DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDICES

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX A Rationale for the Criteria, Services and Levels of Service

Introduction

The CTCPEC is an attempt to create disjoint useful categorizations of security mechanisms in terms of the service provided and its strength. This appendix attempts to detail some of the decisions made in the selection of the services and the levels within the service.

One of the primary goals was to create each service level such that the following goals were met:

1. Each service level should offer an *identifiable increase in protection*. It was not a requirement that the levels be strictly hierarchical other than in level of protection; and
2. Each service level should be *non-ambiguously and quantitatively different* from those above and below it so that products can be evaluated more than once and all will come to the same conclusions as to the levels of service achieved.
3. *Business issues* of evaluation such as sufficiency of a given security policy or covert channel bandwidths should not be a part of the criteria.

Security Policy

Each service level requires that a security policy be developed and stated by the vendor and enforced by the product. The security policy needs to define which of the products objects the service applies to. Some services explicitly state that the service must apply to all objects; otherwise it is expected that the vendor will define a useful subset of all objects. This defined subset of the products objects which are protected by a given policy are known as *protected objects* with respect to that policy.

All service security policies must be approved by the evaluation authority. The evaluation authority reserves the right to reject a product or a service security policy on the basis of the set of objects protected or the mechanism used to implement the security policy.

Location of Services

Although the CTCPEC places each service within a specific set of *Criteria,* the locations of a given service do not indicate nor imply that the service is specific to that particular *Criteria.* For example, the Object Reuse service is located within the Confidentiality Criteria. This placement is historical. Object Reuse is a service taken from the TCSEC; as such, it was closely associated with confidentiality. The CTCPEC, in keeping with the historical precedent, includes Object Reuse in the Confidentiality Criteria.

The placement of any given service within a *Criteria* was dependant upon historical precedent as well as most logical association. Thus, both Object Reuse and Covert Channels are found within the Confidentiality Criteria. However, both are applicable to services found within the Integrity Criteria.

Furthermore, readers may feel that certain services would have fit better into one or another of the four functionality criteria. The decision of the authors to place a service in specific location in the CTCPEC is based on the following guideline:

1. What type of *service* does the service provide?

   a. Confidentiality;
   b. Integrity;
   c. Availability; or
   d. Accountability.

2. Is there an historical precedent?
3. If the service *can* fall into more than one functional criteria, to which does it *best* fit?

Tagging

Tagging is the process the TCB uses to encapsulate a user or resource. For instance, a disk may be comprised of sectors, which the product encapsulates as files which have an associated security attributes. There can be user tags, process tags and object tags. For a given security policy, tags need not exist in a one to one mapping with users, processes or objects. For example:

1. If a system implements a label based mandatory confidentiality policy, a label may be attached to a user. This label is a user tag which can be associated with any number of concurrent users.
2. A system which implements a label based discretionary integrity policy could be designed where a user may place a label on a program so that all processes executing that program carry that process label regardless of the user who runs the program.
3. The user identity of a user may be a unique tag only existing in one to one correspondence with users.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Access Matrix

The concept of an access matrix is used as a target for mapping the functionality of a product into a non-proprietary space. A sample access matrix is show in table 3 where the tag names are placed along each axis and the modes of access allowed or disallowed are placed in each matrix element.

	User 1	User 2	User 3	User 4
File 1	r		r	
File 2		rw	rw	
File 3				w
File 4				r

Table 3 Sample Access Matrix of User Tags and Object Tags

The Confidentiality and Integrity Criteria make use of this under their Discretionary and Mandatory services as laid out in table 4.

	Level:			
Service:	**1**	**2**	**3**	**4**
CD	PxO	UxO	UxO complete	UxPxO complete
ID	UxO	PxO	PxO complete	UxPxO complete
CM	PxO	UxO	UxO all U,O	UxPxO all U,P,O
CD	UxO	PxO	PxO all U,O	UxPxO all U,P,O

U = User Tag P = Process Tag O = Object Tag

Table 4 Use of Access Matrixes by Services

These services form a complete means of controlling *WHO* (user) can do what *HOW* (process) to *WHAT* (object) and are the basis for access mediation in the CTCPEC. At the highest levels, the matrix will have all three dimensions, one each for the user tags, process tags and object tags.

