO_REL.2.4C

依赖信息应描述依赖 TSF 是如何保护自身,以避免被基本部件干扰和削弱。

16.3.5.3 评估者行为元素

16.3.5.3.1 ACO_REL.2.1E

评估者应**确认**所提供的信息满足证据的内容和形式的所有要求。

16.4 组合 TOE 测试(ACO_CTT)

16.4.1 目的

本族要求执行组合 TOE 测试以及组合 TOE 中用到的基本部件测试。

16.4.2 组件分级

本族组件是基于进行接口测试时不断增加的严格性,以及分析证实组合 TOE 的运行是与依赖信息和组合 TOE 安全功能要求符合性测试的充分性不断增加的严格性而分级的。

16.4.3 应用注释

与本族有关的,两个不同方面的测试如下:

- a) 依赖部件执行安全功能所依赖的基本部件和依赖部件之间的接口测试,目的是证实它们的兼容性;
- b) 组合 TOE 测试,目的是证实 TOE 运行与组合 TOE 的安全功能要求是一致的。

如果评估期间依赖部件使用的测试配置项包括把基本部件作为一个“平台”使用,而且测试分析充分证实了 TSF 是按照安全功能要求去运行的,则开发者不需要执行进一步的组合 TOE 功能测试。但是,如果在依赖部件的测试中基本部件没有被使用,或者任一部件的配置项改变了,那么开发者应该执行组合 TOE 的测试。可采取复测依赖部件开发者对依赖部件的测试的形式提供证据,充分证实组合 TOE TSF 的运行与安全功能要求是一致的。

开发者应提供对组合中用到的基本部件接口的测试证据。基本部件 TSFI 的运行可能已经作为 ATE;基本部件评估期间的测试活动的一部分通过了测试。因此,提供适当包含在基本部件评估的测试样例中的接口,组合基本原理(ACO_COR)确定基本部件按照基本部件评估配置项在运行,并且具有依赖部件 TSF 要求的所有安全功能,那么评估者行为 ACO_CTT.1.1E 可以通过重用基本部件的 ATE;测试结论来满足。

如果不是这种情形,组合中使用的基本部件接口,可能受到任何变更评估配置项的影响,那么所有额外的安全功能都要被测试,以确保它们能证实预期行为。需要被测试的预期行为是在依赖信息中所描述的那些(依赖部件的依赖性证据(ACO_REL))。

16.4.4 ACO_CTT.1 接口测试

依赖组件:ACO_REL.1 基本的依赖信息;
ACO_DEV.1 功能描述。

16.4.4.1 目的

本组件的目的是确保依赖部件所依赖的基本部件的每个接口都通过了测试。

16.4.4.2 开发者行为元素

16.4.4.2.1 ACO_CTT.1.1D

开发者应提供组合 TOE 测试文档。

16.4.4.2.2 ACO_CTT.1.2D

开发者应提供基本部件接口测试文档。

16.4.4.2.3 ACO_CTT.1.3D

开发者应提供用于测试的组合 TOE。

16.4.4.2.4 ACO_CTT.1.4D

开发者应提供与基本部件开发者执行基本部件功能测试所用的同等的一系列资源。

16.4.4.3 内容和形式元素

16.4.4.3.1 ACO_CTT.1.1C

组合 TOE 和基本部件接口测试文档应由测试计划、预期测试结果和实际测试结果组成。

16.4.4.3.2 ACO_CTT.1.2C

开发者执行组合 TOE 测试的测试文档应证实 TSF 按照规定执行。

16.4.4.3.3 ACO_CTT.1.3C

开发者执行基本部件接口测试的测试文档应证实依赖部件所依赖的基本部件接口按照规定运行。

16.4.4.3.4 ACO_CTT.1.4C

基本部件应适于测试。

16.4.4.4 评估者行为元素

16.4.4.4.1 ACO_CTT.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

16.4.4.4.2 ACO_CTT.1.2E

评估者应执行测试文档中的测试样本,以验证开发者的测试结果。

16.4.4.4.3 ACO_CTT.1.3E

评估者应测试组合 TOE TSF 接口的一个子集,以确认复合 TSF 按照规定执行。

16.4.5 ACO_CTT.2 严格的接口测试

依赖组件: ACO_REL.2 依赖信息;
ACO_DEV.2 基本的设计证据。

16.4.5.1 目的

本组件的目的是确保依赖部件所依赖的基本部件的每个接口都通过了测试。

16.4.5.2 开发者行为元素

16.4.5.2.1 ACO_CTT.2.1D

开发者应提供组合 TOE 测试文档。

16.4.5.2.2 ACO_CTT.2.2D

开发者应提供基本部件接口测试文档。

16.4.5.2.3 ACO_CTT.2.3D

开发者应提供用于测试的组合 TOE。

16.4.5.2.4 ACO_CTT.2.4D

开发者应提供与基本部件开发者执行基本部件功能测试所用的同等的一系列资源。

16.4.5.3 内容和形式元素

16.4.5.3.1 ACO_CTT.2.1C

组合 TOE 和基本部件接口测试文档应由测试计划、预期测试结果和实际测试结果组成。

16.4.5.3.2 ACO_CTT.2.2C

开发者执行组合 TOE 测试的测试文档应证实 TSF 按照规定运行**并且是完整的**。

16.4.5.3.3 ACO_CTT.2.3C

开发者执行基本部件接口测试的测试文档应证实依赖部件所依赖的基本部件接口按照规定运行**并且是完整的**。

16.4.5.3.4 ACO_CTT.2.4C

基本部件应适于测试。

16.4.5.4 评估者行为元素

16.4.5.4.1 ACO_CTT.2.1E

评估者应**确认**所提供的信息满足证据的内容和形式的所有要求。

16.4.5.4.2 ACO_CTT.2.2E

评估者应**执行**测试文档中的测试样本,以验证开发者的测试结果。

16.4.5.4.3 ACO_CTT.2.3E

评估者应**测试**组合 TOE TSF 接口的一个子集,以确认复合 TSF 按照规定执行。

16.5 组合脆弱性分析(ACO_VUL)

16.5.1 目的

本族要求分析公共可用脆弱性信息,同时分析由于组合可能引入的脆弱点。

16.5.2 组件分级

本族的组件是基于审查的公共脆弱性信息和进行独立脆弱性分析不断增加的详细程度而分级的。

16.5.3 应用注释

开发者应提供部件评估期间报告的任何残余脆弱点的详细资料。这些或许能从部件开发者或部件

的评估报告中获得。这些将用作评估者对组合 TOE 在运行环境中脆弱性分析的输入。

检查组合 TOE 的运行环境,确保在组合 TOE 中,各部件运行环境(每个部件的 ST 中规定)的假设和目的是满足的。在实施组合 TOE 的 ASE 活动期间,已经对各部件和组合 TOE ST 之间的假设和目的一致性进行了初始分析。但是,随着 ACO_REL,ACO_DEV 和 ACO_COR 评估活动期间信息的不断获得,还需要对这个分析进行重新审查,确保,例如,依赖部件的假设是依赖部件 ST 中环境所陈述的,而不是由于组合重新提出的(例如,基本部件充分地处理了组合 TOE 中依赖部件 ST 的假设)。

评估者针对每个部件执行问题调查,标识自从部件评估完成之后,公共领域报告的潜在脆弱点。每个脆弱点都要进行测试。

如果组合 TOE 用到的基本部件是保障连续性证明活动的主体,评估者将考虑组合 TOE 脆弱性分析活动期间基本部件的变化。

16.5.4 ACO_VUL.1 组合脆弱性审查

依赖组件:ACO_DEV.1 功能描述。

16.5.4.1 开发者行为元素

16.5.4.1.1 ACO_VUL.1.1D

开发者应提供用于测试的组合 TOE。

16.5.4.2 内容和形式元素

16.5.4.2.1 ACO_VUL.1.1C

组合 TOE 应适于测试。

16.5.4.3 评估者行为元素

16.5.4.3.1 ACO_VUL.1.1E

评估者应确认所提供的信息满足证据的内容和形式的所有要求。

16.5.4.3.2 ACO_VUL.1.2E

评估者应执行分析,确定基本部件和依赖部件标识的所有残余脆弱性,在组合 TOE 运行环境中都不能被利用。

16.5.4.3.3 ACO_VUL.1.3E

评估者应执行公共领域信息的调查,标识在组合 TOE 运行环境中使用基本部件和依赖部件产生的潜在脆弱点。

16.5.4.3.4 ACO_VUL.1.4E

评估者应根据已标识的脆弱点,实施穿透性测试,证实组合 TOE 能抵抗具有基本攻击潜力的攻击者进行的攻击。

16.5.5 ACO_VUL.2 组合脆弱性分析

依赖组件:ACO_DEV.2 基本的设计证据。

16.5.5.1 开发者行为元素

16.5.5.1.1 ACO_VUL.2.1D

开发者应提供用于测试的组合 TOE。

16.5.5.2 内容和形式元素

16.5.5.2.1 ACO_VUL.2.1C

组合 TOE 应适于测试。

16.5.5.3 评估者行为元素

16.5.5.3.1 ACO_VUL.2.1E

评估者应**确认**所提供的信息满足证据的内容和形式的所有要求。

16.5.5.3.2 ACO_VUL.2.2E

评估者应**执行**分析,确定基本部件和依赖部件标识的所有残余脆弱性,在组合 TOE 运行环境中都不能被利用。

16.5.5.3.3 ACO_VUL.2.3E

评估者应**执行**公共领域信息的调查,标识在组合 TOE 运行环境中使用基本部件和依赖部件产生的潜在脆弱点。

16.5.5.3.4 ACO_VUL.2.4E

评估者应**利用指导性文档、依赖信息和组合基本原理**对组合 TOE 执行**独立的脆弱性分析**,标识组合 TOE 的潜在脆弱点。

16.5.5.3.5 ACO_VUL.2.5E

评估者应根据已标识的脆弱点,**实施**穿透性测试,证实组合 TOE 能抵抗具有基本攻击潜力的攻击者进行的攻击。

16.5.6 ACO_VUL.3 增强的基本组合脆弱性分析

依赖组件:ACO_DEV.3 详细的设计证据。

16.5.6.1 开发者行为元素

16.5.6.1.1 ACO_VUL.3.1D

开发者应提供用于测试的组合 TOE。

16.5.6.2 内容和形式元素

16.5.6.2.1 ACO_VUL.3.1C

组合 TOE 应适于测试。

16.5.6.3 评估者行为元素

16.5.6.3.1 ACO_VUL.3.1E

评估者**应确认**所提供的信息满足证据的内容和形式的所有要求。

16.5.6.3.2 ACO_VUL.3.2E

评估者**应执行**分析,确定基本部件和依赖部件标识的所有残余脆弱性,在组合 TOE 运行环境中都不能被利用。

16.5.6.3.3 ACO_VUL.3.3E

评估者**应执行**公共领域信息的调查,标识在组合 TOE 运行环境中使用基本部件和依赖部件产生的潜在脆弱点。

16.5.6.3.4 ACO_VUL.3.4E

评估者**应**利用指导性文档、依赖信息和组合基本原理对组合 TOE **执行**独立的脆弱性分析,标识组合 TOE 的潜在脆弱点。

16.5.6.3.5 ACO_VUL.3.5E

评估者**应根据**已标识的脆弱点,**实施**穿透性测试,证实组合 TOE 能抵抗具有**增强型基本**攻击潜力的攻击者进行的攻击。

附 录 A
(资料性附录)
开发(ADV)