DRAFT

CD-1, CM-1, ID-1, and IM-1

These levels of service were included to allow for services which provide a mechanism which meets the goal of the service but which does so in a manner which would fit better under the opposite service (e.g., CD-1 is like ID-2). In this manner policies which perform integrity on the basis of a user tag, for instance, can be evaluated.

Confidentiality

Nominally, the services listed under Confidentiality provide services for the protection from disclosure of information.

Discretionary Confidentiality (CD)

The CD levels of service are aimed at user controls over the sharing of information between users. The CD-2 level allows for an extremely large breadth of possibilities but there is no non-ambiguous quantitative way to subdivide CD-2. The next logical non-ambiguous stage after a partial matrix is a complete matrix.

Mandatory Confidentiality (CM)

The CM levels of service are aimed at administrative controls over the sharing of information between users. By having administrative controls, complete maps of possible information flows are possible and then the access matrix can then be implemented to restrict information flows. With a discretionary policy this cannot be done because another user changing access will affect where information can flow.

One of the requirements for a CM policy at CM-3 and above is that it apply to *all* objects. This allows an administrator to completely specify the information flows in the product, subject of course to Covert Channels and flaws.

Object Reuse (OR)

The OR service level details the requirements for a service that underpins the CD, CM, ID and IM services such that deallocation leads consistently to destruction of information so that it is no longer available. The CTCPEC does not preclude the use of an approved encryption mechanism in performing object reuse.

Covert Channels (CC)

The CC levels of service detail increasing work by the vendor in removing the risk of covert information leakage outside of the enforcement mechanisms of the products security policy. Although found under Confidentiality, a CC rating may apply to any of IM, ID, CM and CD. The vendor must state in the covert channel policy which of these other policies the covert channel rating is being applied to.

Absolute bandwidth requirements do not seem to create a non-ambiguous format since the risk from covert channels varies greatly depending upon the ease of exploitation and the operation environment in which a product is used. As such, no bandwidth limitations are used in the CC levels of service. This does not preclude operational requirements where bandwidths and the covert channels of a product

The CC-3 requirement that all identified covert channels be removed is an extreme but easy to verify non-ambiguous quantitative improvement in a product.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Integrity

Discretionary Integrity (CD)

Mandatory Integrity (CM)

Physical Integrity (IP)

Rollback (IR)

Separation of Duties (IS)

Self Testing (IT)

Availability

Containment (AC)

Fault Tolerance (AF)

Availability

To be met it will probably require that a product have some fairly severe functionality restrictions in terms of what the product does but there are environments where this protection is needed.

The services under integrity aide in preserving the integrity of the product and preventing modification of data.

The ID levels of service are aimed at user controls over the sharing of information between processes which affects how information can be shared. The ID levels are the dual of the CD levels.

The IM levels of service are aimed at administrative controls over the sharing of information between processes. The IM levels of service are the dual of the CM levels.

The IP levels of service denote increasing ability on the part of the product to defend itself from physical attacks such as attempts to modify or remove parts of the product. These ratings are an integral part of being able to use a product in a stand-alone environment where other physical protection such as locked rooms are unavailable.

The IR levels of service denote finer granularity of control over protected objects in being able to undo actions.

The IS levels of service denote increasing ability of the product to separate functions into compartments such that users have no more available functionality than needed to perform an assigned duty.

The IT levels of service provide for increasing abilities to detect faults on the premise that it is not the test coverage, which is assumed to be complete, but instead the frequency with which the tests are run.

With respect to test coverage, this issue must be a part of the Self Testing policy.

The AC levels of service provide for increasing coverage or granularity of quotas to restrict users from hoarding resources and thereby denying other users access.