本附录包含辅助材料,为 ADV(开发类)的族中提出的话题做更进一步的解释和提供额外的例子。

A.1 ADV_ARC:安全架构的补充材料

安全架构是 TSF 所展示的属性集合;这些属性包括自保护、域分离和不可旁路。拥有这些属性给予了 TSF 提供安全服务的信心基础。本附录给出了这些属性的附加材料,以及在安全架构描述内容上的讨论。

A.1.1~A.1.2 首先解释了这些属性,然后讨论描述 TSF 如何展示那些特性所需要的信息。

A.1.1 安全架构属性

自保护是 TSF 用于保护自身的能力,以抵御外部实体可能导致 TSF 改变的操作。没有这些属性,TSF 可能无法执行它的安全服务。

常常有这种情况,TOE 使用其他 IT 实体提供的服务或资源来执行它的功能(例如依赖它下层操作系统的应用)。在这些情况中,TSF 不完全凭借自身保护自己,因为它依赖于其他 IT 实体保护其使用的服务。

域分离是这样一种属性,TSF 为每个操作其资源的不可信激活实体创建各自的安全域,并维持那些域彼此分离,这样没有实体能在其他实体域中运行。例如,操作系统 TOE 为每个与不可信实体相关的过程提供一个域(地址空间、进程环境变量)。

对有些 TOE 来说这样的域是不存在的,因为所有不可信实体的行为都由 TSF 来代理。包过滤防火墙就是此类 TOE 的一个例子,它没有不可信实体域;只有由 TSF 所维护的数据结构。那么,域的存在性依赖于 1)TOE 类型以及 2)TOE 的安全功能要求。TOE 若提供针对不可信实体的域,本族要求拥有不可信实体的域要与其他域隔离,以免受其他不可信实体域的干扰(不受 TSF 控制的影响)。

不可旁路性是 TSF(用安全功能要求说明)安全功能经常调用的一个属性,并且对于特定机制它不能被绕过。例如,通过安全功能要求对文件的访问控制被说明为 TSF 的一种能力,那么在调用 TSF 的访问控制机制(通过原始磁盘访问可能是这样一个接口的例子)之前必须不能存在直接访问此文件的接口。

拿自保护做例子,有些 TOE 可能很自然地依赖它们所处的环境,在 TSF 的不可旁路性中发挥作用。例如,某安全应用 TOE 要求它被底层操作系统调用。类似的,防火墙依赖于这样一个事实,即不存在内部网络和外部网络的直接连接通路,所有它们之间的通讯必须通过防火墙。

A.1.2 安全架构描述

安全架构的描述解释 TSF 如何展示上面描述的这些属性。它描述域如何定义,TSF 之间如何保持隔离。它描述怎么阻止不可信进程获得 TSF 并修改它。它描述怎样确保 TSF 控制之下的所有资源得到充分的保护,以及确保 TSF 所要满足的安全功能要求相关的所有行为。它解释所有情况下(例如,假定它的低层环境调用正确,它的安全功能如何被调用?)环境扮演的角色。

安全架构以分解描述的方式描述展现 TSF 的自保护、域分离和不可旁路性属性。描述的级别与 ADV_FSP,ADV_TDS 和 ADV_IMP 要求的 TSF 描述相当。例如,如果 ADV_FSP 是仅有的 TSF 描述,那么就很难提供明确的架构级设计,因为无法获取可用的 TSF 内部工作细节。

然而,如果可获得 TOE 设计,即使最基础的级别(ADV_TDS.1),也会有一些构成 TSF 子系统的相关信息,或许也会有它们怎样执行自保护、域分离和不可旁路性的描述。例如,或许用户与 TOE 的所有交互被强制限制到某个步骤,服务于用户,选定所有用户的安全属性;架构设计应描述这样的进程如何产生,此进程的行为如何受 TSF 限制(因此它不能破坏 TSF),TSF 如何协调该进程中的所有行为(因此可解释为什么 TSF 不可被旁路),等等。

如果可获得的 TOE 设计更详细(例如,在模块级),或实现表示也可获得,那么对架构设计的描述相应地也要更详细,说明用户如何与 TSF 进行交互的过程,TSF 如何处理不同的请求,什么样的参数通过,什么样的程序保护(缓冲区预防,参数边界检查,校验时间/使用校验的时间,等等)在生效。类似的,若在 ST 中声明了 ADV_IMP 组件的 TOE 应加入执行相关的细节。

在安全架构描述中提供的解释应足够详细,应让人可判断其准确度。换句话说,简单的声明(例如,TSF 保持域分离)没有提供有用的信息让读者确信 TSF 的确创建并隔离了域。

A.1.2.1 域分离

如果 TOE 展现的域分离整个由其自己实现,应就如何实现有一个明确的描述。安全架构描述应解释由 TSF 定义的不同类型的域,怎么定义它们(例如,每个域中分配什么资源),所有资源是如何被保护的,域是如何被隔离的,因此在某域中的活动实体无法破坏另一个域中的资源。

对于 TOE 依赖其他 IT 实体来实现域分离的情况,公用的角色必须要明确说明。例如,某 TOE 是一个应用软件,它依赖底层操作系统去正确地呈现 TOE 定义的域;如果对于每个域,TOE 定义了分离进程空间,内存空间等,它依赖底层操作系统去正确且良好地执行(例如,仅允许在 TOE 软件申请的执行空间中执行进程)。

例如,执行域分离的机制(例如,内存管理,由硬件提供的受保护的受保护的处理模式,等等)将被识别和描述。要么,TSF 可以执行软件保护结构或编码习惯以有助于执行软件域分离,也可能通过将用户地址空间从系统地址空间勾画出来去实现。

脆弱性分析和测试(参见 AVA_VAN)活动应包括通过使用监控或直接攻击 TSF 的方式去破坏所描述的 TSF 域分离。

A.1.2.2 TSF 自保护

如果 TOE 展现的自保护整个是由其自身实现,应就如何实现有一个明确的描述。提供域分离的机制是为了定义 TSF 保护域,以阻止其他(用户)域,此机制应被识别并描述。

对于 TOE 依赖其他 IT 实体来实现保护自身的情况,公用的角色必须要明确说明。例如,某 TOE 是一个独立应用软件,它依赖底层操作系统去正确、良好地执行;该应用软件无法保护自身以对抗恶意操作系统对其破坏(例如,覆写其执行代码或 TSF 数据)。

安全架构描述也包括 TSF 如何处理用户输入,从而使 TSF 不会被用户输入所破坏。例如,TSF 可执行权限识别,并通过使用特权模式程序来处理用户数据从而自我保护。TSF 有可能使用基于处理器的隔离机制(例如,权限级别或环级别)将 TSF 代码和数据与来自于用户的代码和数据进行隔离。TSF 可以执行软件保护结构或编码习惯以有助于执行软件隔离,也可能通过将用户地址空间从系统地址空间勾画出来去实现。

有些 TOE 以消减功能模式(例如,单用户模式仅易于安装者或管理员进入)启动,接着转变至经评估之后的安全配置(不可信用户可以登录并使用 TOE 服务和资源的模式),安全架构描述也应包括对 TSF 如何保护该初始化代码不在评估后的配置模式下运行的描述。对于这样的 TOE,安全架构描述应解释是什么防止了仅初始化过程中可使用的服务(例如,直接访问资源)被评估后的配置所访问。还应解释当 TOE 正处于评估后的配置时,是什么防止了初始化代码被运行。

必须还要解释,可信初始化代码是如何维护 TSF(以及它的初始化进程)完整性的,这样初始化进

程能够探测到可能导致 TSF 被欺骗以致相信它处于一个初始安全状态的所有的修改。

脆弱性分析和测试(参见 AVA_VAN)活动应包括通过使用篡改、直接攻击或监控 TSF 的方式去破坏所描述的 TSF 自我保护。

A.1.2.3 TSF 不可旁路性

不可旁路性的特性与那些允许旁路执行机制的接口有关。在大多数情况下与实现是有因果关系的,如果某程序员在编写了一个接口去访问或使用某对象时,针对这个对象的某些接口是安全功能要求机制的一部分,该程序员有责任使用这些接口,而不能尝试绕过这些接口。由于描述的是不可旁路性,那么有两个方面是必须要覆盖到的。

首先是组成 SFR-执行的这些接口。这些接口的属性是它们当中不能包含用于旁路 TSF 的操作或模式。ADV_FSP 和 ADV_TDS 使用的证据可能在很大程度上可被用于做出这个决定。因为不可旁路性关注的是,如果仅有可用在这些 TSFI 上的特定操作在文档(因为他们是 SFR-执行)中作了声明,其他操作都不可用,开发者应考虑是否额外的信息也有必要做出声明,即 SFR-支撑以及 SFR-无关的 TSFI 的操作不会提供给一个不可信实体去旁路正在执行的策略。如果此类信息有必要的话,也要被包含进安全架构描述。

不可旁路性的第二个方面涉及与 SFR-执行不相关的那些接口。依赖于 ADV_FSP 和 ADV_TDS 组件的声明,这些接口中的有些信息可能在功能规范和 TOE 设计文档中,也可能不在其中。这样的接口(或接口组)的描述信息应充分,使得读者能够确定执行机制不可被旁路(描述的详细程度与 ADV:开发类中提供的其他证据相当)。

安全功能不能被旁路的属性均应被应用于所有的安全功能。即设计描述应覆盖安全功能要求保护下的目标(例如 FDP_* 组件)和 TSF 所提供的功能(例如,审计)。这些描述也应标识出那些与安全功能相关联的接口;也可以利用功能规范中使用的信息。该描述还应包括所有设计构件,例如目标管理器,以及它们的使用方法。例如,如果程序是被用作一个标准的宏来处理一个审计记录,该规定则是设计的一部分,促成了审计机制的不可旁路性。很重要的一个问题是,本文中不可旁路性不是尝试回答此问题“如果是恶意的话,TSF 执行的某部分是否可旁路安全功能”,而是在文档中描述如何去执行不会旁路安全功能。

脆弱性分析和测试(参见 AVA_VAN)活动应包括通过绕过 TSF 的方式去破坏所描述的不可旁路性。

A.2 ADV_FSP:TSFI 补充材料

制定 TSFI 的目的是为引导测试提供必须的信息;在不知道与 TSF 交互的各种可能途径的情况下,不可能充分测试 TSF 的行为。

要声明 TSFI 的两个方面:标识它们和描述它们。由于 TOE 可能的多样性,以及不同的 TSF,没有组成“TSFI”的标准接口集合。本附录提供的指南是关于确定哪些接口是 TSFI 的要素。

A.2.1 确定 TSFI

为了标识 TSF 接口,必须首先识别构建 TSF 的 TOE 部分。这种识别实际上是 TOE 设计(ADV_TDS)分析的一部分,但是在保障包中不包含 TOE 设计(ADV_TDS)的情况下,开发者会暗含着(通过 TSFI 识别和描述)执行识别。在此分析中,如果 TOE 的一部分能够满足 ST(全部或部分)中的安全功能要求,那么必须认为它是 TSF 的一部分。例如,这包括 TOE 中用于 TSF 运行时初始化的所有细节,比如安全功能要求的执行没有开始(例如启动时)优先于 TSF 运行的软件能自我保护。对 TSF 自保护、域分离和不可旁路的架构原则提供支持的 TOE 所有部分也包含在 TSF 中(参加安全架构 ADV_ARC)。

一旦定义了 TSF,也就表明了 TSFI。TSFI 是由用户调用 TSF(通过提供经 TSF 处理过的数据)服务的所有途径以及对这些服务的响应所构成。这些服务规定和响应是穿越 TSF 边界的方法。当然,有些服务会很明显,有些则不那么明显。当确定 TSFI 的时候要问的问题是“一个企图破坏安全功能要求的潜在攻击者是如何与 TSF 进行交互的?”下面的讨论将展现 TSFI 定义在不同情境下的应用。

A.2.1.1 电气接口

在智能卡这样的 TOE 中,攻击者不仅仅能够逻辑访问 TOE,而且能够完全物理访问 TOE,TSF 的边界是物理边界。因此,所暴露的电口应被纳为 TSFI,因为它们的操作有可能影响到 TSF 的行为。因此,所有此类接口(电气触点)均需描述:可适用的多种电压,等等。

A.2.1.2 网络协议栈

对于执行协议处理的 TOE 来说,潜在攻击者可直接访问的 TSFI 应该是这些协议层。这不一定是整个协议栈,但也有可能是。

例如,TOE 是一系列网络应用的集合,允许潜在攻击者影响协议栈的每个级别(例如,发送任意信号,任意电压,任意信息包,任意数据,等等),那么 TSF 的边界存在于协议栈的每层。因此,功能规范应处理每个协议栈的所有层次。

但是,如果 TOE 是个保护内网防止外网攻击的防火墙,潜在攻击者可能无法直接利用电压而进入 TOE,所有极端的电压都不能穿透到因特网。即攻击者只能访问到网络层或更上层的协议。TSF 边界存在于协议栈的每个层次。因此,功能规范应只描述到网络层或更上层的协议:它应根据有可能出现在网路上的正常输入描述防火墙所暴露的每个不同通讯层,以及正常和恶意输入所导致的后果。例如,网络协议层的描述应包括一个良好形态的 IP 包怎么构成,当同时接受到正确形态和错误形态的数据包时会发生什么。同样的,TCP 层的描述应包括成功 TCP 连接,当同时发生成功建立连接和不成功建立连接或异常丢弃时会发生什么。假设防火墙的目的是为了过滤应用层命令(例如 FTP 或 TELNET),对应用层的描述应包括防火墙能够识别和过滤的应用层命令,以及碰到未知命令时的处理结果。

对这些层的描述可参考公开发布的正使用的通讯标准(TELNET、FTP、TCP,等等),可注释出用户定义的选项。

A.2.1.3 外包

“封装程序”将复杂的系列化的交互动作翻译成简单的共用服务,例如,当操作系统创建应用程序所使用(见图 A.1)的 API。无论 TSFI 是系统调用还是依赖于应用程序使用的 API:如果应用程序可以直接使用系统调用,那么系统调用就是 TSFI 接口。不过,如果禁止它们直接调用并且要求所有交互信息都通过 API,那么 API 就应是 TSFI。

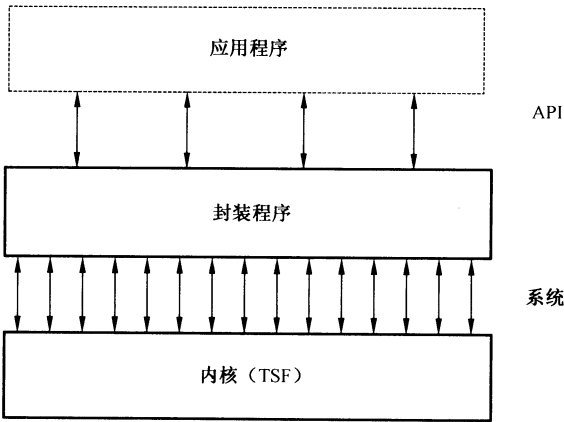


图 A.1 封装程序图

图形化的用户接口也类似：它在机器理解的命令和用户友好界面之间转换。类似的，如果用户访问到命令则它们就是 TSFI，或者用户被强制使用图形则图形（按钮，检查框，文本域）就是 TSFI。

在这两种例子中，若用户被禁止使用更原始的接口（例如系统调用或命令），值得注意的是对这种限制和强制的描述应被包含进安全架构描述（见图 A.1）。同样，封装程序也是 TSF 的一部分。

A.2.1.4 不可访问接口

对于特定 TOE，不是所有接口都是可访问的。即操作环境（安全目标中的）的安全目的有可能禁止访问这些接口，或者以几乎不可访问的方式限制访问。这样的接口不应被列为 TSFI。以下是一些例子：

- a) 如果对于单独防火墙操作环境的安全目标的陈述为“防火墙的操作将会在一间服务器房间环境中，该房间只允许可信且经过培训的个人能访问，它还装配有不间断电源供应（应对电力故障）”，物理以及电力接口将是不可访问的，因为可信且培训过的人将不会试图拆除防火墙并/或中断电力供应。
- b) 如果对于软件防火墙（应用程序）操作环境的安全目标的陈述为“操作系统和硬件将为应用程序提供安全域以防止其他程序的篡改”，那么在操作系统中可被其他程序（例如，删除或修改防火墙可执行文件，直接读或写防火墙的内存空间）访问防火墙的接口就应是不可访问接口，因为操作系统/硬件部分操作环境使得此接口不可访问。
- c) 如果对于软件防火墙操作环境的安全目标的描述为操作系统和硬件会如实地执行 TOE 命令，并且不会以任何方式篡改 TOE，那么防火墙用以从操作系统和硬件（执行机器代码指令，操作系统 API，例如创建、读、写或删除文件，图形 API 等）获取原始功能的接口应是不可访问的，因为操作系统/硬件是能够访问此接口的仅有的实体，并且它们全都可信。

对上述所有例子，这些不可访问接口就不是 TSFI。

A.2.2 范例：复杂的 DBMS

图 A.2 举了一个复杂 TOE 的例子：依赖于 TOE 范围之外的硬件和软件的数据库管理系统（涉及像本讨论其余部分中的 IT 环境）。为了简化本例子，TOE 等同于 TSF。带阴影的正方形代表 TSF，不带阴影的方形代表环境中的 IT 实体。TSF 包含数据库引擎、管理 GUI（标为 DB 的方形表示）和作为 OS 的一部分运行的执行一些安全功能的核心模块（标为 PLG 的方形表示）。TSF 核心模块有 OS 规范定义的入口的点，OS 会调用某些功能（这可能是设备驱动或者鉴权模块等）。关键在于这个可嵌入的核心模块提供了 ST 中功能要求描述的安全服务。

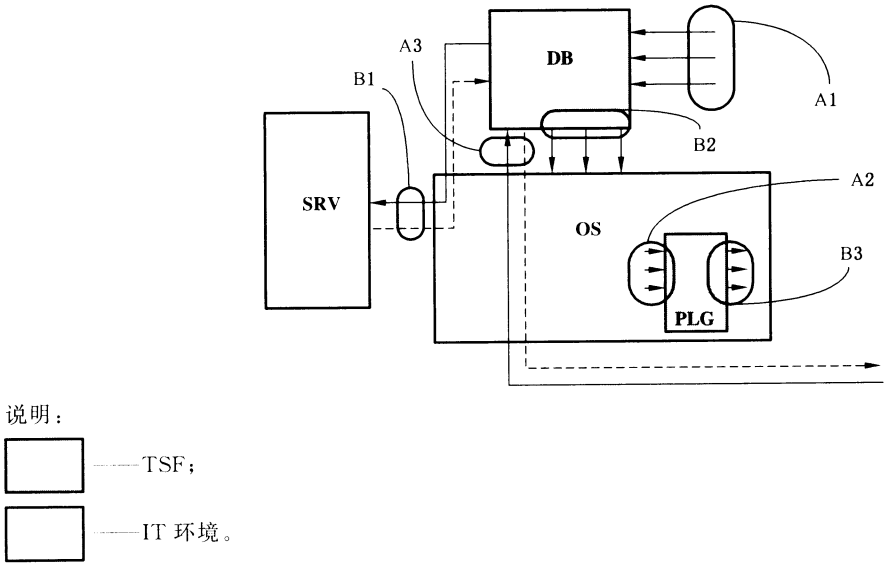


图 A.2 DBMS 系统的接口

IT 环境由操作系统本身(标为OS 的框)以及一个外部服务器(标为SRF)构成。就像操作系统,这个外部服务器提供 TSF 所依赖的某个服务,因此也要被列为 IT 环境。图中Ax 表示 TSFI,Bx 表示在 ACO:组合类中描述的其他接口。每个这样的接口组讨论如下。

接口组 A1 代表 TSFI 中最明显的集合。这些是用户用来直接访问数据库、其安全功能和资源的接口。

接口组 A2 代表 TSFI 中的操作系统调用以获得可嵌入模块所提供的功能。它们与 B3 类型的接口形成对照,B3 代表了可嵌入模块进行的调用以获取 IT 环境中的服务。

接口组 A3 代表了通过 IT 环境的 TSFI。在这种情况下,DBMS 使用一个私有应用级协议进行网络通讯。而 IT 环境负责提供许多支持协议(例如,以太、IP、TCP),用于从 DBMS 获取服务的应用层协议是一个 TSFI,必须在文档中描述。点线表示通过网络连接 TSF 的返回值/服务。

Bx 接口代表 IT 环境中的功能性接口。这些接口不是 TSFI,只有当 TOE 作为 ACO 类活动的一个部分被用于组合评估时才有必要讨论并分析它们。

A.2.3 功能规范范例

在内外网之间使用的防火墙举例。它验证接收到的数据的源地址(确保外部数据不能试图伪装为源自内部的数据);如果它检测到任何这样的企图,它会将侵犯的企图保存为审计日志。通过建立从内部网络到防火墙的 telnet,连接管理员到防火墙。管理员行为可包含鉴权、修改密码、浏览审计日志以及设置或修改内外网的地址。

例如,防火墙为内部网络提供了如下接口:

- a) IP 数据包;
- b) 管理员命令。

为外部网络提供了如下接口:

- a) IP 数据包。

接口描述:IP 数据包

数据包符合 RFC791 中描述的格式。

- 目的:从固定长度地址的源主机向目的主机传输数据块(“数据包”);同时,当通过小包网络进行传输时,如果有必要对长数据报提供分片和重组。
- 使用方法:它们出自底层(例如数据链路层)协议。
- 参数:下列 IP 数据包头:源地址,目的地址,分片标志位。
- 参数描述:[在 RFC791 中定义的(“3.1 网际包头格式”)]。
- 动作:传送非伪造的数据包;如果必要将大数据包进行分片;将分片数据重组形成数据包。
- 错误消息:(无)。无可靠性来担保(可靠性由上层协议提供)不可达数据包(例如,传输是必须要分片,但分片标志位未被设置)的丢弃。