The AF levels of service are closely tied into the AR levels. A product that is fault tolerant must be robust. Fault tolerant products allow for replacement of faulty components without incurring any disruption in service. Therefore, the removal of a component from an active product will not result in the product going down or from the users noticing any disruption in service.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Robustness (AR)

The AR levels of service provide for increasing reliability under component failure. The method used to achieve robustness is vendor specific. However, the product must be capable of providing continuous service even during component failure. The definitions of "full service" and "degraded service" are also vendor specified. A product will enter "degraded service" whenever specific components fail. At the highest level of service the product will *not* enter a degraded level of service regardless of component failure. Full service will always be maintained.

Recovery (AY)

The AY levels of service provide for increasing host recoverability. Products which are capable of recovering automatically *and* ensure that their security policies have not been compromised are capable of attaining the highest service levels. A level of "ease of use" is attained in that products at AY-2 or AY-3 can guarantee that, for most product crashes, the product will restart successfully.

At AY-1, because it requires only manual recovery, there is a possibility that periods may exist during which the product will remain down due to lack of operator intervention.

Accountability

The services under accountability aide in guaranteeing users of the product are identified and monitored per the security policy.

Audit (WA)

The WA levels of service provide auditing capabilities by which user and process actions upon objects can be monitored and recorded. Actions deemed contrary to the security policy can then be actioned by the TCB.

Identification and Authentication (WI)

The WI levels of service provide identification and authentication capabilities to the TCB which allow the TCB to uniquely identify all users or processes attempting access. The TCB's use of identification and authentication in conjunction with audit allows for the proper application of the security policy.

Trusted Path (WT)

The WT levels of service a capability whereby the user or the TCB can request a guaranteed channel of communication between one another. This capability can be used to initially request service from the TCB or for the user or TCB to acknowledge requests for specific restricted commands.

Assurance[12]

Assurance in the CTCPEC is targeted at the Vendor. All requirements found within the Assurance Criteria reflect documentation, Vendor, and product requirements which will be used during an evaluation. The documentation and other information requested from a Vendor are used by the Evaluation Team to determine whether the Vendor has successfully met the requirements of the targeted trust level and the targeted functionality.

[12] To be completed for Final Release.

APPENDIX B

Constraints

Introduction

This appendix provides a general guide to understanding the constraints associated with individual levels of service. The constraints listed in the Criteria are the ones directly required for the given service to perform properly. The constraint list is a *minimal* list. Those services listed may, themselves, be constrained by other services. This appendix provides the expanded set with the additional services indicated in italics and provides a discussion of the rationale for their definition.

Each service is discussed in turn, in the order of appearance found in the Criteria proper.

Scope

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

Covert Channels

CC–1 Constraint: CR-1, T-3

The TCB requires supporting functionality and assurance to provide for effective identification of covert channels. Specifically, CC-1 requires Object Reuse at CR-1 and Assurance at T-3 as constraints.

Object Reuse prevents "overt" channels from being exploited: between allocations to different users, the contents of shared protected objects are cleansed of residual data. Once these common storage areas are eliminated, only covert channels remain as potential information flows. If Object Reuse is not provided by the product then identification of covert channels becomes an extraneous exercise since "overt" channels are much easier to exploit.

Assurance at T-3 requires the TCB to be internally structured so as to ease assurance analyses. An understandable internal structure is required to facilitate an effective covert channel analysis as this analysis requires a detailed understanding of the TCB. The level of detail of the development evidence is also increased at T-3 and this will aid in the comprehension of the TCB.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

CC–2 Constraint: CR-1, WA-1, *WI-1*, T-3

In addition to the functionality and assurance requirements of CC-1, the TCB requires supporting functionality to provide for effective auditing of the identified covert channels. Specifically, CC-2 requires Audit at WA-1 as a constraint which is further constrained by Identification and Authentication at WI-1

Audit at WA-1 requires that the TCB be able to perform basic auditing of security relevant events and provide an audit trail. If identified covert channels are to be audited, the TCB must be able to perform at least this level of auditing.

WA-1 also requires that the audit trail contain sufficient information to recover the identity of the users involved in each audited event. In order to provide this service, the TCB must be able to uniquely identify individual users, a service which is provided by WI-1.

CC–3 Constraint: CR-1, T-3

Covert Channels at CC-3 requires that identified covert channels be eliminated. The requirement for auditing is therefore eliminated and the rationale provided for CC-1 applies.