接口描述:管理员命令

管理员命令为管理员提供了一条通路以与防火墙交互。这些命令和应答倚靠在任意内部网络主机所建立的一条 telnet 连接。可用的命令有:

- 密码
 - 目的:管理员密码集合;
 - 使用方式:Passwd <password>;
 - 参数:密码;
 - 参数描述:新密码的值;
 - 动作:将密码改变为新提供的值。没有什么约束;
 - 错误信息:无。

- **读审计**
 - 目的:为管理员呈现审计日志;
 - 使用方式:**Readaudit**;
 - 参数:无;
 - 参数描述:无;
 - 动作:提供审计日志文本;
 - 错误信息:无。
- **设置内部地址**
 - 目的:设置内部网络地址;
 - 使用方式:**Setintaddr** *<address>*;
 - 参数:地址;
 - 参数描述:IP 地址(RFC791 中定义)的前三个域。例如,123.123.123;
 - 动作:改变用于定义内部网络的变量的内部值,该值用于判断伪装尝试;
 - 错误信息:“地址正在使用”:表明所识别出的地址与外部网络地址相同。
- **设置外部地址**
 - 目的:设置外部地址;
 - 使用方式:**Setextaddr** *<address>*;
 - 参数:地址;
 - 参数描述:IP 地址(RFC791 中定义)的前三个域。例如,123.123.123;
 - 动作:改变用户定义外部网络的变量的内部值;
 - 错误信息:“地址正在使用”:表明所识别出的外部地址与内部网络地址相同。

A.3 ADV_INT:TSF 内部辅助材料

TOE 的多样性使得将所有东西整理地“结构清晰”或“最小复杂度”变得不可能。对结构化及复杂度的考量来自于 TOE 中使用的特定技术。例如,如果软件所展现出的特征符合软件工程的要求,则其可被认为结构清晰。

本附录所提供的辅助材料用于评估 TSF 基于过程的软件结构化和复杂度。这些材料基于软件工程著作中众所周知的信息。对于其他类型的内部构件(例如,硬件,面向对象的代码这样的非过程化软件等),也会考量相应最佳实践方面的著作。

A.3.1 程序软件的结构

传统上按照程序软件的模块化程度评估程序软件的结构。用模块化设计编写的软件易于理解,该理解是通过阐明一个模块与其他模块的依赖关系(耦合),并且通过包含一个模块中仅与其他任务之间有强关联的任务(内聚)来实现的。模块化设计的使用减少了 TSF 元素之间的相互依赖,从而降低了因一个模块中的修改或错误将影响到整个 TOE 的风险。它的使用增强了设计的清晰度并增强了不发生非预期影响的保障。模块分解可取的附加属性是减少了冗余或非必要代码的数量。

最小化 TSF 中功能的数量使得评估者以及开发者只关注于 SFR 执行必需的功能、对易于理解贡献更多并更能降低设计或实现错误的可能性。

模块化解解、分层以及设计和实现过程中的简化,必须辅之以完善的软件工程的考虑。一个实用的、有用的软件系统通常会伴随着模块之间的一些不良耦合,一些松散相关的功能模块,以及模块设计中微妙或复杂之处。基于开发者的论证,这些与理想中模块化解解的偏差往往被视为要实现某种目标或约束所必须的,与性能、兼容性、未来的计划功能或其他一些因素有关,并且是可以接受的。在应用本

类的要求中,必须充分考虑到合理的软件工程原则;但是,无论如何要达到易懂性的整体目标。

A.3.1.1 内聚

内聚是单一软件模块所执行的任务彼此之间的关联方式和程度。内聚的类型包括一致、通信、功能、逻辑、顺序和暂时。下面描述了这些类型内聚的特征,以需求性递减的顺序列出。

- a) 功能内聚:带有功能内聚的模块执行与单一目的相关的活动。功能内聚的模块将单一的输入类型转换成单一的输出类型,如一个堆栈管理器或队列管理器;
- b) 顺序内聚:带有顺序内聚的模块包含多个功能,其中每个功能的输出是其后续功能的输入。顺序内聚模块的一个例子是包含写审计记录和维护特定类型审计侵害持续累积计数功能的模块;
- c) 通信内聚:带有通信内聚的模块包含为模块中的其他函数产生输出,或使用模块内其他函数的输出。通信内聚模块的一个实例是包含强制、自主以及能力验证的访问验证模块;
- d) 暂时内聚:带有暂时内聚的模块包含着需要几乎同时执行的各类功能。暂时内聚模块的例子如初始化、恢复和关机模块;
- e) 逻辑(或程序)内聚:带有逻辑内聚的模块对不同数据结构执行类似活动。当模块功能对于不同的输入执行相关的、但又不同的操作,表现为逻辑内聚;
- f) 偶然内聚:带有偶然内聚的模块执行无关或松散的活动。

A.3.1.2 耦合

耦合是软件模块之间相互依赖的方式和程度;耦合的类型包括调用、公有和内容耦合。这些类型的耦合特征如下,以需求性递减的顺序列出:

- a) 调用:如果两个模块通过使用它们文档说明的函数调用进行严格通信,这两个模块就是调用耦合;调用耦合的实例是数据、标记和控制,定义如下。
 - 1) 数据:如果两个模块通过使用代表单一数据项的调用参数进行严格通信,这两个模块就是数据耦合。
 - 2) 标记:如果两个模块通过使用由多个域组成或者包含有意义的内部结构的调用参数进行严格通信,这两个模块就是标记耦合。
 - 3) 控制:如果一个模块传递意图影响另一个模块内部逻辑的信息,这两个模块就是控制耦合。
- b) 共有:如果两个模块共享公共数据区或公共系统资源,这两个模块就是共有耦合。全局变量表明使用全局变量的模块是公共耦合的。通过全局变量的公共耦合通常是被允许的,但要有限度。例如,只被一个模块使用的变量放在全局区域是不合适的,应该移除。在评估全局变量的适宜性中需要考虑的其他因素有:
 - 1) 修改一个全局变量的模块个数:一般来说,应该只分配一个模块负责控制全局变量的内容,但可以由第二个模块共同承担该职责,在这种情况下,必须提供充分的论证。多于两个模块共同承担该职责是不能接受的(在评估中,应该谨慎确定哪个模块对变量内容负实际职责,例如,如果变量由单一程序修改,但该程序仅仅执行其调用者请求的修改,这是调用模块的职责,可以有多个这样的模块)。此外,作为确定复杂性的一部分,如果两个模块共同负责一个全局变量的内容,应清楚地说明它们之间是如何协调修改的。
 - 2) 引用全局变量的模块个数:虽然一般来说对引用全局变量的模块个数没有限制,多个模块引用的情况中应检查引用的有效性和必要性。
- c) 内容:如果一个模块能够直接引用另一个模块内部,这两个模块就是内容耦合(例如修改其他模块的代码或者引用其他模块内部的标签)。结果是一个模块的部分或所有内容将有效的包

含在另一个模块中。可将内容耦合理解为使用了未公开模块接口,而在调用耦合中,仅仅使用了公开的模块接口。

A.3.2 程序软件的复杂度

复杂度是对代码执行的判断点和逻辑路径的度量。软件工程学把复杂度作为软件的消极特征,因为它阻碍对代码逻辑性和流程的理解。其他妨碍对代码理解的是不必要的代码,因为它是无用的或冗余的。

用于划分抽象级别以及尽量减少循环依赖的分层的使用,可以更好的理解 TSF,为 TOE 安全功能要求是在实现上准确、完整的实例化提供更多的保障。

降低复杂度还包括减少或消除相互依赖关系,这既适合单层模块,也适合那些分层的模块。相互依赖的模块可互相依靠,以形成单一的结果,这可能导致死锁条件,或更糟糕的是,竞争状态(例如,检查时间与使用关注的时间),其最终结论是不确定的并且受在特定时刻的计算环境的影响。

设计复杂度最小化是一个参考验证机制的关键特征,其目的是得出一个很容易理解的 TSF,这样可以完全分析它(参考验证机制还有其他重要特征,如 TSF 自我保护和不可旁路性,ADV_ARC 族中的要求覆盖了这些特征)。

A.4 ADV_TDS:子系统和模块

本节额外提供了 TDS 族和其术语“子系统”和“模块”使用的指导。接下来讨论了诸如更详细的描述是如何变得有用,不详细的要求是如何被降低的。

A.4.1 子系统

图 A.3 表明根据 TSF 的复杂度,设计可以用子系统和模块来描述(子系统比模块的抽象级别高);或者可以仅在一个抽象级别方面描述(例如低保障级别中的子系统、高级别中的模块)。如果给出了低级别的抽象(模块),在本质上也默认符合了对高级别抽象(子系统)的需求。下面在对子系统和模块的讨论中更详尽的阐述这个概念。

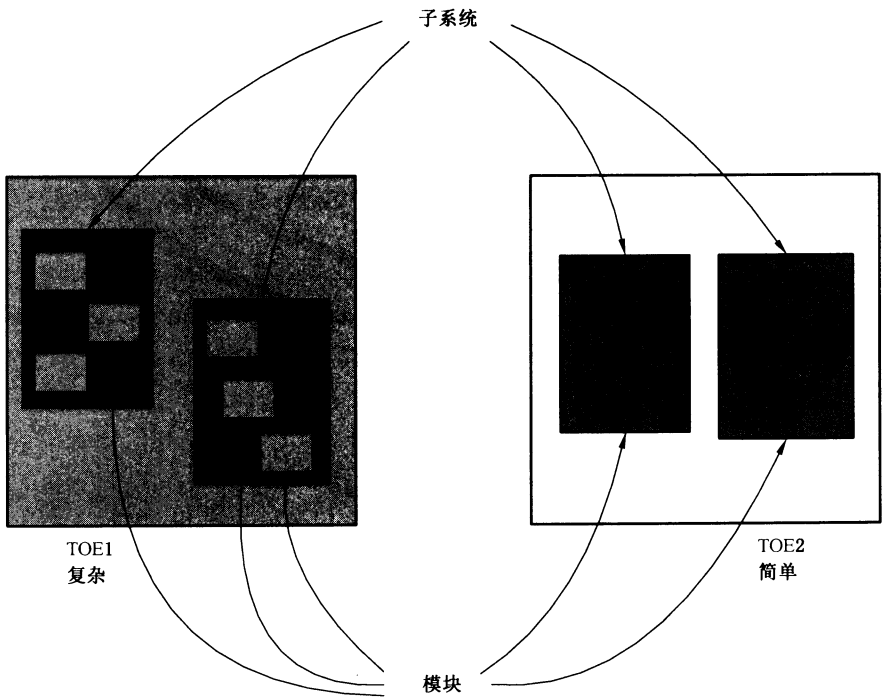


图 A.3 子系统和模块

期望开发者以子系统的方式描述 TOE 的设计。术语“子系统”用起来比较模糊,因此应将其对应于类似 TOE 的单元(例如,子系统、模块)。子系统甚至可以在范围上是不均匀的,只要描述子系统的要求得到满足。

子系统的第一个用途是区分 TSF 的边界,即组成 TOE 的 TSF 部分。一般情况下,如果子系统有能力(无论通过设计还是实现)影响任何安全功能要求的正确操作,则它是 TSF 的一部分。例如,对于依赖于不同的硬件执行模式的软件提供域分离(见 A.1),SFR-执行的代码是在一个域中执行,那么认为在该域中执行的所有子系统是 TSF 的一部分。同样,如果该域以外的一台服务器实现了一个安全功能要求(例如,对它管理的对象执行访问控制策略),那么它也将予以考虑为 TSF 的一部分。

子系统的第二个用途是当描述 TSF 如何工作时,它为在描述 TSF 的描写级别上提供的一种结构,没有必要包含应在模块描述中的低层实现细节(稍后讨论)。要么从高层描述(缺乏丰富的实现细节)要么从细节层面描述(提供对实现更多的洞察)子系统。对子系统的描述级别是由子系统对实施一个安全功能要求的负责程度确定的。

SFR-执行子系统,是一个提供了实现任意安全功能要求要素的机制子系统,或者直接支撑实现一个安全功能要求的子系统。如果子系统提供(实现)一个 SFR-执行 TSFI,则称为 SFR-执行子系统。

子系统也可确定为 SFR-支撑和 SFR-无关。SFR-支撑的子系统是 SFR-执行子系统为了实现一个安全功能要求所依赖的子系统,但不直接作为 SFR-执行子系统发挥作用。SFR-无关子系统是指不以支持的或实施的角色依赖于该子系统实现安全功能要求。

A.4.2 模块

模块通常是一个相对较小的结构单位,能够按照所讨论的 TSF 内部(ADV_INT)对其进行刻画。当 ADV_TDS.3 基础模块设计(或以上)的要求和 TSF 内部(ADV_INT)要求都出现在一个 PP 或 ST 中,在 TOE 设计(ADV_TDS)要求中的术语“模块”与 TSF 内部(ADV_INT)要求的“模块”是同一实体。不像子系统,模块很详细地描述了实现,详细程度可以作为评估实现表示的指导。

重要的是要注意,取决于 TOE、模块和子系统可以参照相同的抽象。对于 ADV_TDS.1 基础设计和 ADV_TDS.2 结构化设计(在模块级不需要描述),子系统描述提供有关 TSF 的细节的最低水平。对于 ADV_TDS.3 基础模块设计(需要模块的描述),这些描述提供最低级别的细节,而子系统描述(如果它们作为独立的实体存在)只是把模块描述置于上下文中。也就是说,如果模块描述存在,就没有必要提供详细的子系统描述。在十分简单的 TOE 中,单独的“子系统描述”是没有必要的,可通过由模块提供的文件满足要求。对于复杂的 TOE,子系统描述(关于 TSF 的)的目的是向读者提供上下文,这样他们能够适当地集中他们的分析。这种差异如图 A.3 所示。

SFR-执行模块是直接实现 ST 安全功能要求(SFR)的模块。这种模块可以实现一个 SFR-执行的 TSFI,但在安全功能要求中陈述的一些功能性(例如,审计和对象重用功能)可能无法直接依赖于单一的 TSFI。子系统是这种情况,SFR-支撑模块是 SFR-执行模块所依赖的模块,但不直接负责实现安全功能要求。SFR-无关模块是不直接或间接地处理 SFR-执行的那些模块。

这一点很重要,需要注意的是确定“直接实现”的意思带有一定的主观性。该术语最狭义上讲,它可以被解释为一或两行实际执行比较,归零操作等的代码,实现了一个要求。一个更宽泛的解释,可能是它包括在 SFR-执行 TSFI 的模块,以及该模块可能会依次调用的所有模块(依此类推,直至完成调用)。这些解释都不是特别令人满意,因为第一种解释的狭隘可能会导致被错误地归类为安全功能要求的支撑模块,而第二个信息模块实际上不是真正安全功能要求执行模块但却被归为此类。

模块的描述应该是这样,人们可以从该描述创建模块的实现,将导致实现 1) 根据给出的接口以及模块使用,应与实际 TSF 的实现相同,2) 与 TSF 模块一致。例如,RFC 793 提供了一个高层次的 TCP 协议的描述。它必然是独立实现的。虽然它提供了丰富的细节,但这**不是**一个合适的设计说明,因为它在实现方面不是具体的。在实际实现中,可在该 RFC 中添加指定的协议,执行选择(例如,在执行的

各个部分中全局数据与本地数据的使用)可能影响到所进行的分析。TCP 模块的设计说明会列出实现相关的接口(而不是那些在 RFC 793 中定义的),以及处理与实现 TCP 模块(假设它们是 TSF 的一部分)相关的算法说明。

在设计中,模块详细介绍了它们提供的功能(目的)、它们给出的接口、这些接口的返回值、它们的用途(其他模块提出),以及它们如何提供其功能的描述(这是一种描述算法的可能的途径)。

模块的目的应描述该模块提供什么功能。描述应足以使读者能够获得在架构上该模块的功能是什么的总体思路。

模块给出的接口是其他模块用来调用其功能的那些接口。接口包括显式接口(例如,其他模块调用的调用序列)以及隐式接口(例如,该模块操作的全局数据)。要描述接口如何被调用和返回的值。这种描述将包括参数列表,以及这些参数的描述。如果期望一个参数具有一组值(例如“flag”参数),该参数可能取的全部值将影响模块的处理也应被说明。同样,描述参数的数据结构,要标识和描述数据结构的每个字段。对全局数据的描述要提到它们是否可被模块进行读或写(或都可以)。

注意,不同的编程语言可能有额外不明显的“接口”,一个例子是在 C++ 中操作数或函数重载。在类描述中,这种“隐式接口”也将作为模块设计的一部分描述。注意,虽然模块可能只呈现一个接口,但更常见的是一个模块可呈现到一小组相关接口。

与此对应,某模块使用的接口必须被标识,这样它可以确定哪个模块正在被所描述的模块调用。从设计描述中必须也应清晰看出正在被调用模块的算法前提。例如,如果描述模块 A,并且它调用了模块 B 的冒泡排序程序,一个不充分的算法描述可能是这样的“模块 A 调用模块 B 中的 *double_bubble()* 接口去执行一个冒泡排序”。一个充分的算法描述应是“模块 A 用访问控制实体列表调用 *double_bubble* 程序, *double_bubble()* 将返回用户名首字母排序后的实体,接着按如下规则返回允许访问的域……”。设计中对一个模块的详细描述必须提供足够的细节,使其清晰看出作用于模块 A 的是冒泡排序接口。注意,展现这种调用接口的一种方法是调用树,这样算法描述就可被包含进被调用模块的算法描述中。

如前所述,模块的算法描述应说明模块的算法实现方式。这能够通过伪代码、流程图或非正式文本(ADV_TDS.3 基础模块设计)做到。它讨论了如何用模块的输入和调用的函数来完成模块的功能。它注意到全局数据、系统状态以及模块产生的返回值的变化的。它达到这样的详细程度,得出的实现非常类似于 TOE 的真正实现。

应该注意的是源代码不符合模块文档的要求。虽然模块设计描述了实现,但这不是真正的实现。如果他们提供了源代码含义的解释,源代码周围的注释可能是足够的文档说明。只是说明每一行代码是做什么的行注释是无用的,因为它们没有解释该模块旨在完成什么。

在下面的元素中,对子系统和模块的分类命名方法(SFR-执行、SFR-支撑和 SFR-无关)用来描述信息的数量和类型,需要由开发者提供。元素被结构化,因此不要期望开发者仅提供指定的信息。也就是说,如果开发者的 TSF 文档在下面的要求中提供了信息,不期望开发者更新他们的文件以及将子系统和模块分类为 SFR-执行、SFR-支撑和 SFR-无关。这个标签的主要目的是允许用不太成熟开发方法的开发者(以及相关的构件,如详细的接口和设计文档)在不消耗必要成本的情况下提供必要的证据。

A.4.3 分级方法

由于确定什么是 SFR-执行与 SFR-支撑(以及在某些情况下,甚至确定什么是 SFR-无关)带有主观性,下面的范型已在本族中采用。在本族的早期组件中,开发者提供适当的信息确定子系统分类到 SFR-执行等,并且很少有额外的证据提供给评估者用于考查对于这种说法的支持。当所需的保障级别增加,随着开发者仍然确定分类,评估者得到越来越多的证据是用来确认开发者的分类。

为了把评估者的分析重点放在 TOE 的安全功能要求相关部分,尤其是在较低保障级别中,族的组件被分级,这样,最初的详细信息只用来描述 SFR-执行的架构实体。当保障级别增加,需要为 SFR-支撑及(最终)SFR-无关的实体提供更多的信息。应该注意,即使是需要完整的信息,它不要求所有这些

信息在同一详细级别上分析。在所有情况下重点应放在是否已提供和分析了必要的信息。

表 A.1 总结出本族每个组件描述的结构实体所需的信息。

表 A.1 分级详细描述

	TSF 子系统			TSF 模块		
	SFR-执行	SFR-支撑	SFR-无关	SFR-执行	SFR-支撑	SFR-无关
ADV_TDS.1 基础设计（非形式化 陈述）	结构、SFR-执行 行为概述、相互 作用	指定支撑 ^a	指定支撑			
ADV_TDS.2 结构化设计（非形式化 陈述）	结构、SFR-执行 行为的详细描述、其他行为的 概述、相互作用	结构、其他 行为的概 述、相互 作用	指定支撑、 相互作用			
ADV_TDS.3 基础模块设计（非形式 化陈述）	描述、相互作用	描述、相互 作用	描述、相互 作用	目的、安全功 能要求接口 ^b	相互作用、 目的	相互作用、 目的
ADV_TDS.4 半形式化模块设计（半 形式化陈述）	描述、相互作用	描述、相互 作用	描述、相互 作用	目的、安全功 能要求接口	目的、安全功 能要求接口	相互作用、 目的
ADV_TDS.5 完全半形式化模块设计 （半形式化陈述）	描述、相互作用	描述、相互 作用	描述、相互 作用	目的、所有 接口 ^c	目的、所有 接口	目的、所有 接口
ADV_TDS.6 带形式化高层设计表示 的完全半形式化模块设 计（半形式化陈述、附加 的形式化陈述）	描述、相互作用	描述、相互 作用	描述、相互 作用	目的、所有 接口	目的、所有 接口	目的、所有 接口
^a 指定支撑意味着只需要提供的文档足够支撑子系统或模块的分类。 ^b 安全功能要求接口意味着模块描述包含每个安全功能要求相关接口的返回值和其他模块调用的接口。 ^c 所有接口意味着模块描述包含每个接口的返回值以及其他模块调用的接口。						

A.5 形式化方法补充材料

形式化方法提供 TSF 及其行为的数学表示法，ADV_FSP.6 附加形式化描述的完备的半形式化功能规范、ADV_SPM.1 形式化 TOE 安全策略模型和 ADV_TDS.6 带形式化高层设计表示的完全半形式化模块设计都要求形式化方法。形式化方法有两个方面：用于形式化表示的规范语言，以及算术证明形式化规范完整性和正确性的理论证据。

形式化规范是基于完善的数学概念表达的。这些数学概念用于定义良好的语义、语法和推理规则。形式化系统是一种身份和关系的抽象系统，能够通过指定形式化字母表，在基于形式化语法的字母之上的形式化语言，以及用形式化语言中的句子建立的一套形式化的推理规则描述。

评估者应检查识别的形式化系统，以确保：

- 定义了形式化系统的语义、语法和推理规则或定义被引用。
- 每个形式化系统伴随着解释性的文字,提供了定义的**语义**,以确保:
 - 1) 解释性文字提供被正规使用公认之外的上下文中使用的术语、缩写和首字母缩写的已定义的含义;
 - 2) 使用形式化方法和半形式化符号要伴随着非正式风格的支持性解释性文字,用于明确其含义;
 - 3) 形式化系统能够表达适用的 SFP 的规则和特点、TSF 安全功能和接口(提供作用、异常和错误消息的详细信息)、本保障族指定的使用了符号的子系统或模块。
 - 4) 符号提供规则来确定有效语法结构的意义。
- 每一个形式化系统采用一个形式化语法,提供了用以明确地识别结构的规则。
- 每一个形式化系统提供证据规则
 - 5) 支持完善的数学概念的逻辑推理;
 - 6) 有助于防止矛盾的产生。

如果开发人员使用了评估机构已经接受的形式化系统,评估者能够依靠系统的形式化和强度水平,并关注于 TOE 规范和相应证据相关的形式化系统实例化。

形式化风格支持基于安全特征、改进的一致性和陈述的对应关系安全属性的数学证明。形式化工具支持似乎应该足够,否则人工推导变得长篇大论、无法理解。形式化工具也易于降低人工推导固有的错误概率。

形式化系统的例子:

- **Z 规范语言**极富表现力,并支持许多不同的方法或风格的形式化规范。已使用的 Z 主要面向模型的规范,使用图解来形式化指定操作。更多信息参见 <http://vl.zuser.org/>。
- **ACL2** 是一个开放源码的形式化系统,包括基于 Lisp 的规范语言和定理证明。更多信息参见 <http://www.cs.utexas.edu/users/moore/acl2/>。
- **Isabelle** 是一种流行的通用定理,证明环境允许用形式化语言表达数学公式,并提供工具用来证明在逻辑演算中的这些公式(其他信息参见,例如 <http://www.cl.cam.ac.uk/Research/HVG/Isabelle/>)。
- **B 方法**是一个基于命题演算、带推理规则和理论集合的一阶谓词演算的形式化系统(更多信息参见,例如 <http://vl.fmnet.info/b/>)。

附 录 B
(资料性附录)
组合(ACO)

本附录的目的是解释组合评估和 ACO 标准的一些概念。本附录没有定义 ASE 标准,在第 10 章能找到有关的定义。

B.1 组合 TOE 评估的必要性

整个 IT 市场是由许许多多特定种类的产品/技术提供商组成的。尽管有些交叉,PC 硬件提供商可能也提供应用软件和/或操作系统,或者芯片厂商可能也开发用于他们自己芯片组的专用操作系统,但通常情况是,一个 IT 解决方案是由很多不同类型的提供商共同实现的。

有时除了需要提供单个部件的保障之外,还需要提供组合部件的保障。尽管这些提供商之间有合作,但是部件的技术整合所需的特定材料的分发,协议则很少涉及提供的设计信息和开发过程/程序证据的详细程度。从被另一个部件依赖的部件开发者得到的信息不足,意味着依赖部件的开发者不能访问对依赖部件和基本部件执行 EAL2 或更高级别的评估所需的信息。因此,虽然仍能执行依赖部件的任意保障级别的评估,但对组合部件执行 EAL2 保障级或更高级的评估,需要重用对部件开发者执行各种评估的评估证据和结果。

ACO 标准的目标是用于一个 IT 实体依赖另一个实体提供安全服务的情形。提供服务的实体称为“基本部件”,接受服务的实体称为“依赖部件”。这种关系在很多情况下存在。例如,一个应用程序(依赖部件)可能使用一个操作系统(基本部件)提供的服务。可选地,两个关联的应用程序的情形,要么在通用操作系统环境下,要么运行在单独的硬件平台上,这种关系可能是对等的。如果存在一个占支配地位的实体向辅助实体提供服务,那么这个占支配地位的实体被认为是基本部件而辅助实体认为是依赖部件。如果对等体以相互的方式为彼此提供服务,那么,每个实体被认为既是提供服务的基本部件,也是得到服务的依赖部件。这将要求对每种部件实体重复运用 ACO 组件的所有要求。

另一个目标是更加广泛的适用性,逐步地(组合 TOE 由一个依赖部件和一个基本部件组成,而这个基本部件又是另一个组合 TOE 的依赖部件)适用于更加复杂的关系,但这或许需要进一步的阐述。

组合 TOE 评估仍然要求各个部件进行了独立的评估,因为组合评估以各个部件的评估结果为基础。或许当组合 TOE 开始评估时,依赖部件的评估仍在进行中。但是,依赖部件的评估必须在组合 TOE 评估完成之前完成。

组合评估活动可能与依赖部件评估同时进行。这将归因于以下两个因素:

- a) 经济/业务驱动——依赖部件开发者要么是组合评估活动的发起者,要么支持这些活动,因为从依赖部件评估得到的可交付的评估证据或结果是组合评估活动必需的。
- b) 技术驱动——随着逐步理解依赖部件最近经历(正在经历)部件评估,并且所有与评估有关的可交付的评估证据或结果都是可用的,组件要考虑所必需的保障级是否由基本部件提供(例如,考虑基本部件自部件评估完成以来的变化)。因此,没有要求组合期间重新验证依赖部件的评估活动。同时,基本部件组成的(其中一个)用作依赖部件评估期间的依赖部件测试的测试配置项,由 ACO_CTT 去考虑这个配置项中的基本部件。

要求依赖部件评估得到的评估证据作为组合 TOE 评估活动的输入。要求基本部件评估得到的唯一的评估资料作为组合 TOE 评估活动的输入,该资料是:

- a) 基本部件评估期间报告的基本部件的残余脆弱点。这是 ACO_VUL 评估活动要求的。

组合 TOE 评估不要求其他从基本部件评估活动得到的评估证据,因为基本部件组合评估的结果

将被重用。如果组合 TOE TSF 包括的要比基本部件的部件评估期间 TSF 要多,那么可能要求基本部件的其他信息。

基本部件和依赖部件的部件评估假定在为 ACO 组件确定最终结论的时候已经完成了。

ACO_VUL 组件仅考虑抵御具有增强型基本攻击潜力的攻击者。这将根据提供的关于基本部件是如何提供依赖部件依赖的服务的设计信息的级别,运用 ACO_DEV 活动。因此,尽管构成组合 TOE 的部件的保障级可能高于 EAL4 级,但采用 CAP 对组合 TOE 进行评估所能获得的信心与对部件 TOE 进行 EAL4 评估所能获得的信心相当。

B.2 执行组合 TOE 的安全目标评估

开发者将会提供一份 ST 文档,用于组合 TOE(基本部件+依赖部件)的评估。ST 会标识用于组合 TOE 的保障包,该保障包利用部件评估获得的保障为复合实体提供保障。

在 ST 中考虑部件组合的目的是从环境和要求的角度验证部件间的兼容性,同时也评定组合 TOE ST 与部件 ST 以及 ST 描述的安全策略是一致的。这包括要确定部件 ST 与其描述的安全策略是兼容的。

组合 TOE ST 可能涉及部件 ST 的内容,或 ST 作者可能在组合 TOE ST 中选择性的重述部件 ST 的内容,并提供如何在组合 TOE ST 中描述部件 ST 的基本原理。

在为组合 TOE ST 执行 ASE_CCL 评估活动期间,评估者要确定在组合 TOE ST 中部件 ST 的描述是准确的。这将通过确定组合 TOE ST 与部件 TOE ST 的一致性得到了论证获得。同时,评估者也需要确定依赖部件运行环境的依赖条件在组合 TOE 中得到了充分的满足。

组合 TOE 描述将会描述组合方案。描述组合方案的逻辑上和物理上的范围和边界,也会标识部件间的逻辑边界。描述中会标识每个部件提供的安全功能。

组合 TOE 安全功能要求的陈述会标识满足每个安全功能要求的部件。如果某个安全功能要求是由两个部件满足的,那么陈述会标识各个部件满足安全功能要求的不同方面。同样的,组合 TOE 概要规范会标识各个部件提供的上述安全功能。

应用于组合 TOE ST 的安全目标评估要求(ASE 包)应与部件评估中使用的安全目标评估要求(ASE 包)一致。

部件 ST 评估的评估结果重用的一个例子是组合 TOE ST 直接引用部件 ST。例如,如果组合 TOE ST 引用部件 ST,作为其安全功能要求陈述的一部分,评估者能理解完成所有赋值和选择的操作(ASE_REQ 中陈述的)在部件评估中已经满足了。

B.3 组合 IT 实体间的相互作用

基本部件的 TSF 经常在缺少对潜在应用的依赖性的知识的情况下被定义,有了这些知识基本部件才能被组合。基本部件的 TSF 被定义成包括执行基本部件安全功能要求依赖的所有部分,这也将包括实现基本部件安全功能要求必需的所有部分。

基本部件的 TSFI 表示 TSF 为安全功能要求陈述中定义的外部实体提供调用 TSF 服务的接口。包括用户接口和外部 IT 实体的接口。但是,TSFI 仅包括 TSF 的接口,因此,不需要外部实体和基本部件之间所有可用接口的详尽说明。基本部件可能提供被认为是与安全无关的服务接口,或者因为这种服务的固有用途(如,调整字体),又或者因为与 ISO/IEC 15408 相关的安全功能要求在基本部件的 ST 中没有声明(如,当在 ISO/IEC 15408-2 的 FIA:标识和鉴别安全功能要求中没有声明登录接口时)。

基本部件提供的功能接口是除安全接口(TSFI)之外的接口,在基本部件评估期间是不要求考虑的。通常包括依赖部件调用基本部件提供的服务使用到的接口。

基本部件可能包括一些间接调用 TSFI 的接口,例如,用于调用 TSF 服务的 API,这些在基本部件评估期间是不考虑的。

图 B.1 表示了基本部件和接口的抽象。

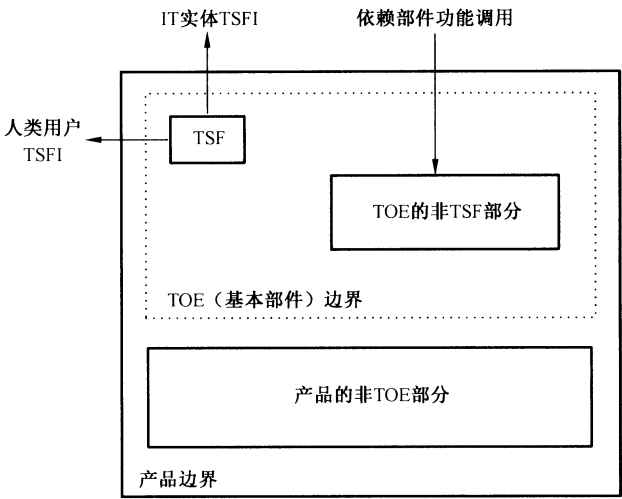


图 B.1 基本部件抽象

依赖于基本部件的依赖部件类似于以下定义:部件 ST 安全功能要求中定义的外部实体接口被划分为 TSFI 类,并且在 ADV_FSP 中检查。

任何依赖 TSF 要求环境对某一安全功能要求提供支持,应表明为了确保所陈述的依赖部件安全功能要求的执行,依赖 TSF 要求从环境获得某种服务。这种服务是在依赖部件边界之外的,基本部件也不可能在依赖部件 ST 中作为外部实体被定义,因此,依赖 TSF 调用低层平台(基本部件)的服务,不会作为功能规范(ADV_FSP)评估活动的一部分被分析。这种对基本部件的依赖将作为环境安全目的在依赖部件 ST 中描述。

图 B.2 表示了依赖部件和接口的抽象。

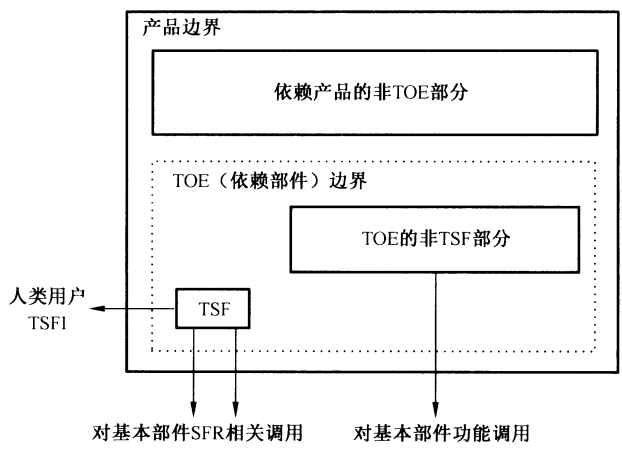


图 B.2 依赖部件抽象

当考虑基本部件和依赖部件的组合,如果依赖部件的 TSF 要求从基本部件获得服务以支持安全功能要求的实施,那么需要定义服务接口。如果该服务是由基本部件 TSF 提供的,那么接口应是基本部件的 TSFI,因此,会在基本部件的功能规范中定义。

但是,如果依赖部件 TSF 调用的服务不是基本部件 TSF 提供的(即它是在基本部件的非 TSF 部分或者甚至是基本部件的非 TOE 部分实现的(图 B.3 中没有实例)),除非该服务是基本部件 TSF 间接

产生,否则将不存在与该服务相关的基本部件 TSFI。依赖部件服务接口到运行环境都在“依赖部件的依赖性”族(ACO_REL)中考虑。

组合 TOE TSF 中涉及基本部件的非 TSF 部分,是由于依赖部件依靠基本部件支持依赖部件的安全功能要求的这种依赖性。因此,这种情况下,组合 TOE 的 TSF 会大于各部件 TSF 之和。

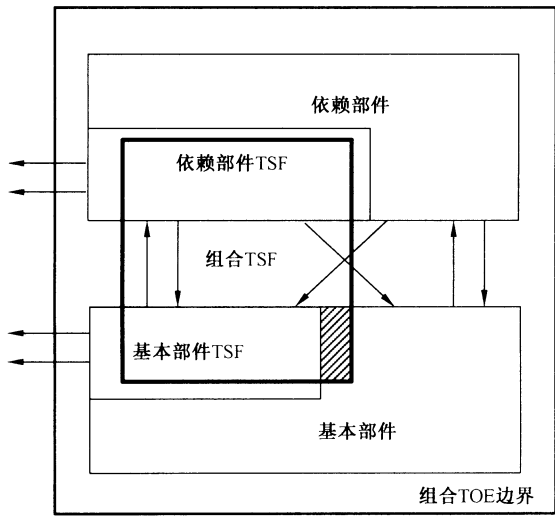


图 B.3 组合 TOE 抽象

在基本部件评估过程中,可能会出现基本部件 TSFI 以不可预料的方式被调用的情况。因此,要求对基本部件 TSFI 进行进一步的测试。

图 B.4 以及支持性文本将进一步描述可能的接口。

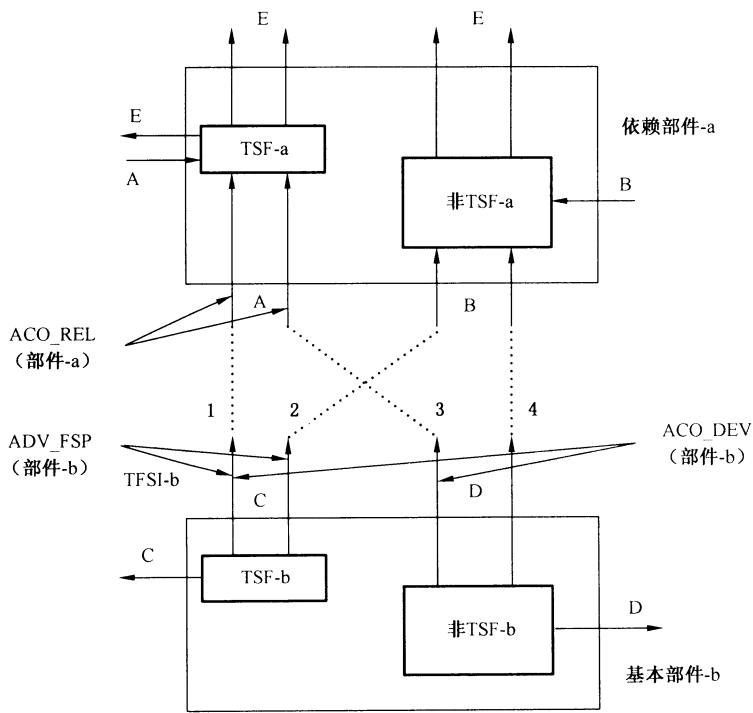


图 B.4 组合 TOE 的接口

- a) 进入依赖部件:a 的箭头(A 和 B)表示部件期望环境对服务请求作出回应(依赖部件对环境要求的回应)；

- b) 从基本部件:b 出来的箭头(C 和 D)表示基本部件向环境提供的服务接口;
- c) 部件之间的虚线表示两个接口之间的交互;
- d) 其他的(灰色)箭头表示 ISO/IEC 15408 描述的接口。

下面是一个简要说明,解释了需要考虑的事项。

图中有部件-a(依赖部件-a)和部件-b(基本部件):TSF-a 出来的箭头是 TSF-a 提供的服务,因此称为 TSFI(a);类似的,TSF-b(“C”)出来的箭头称为 TSFI(b)。在它们各自的功能规范中有详细描述。部件-a 向它的环境请求服务:TSF(a)需要的服务标记为“A”,其他的(与 TSF-a 无关的)服务标记为“B”。

当部件-a 和部件-b 组合起来,{部件-a 需要的服务}和{部件-b 提供的服务}会有 4 种可能的组合形式,如图中虚线表示(接口对之间的交互)。任何一种都可能存在于某种特定的组合中。

- a) TSF-a 需要 TSF-b 提供的服务(“A”连接“C”):这是非常直接的,“C”的细节在部件-b 的功能规范中。这种情况的所有接口应在部件-b 的功能规范中定义。
- b) 非 TSF-a 需要 TSF-b 提供的服务(“B”连接“C”):这是非常直接的(“C”的细节也在部件-b 的功能规范中),但不是重要的:安全方式。
- c) 非 TSF-a 需要非 TSF-b 提供的服务(“B”连接“D”):没有关于“D”的细节,但这些接口的使用不涉及安全相关的内容,因此在评估中不需要考虑,尽管对于开发者来说是一个综合性的问题。
- d) TSF-a 需要非 TSF-b 提供的服务(“A”连接“D”):当部件-a 和部件-b 对“安全服务”有不同的理解时,这将会出现。或许部件-b 没有关于 I&A 的声明(在 ST 中没有 FIA 安全功能要求),但是部件-a 需要环境提供鉴权。关于“D”接口没有可用的详细资料(它们不是 TSFI(b),因此它们不在部件-b 的功能规范中)。

注意:如果上述 d 描述的这种交互存在,那么组合 TOE 的 TSF 将会是 TSF-a+TSF-b+非 TSF-b。否则,组合 TOE 的 TSF 将会是 TSF-a+TSF-b。

图 B.4 中的第 2 和第 4 种接口与组合 TOE 评估不直接相关。接口 1 和接口 3 将会在不同族的实施期间考虑:

- a) 功能规范(ADV_FSP)(部件-b)描述 C 接口。
- b) 依赖部件的依赖性(ACO_REL)描述 A 接口。
- c) 开发证据(ACO_DEV)描述第一种连接中的 C 接口和第三种连接中的 D 接口。

可能运用组合的一个典型的例子是数据库管理系统(DBMS)依赖它低层的操作系统(OS)。在 DBMS 部件评估期间,将会对 DBMS 的安全属性做一个评定(在评估中运用的保障组件要指定到何种严格程度):标识 TSF 边界,评估功能规范,以确定其是否描述了 TSF 提供的安全服务接口,也许会提供 TSF 的一些额外信息(其设计、架构、内部组织),测试 TSF,评估生命周期的各个阶段和指导性文档,等等。

但是,DBMS 评估不要求 DBMS 依赖 OS 的依赖性相关的任何证据。DBMS 的 ST 最可能在它的假设部分陈述关于 OS 的假设,在它的环境部分,陈述 OS 的安全目的。DBMS ST 甚至可能根据 OS 的安全功能要求呈现环境目的。但是,没有 OS 规范与功能规范、架构描述或者关于 DBMS 其他的 ADV 证据中的细节映射。依赖部件的依赖性(ACO_REL)能满足这些需求。

依赖部件的依赖性(ACO_REL)描述依赖 TOE 调用基本部件提供服务的接口。这些是对请求做出响应的接口。该接口描述是从依赖部件的角度提供的。

开发证据(ACO_DEV)描述基本部件提供的用于响应依赖部件服务请求的接口。这些接口映射到依赖信息中标识的相关依赖部件接口。(映射的完整性,即描述的基本部件接口是否对应了所有的依赖部件接口,在这里没有验证,而是在组合基本原理中验证(ACO_COR))。子系统提供的接口在高级别的 ACO_DEV 组件中描述。

依赖部件要求的所有没有在基本部件中描述的接口,在组合基本原理族(ACO_COR)的基本原理中介绍。基本原理也负责陈述依赖部件依赖的基本部件接口是否已在基本部件评估中考虑。对于任何在基本部件评估中没有考虑的接口,将会提供一份关于使用这些接口对基本部件 TSF 影响的基本原理文档。

附 录 C
(资料性附录)

保障组件依赖关系的交叉引用

在第 9 章～第 16 章的组件中列出的依赖关系是保障组件之间的直接依赖性。

以下的保障组件依赖关系表列出了功能组件之间直接、间接和可选的依赖关系。作为功能组件依赖方的每个组件在表中占据一列,每个功能组件在表中占据一行。表格单元中的值表示行中标的组件直接要求列中标的组件(用“×”表示)或间接要求列中标的组件(用“-”表示)。如果表格单元为空,则该组件不依赖于另一个组件。

表 C.1～表 C.8 给出了各保障类组件之间的依赖关系。

表 C.1 ACO:组合类的依赖关系表

	ACO_DEV.1	ACO_DEV.2	ACO_DEV.3	ACO_REL.1	ACO_REL.2	ALC_CMC.1	ALC_CMS.1
ACO_COR.1	×			×		×	—
ACO_CTT.1	×			×			
ACO_CTT.2		×		—	×		
ACO_DEV.1				×			
ACO_DEV.2				×			
ACO_DEV.3					×		
ACO_REL.1							
ACO_REL.2							
ACO_VUL.1	×			—			
ACO_VUL.2		×		—			
ACO_VUL.3			×		—		

表 C.2 ADV:开发类的依赖关系表

	ADV_FSP.1	ADV_FSP.2	ADV_FSP.3	ADV_FSP.4	ADV_FSP.5	ADV_FSP.6	ADV_IMP.1	ADV_TDS.1	ADV_TDS.3	ADV_CMC.5	ADV_CMS.1	ALC_DVS.2	ALC_LCD.1	ALC_TAT.1
ADV_ARC.1	×	—						×						
ADV_FSP.1														
ADV_FSP.2		—						×						
ADV_FSP.3		—						×						
ADV_FSP.4		—						×						

表 C.2 (续)

	ADV_FSP,1	ADV_FSP,2	ADV_FSP,3	ADV_FSP,4	ADV_FSP,5	ADV_FSP,6	ADV_IMP,1	ADV_TDS,1	ADV_TDS,3	ADV_CMC,5	ADV_CMS,1	ALC_DVS,2	ALC_LCD,1	ALC_TAT,1
ADV_FSP.5		—		—			×	×	—					—
ADV_FSP.6		—		—			×	×	—					—
ADV_IMP.1		—		—			—	—	×					×
ADV_IMP.2		—		—			—	—	×	×	—	—		×
ADV_INT.1		—		—			×	—	×					×
ADV_INT.2		—		—			×	—	×					×
ADV_INT.3		—		—			×	—	×					×
ADV_SPM.1		—		×				—						
ADV_TDS.1		×						—						
ADV_TDS.2		—	×					—						
ADV_TDS.3		—		×				—						
ADV_TDS.4		—		—	×		—	—	—					—
ADV_TDS.5		—		—	×		—	—	—					—
ADV_TDS.6		—		—		×	—	—	—					—

表 C.3 AGD:指导性文档类的依赖关系表

	ADV_FSP,1
AGD_OPE.1	×
AGD_PRE.1	

表 C.4 ALC:生命周期支持类的依赖关系表

	ADV_FSP,2	ADV_FSP,4	ADV_IMP,1	ADV_TDS,1	ADV_TDS,3	ALC_CMS,1	ALC_DVS,1	ALC_DVS,2	ALC_LCD,1	ALC_TAT,1
ALC_CMC.1						×				
ALC_CMC.2						×				
ALC_CMC.3						×	×		×	

表 C.4 (续)

	ADV _ FSP, 2	ADV _ FSP, 4	ADV _ IMP, 1	ADV _ TDS, 1	ADV _ TDS, 3	ALC _ CMS, 1	ALC _ DVS, 1	ALC _ DVS, 2	ALC _ LCD, 1	ALC _ TAT, 1
ALC_CMC,4						×	×		×	
ALC_CMC,5						×		×	×	
ALC_CMS,1										
ALC_CMS,2										
ALC_CMS,3										
ALC_CMS,4										
ALC_CMS,5										
ALC_DEL,1										
ALC_DVS,1										
ALC_DVS,2										
ALC_FLR,1										
ALC_FLR,2										
ALC_FLR,3										
ALC_LCD,1										
ALC_LCD,2										
ALC_TAT,1	—	—	×	—	—					—
ALC_TAT,2	—	—	×	—	—					—
ALC_TAT,3	—	—	×	—	—					—

表 C.5 APE:保护轮廓评估类的依赖关系表

	APE _ ECD, 1	APE _ INT, 1	APE _ OBJ, 2	APE _ REQ, 1	APE _ SPD, 1
APE_CCL,1	×	×		×	
APE_ECD,1					
APE_INT,1					
APE_OBJ,1					
APE_OBJ,2					×
APE_REQ,1	×				
APE_REQ,2	×		×		—
APE_SPD,1					

表 C.6 ASE:安全目标评估类的依赖关系表

	ASE_SPD.1	ASE_REQ.1	ASE_OBJ.2	ASE_INT.1	ASE_ECD.1	ADV_TDS.1	ADV_FSP.2	ADV_FSP.1	ADV_ARC.1	
ASE_CCL.1		×		×	×					
ASE_ECD.1										
ASE_INT.1										
ASE_OBJ.1										
ASE_OBJ.2	×									
ASE_REQ.1					×					
ASE_REQ.2			×		×					
ASE_SPD.1										
ASE_TSS.1		×		×	—			×		
ASE_TSS.2		×		×	—	—	—	—	×	

表 C.7 ATE:测试类的依赖关系表

	ATE_FUN.1	ATE_COV.1	ALC_TAT.1	AGD_PRE.1	AGD_OP3.1	ADV_TDS.4	ADV_TDS.3	ADV_TDS.2	ADV_TDS.1	ADV_IMP.1	ADV_FSP.5	ADV_FSP.4	ADV_FSP.3	ADV_FSP.2	ADV_FSP.1	ADV_ARC.1	
ATE_COV.1	×								—					×			
ATE_COV.2	×								—					×			
ATE_COV.3	×								—					×			
ATE_DPT.1	×							×	—				—	—	—	×	
ATE_DPT.2	×						×		—			—		—	—	×	
ATE_DPT.3	×		—			×	—		—		—	—		—	—	×	
ATE_DPT.4	×		—			×	—		—	×	—	—		—	—	×	
ATE_FUN.1	—	×							—					—			
ATE_FUN.2	—	×							—					—			
ATE_IND.1				×	×									×			
ATE_IND.2	×	×		×	×				—					×	—		
ATE_IND.3	×	×		×	×				—			×		—	—		

表 C.8 AVA:脆弱性评定类的依赖关系表

	ADV _ ARC, 1	ADV _ FSP, 1	ADV _ FSP, 2	ADV _ FSP, 4	ADV _ IMP, 1	ADV _ TDS, 1	ADV _ TDS, 3	AGD _ OPE, 1	AGD _ PRE, 1	ALC _ TAT, 1
AVA_VAN.1		×						×	×	
AVA_VAN.2	×	×	—			×		×	×	
AVA_VAN.3	×	—	×	—	×	—	×	×	×	—
AVA_VAN.4	×	—	×	—	×	—	×	×	×	—
AVA_VAN.5	×	—	×	—	×	—	×	×	×	—

附 录 D
(资料性附录)
PP 和保障组件的交叉引用

表 D.1 给出了 PP 与 APE 类的族、组件之间的关系。

表 D.1 PP 保障级汇总

保障类	保障族	保障组件	
		低保障要求 PP	PP
保护轮廓评估	APE_CCL	1	1
	APE_ECD	1	1
	APE_INT	1	1
	APE_OBJ	1	2
	APE_REQ	1	2
	APE_SPD		1

附 录 E
(资料性附录)

EAL 和保障组件的交叉引用

表 E.1 给出了评估保障级与评估类、族和组件之间的关系。

表 E.1 评估保障级汇总

保障类	保障族	评估保障级(EAL)依据的保障组件						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
开发	ADV_ARC		1	1	1	1	1	1
	ADV_FSP	1	2	3	4	5	5	6
	ADV_IMP				1	1	2	2
	ADV_INT					2	3	3
	ADV_SPM						1	1
	ADV_TDS		1	2	3	4	5	6
指导性文档	AGD_OPE	1	1	1	1	1	1	1
	AGD_PRE	1	1	1	1	1	1	1
生命周期支持	ALC_CMC	1	2	3	4	4	5	5
	ALC_CMS	1	2	3	4	5	5	5
	ALC_DEL		1	1	1	1	1	1
	ALC_DVS			1	1	1	2	2
	ALC_FLR							
	ALC_LCD			1	1	1	1	2
	ALC_TAT				1	2	3	3
安全目标评估	ASE_CCL	1	1	1	1	1	1	1
	ASE_ECD	1	1	1	1	1	1	1
	ASE_INT	1	1	1	1	1	1	1
	ASE_OBJ	1	2	2	2	2	2	2
	ASE_REQ	1	2	2	2	2	2	2
	ASE_SPD		1	1	1	1	1	1
	ASE_TSS	1	1	1	1	1	1	1
测试	ATE_COV		1	2	2	2	3	3
	ATE_DPT			1	2	3	3	4
	ATE_FUN		1	1	1	1	2	2
	ATE_IND	1	2	2	2	2	2	3
脆弱性评定	AVA_VAN	1	2	2	3	4	5	5

附 录 F
(资料性附录)

CAP 和保障组件的交叉引用

表 F.1 给出了组合保障级与评估类、族和组件之间的关系。

表 F.1 评估保障级汇总

保障类	保障族	组合保障包(CAP)依据的保障组件		
		CAP-A	CAP-B	CAF-C
组合	ACO_COR	1	1	2
	ACO_CTT	1	2	2
	ACO_DEV	1	2	3
	ACO_REL	1	1	2
	ACO_VUL	1	2	3
指导性文档	AGD_OPE	1	1	2
	AGD_PRE	1	1	2
生命周期支持	ALC_CMC	1	1	2
	ALC_CMS	2	2	2
	ALC_DEL			
	ALC_DVS			
	ALC_FLR			
	ALC_LCD			
	ALC_TAT			
安全目标评估	ASE_CCL	1	1	2
	ASE_ECD	1	1	2
	ASE_INT	1	1	2
	ASE_OBJ	1	2	2
	ASE_REQ	1	2	2
	ASE_SPD		1	2
	ASE_TSS	1	1	2

中 华 人 民 共 和 国
国 家 标 准
信息技术 安全技术
信息技术安全评估准则
第 3 部分:安全保障组件

GB/T 18336.3—2015/ISO/IEC 15408-3:2008

*

中国标准出版社出版发行
北京市朝阳区和平里西街甲 2 号(100029)
北京市西城区三里河北街 16 号(100045)

网址 www.spc.net.cn

总编室:(010)68533533 发行中心:(010)51780238

读者服务部:(010)68523945

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

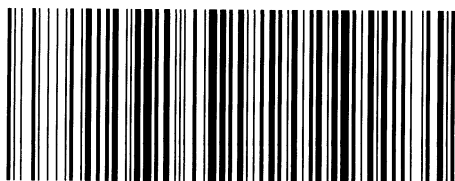
*

开本 880×1230 1/16 印张 10.25 字数 295 千字
2015 年 5 月第一版 2015 年 5 月第一次印刷

*

书号: 155066 • 1-51488 定价 103.00 元

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话:(010)68510107



GB/T 18336.3-2015