Discretionary Confidentiality

CD–1 Constraint: CR-1, WI-1

The TCB requires supporting functionality to allow authorized users to control the flow of information to protect resources against unauthorized disclosure. Specifically, CD-1 requires Object Reuse at CR-1 and Identification and Authentication at WI-1 as constraints.

Object Reuse ensures that all previous information content of a protected object is unavailable prior to reassignment or reallocation. If this information were available, unauthorized users could have access to it and disclosure would occur.

CD-1 requires that requests for changes to access mediation information be serviced by the TCB based upon the *user tag* of the requesting user or process. In order to provide this service, the TCB must be able to uniquely identify individual users, a service which is provided by WI-1.

CD–2 to CD-4 Constraint: CR-1, WI-1

In addition to the rationale provided for CD-1, Identification and Authentication at WI-1 is further required because requests for changes to access mediation information is serviced by the TCB base upon the *user tag*. In order to provide

this service, the TCB must be able to uniquely identify individual users, a service which is provided by WI-1.

Mandatory Confidentiality

CM-1 — Constraint: CR-1, IS-1, *WI-1*

The TCB requires supporting functionality to allow an authorized administrator to control the flow of information to protect resources from unauthorized disclosure. Specifically, CM-1 requires Object Reuse at CR-1 and Separation of Duties at IS-1 which is further constrained by Identification and Authentication at WI-1.

Object Reuse ensures that all previous information content of a protected object is unavailable prior to reassignment or reallocation. If this information were available, unauthorized users could have access to it and disclosure would occur.

CM-1 requires that requests for changes to access mediation information be service by the TCB for administrators and users to whom the required authority has been delegated. IS-1 ensures the TCB is able to identify administrative and nonadministrative user roles and their respective functions. In order to provide this separation of duties the TCB must be able to uniquely identify individual users, therefore, WI-1 becomes an additional constraint.

CM-2 to CM-4 — Constraint: CR-1, IS-1, WI-1

The rationale provided in CM-1 applies, however, WI-1 is now a direct constraint. This due to the fact that access mediation by the TCB is based upon the tag of the user. In order to provide this service, the TCB must be able to uniquely identify individual users, a service which is provided by WI-1.

Discretionary Integrity

ID-1 — Constraint: CR-1, WI-1

The TCB requires supporting functionality to allow users to control the flow of information to provide data integrity or product integrity. Specifically, ID-1 requires Object Reuse at CR-1 and Identification and Authentication at WI-1 as constraints.

Object Reuse ensures that all previous information content of a protected object is unavailable prior to reassignment or reallocation. If this information were available, unauthorized modification could occur since a newly created object could contain information that was not placed there by the creator of the object.

ID-1 requires that access mediation by the TCB be based upon the tag of the user and that requests for changes to access mediation information be serviced by the TCB based upon the *user tag* of the requesting user or process. In order

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Constraints

to provide this service, the TCB must be able to uniquely identify individual users, a service which is provided by WI-1.

ID-2 to ID-3 **Constraint: CR-1, WI-1**

The same rationale as ID-1 applies, however, WI-1 is only required because requests for changes to access mediation information be serviced by the TCB based upon the *user tag* of the requesting user or process.

ID-4 **Constraint: CR-1, WI-1**

The same rationale as ID-1 applies.

IM-1 **Constraint: CR-1, IS-1, WI-1**

The TCB requires supporting functionality to allow users to control the flow of information to provide data integrity or product integrity. Specifically, IM-1 requires Object Reuse at CR-1, Separation of Duties at IS-1 and Identification and Authentication at WI-1 as constraints.

Object Reuse ensures that the previous contents of a protected object are unavailable after reallocation. This ensures objects do not contain extraneous information which can be used by the product to improperly modify another object.

Access mediation by the TCB is based upon the tag of the user, therefore, the TCB must be able to uniquely identify individual users and WI-1 provides this service.

IM-1 requires that requests for changes to access mediation information be service by the TCB for administrators and users to whom the required authority has been delegated. In order to provide this service, the TCB must be able to identify administrative and nonadministrative user roles and their respective functions, a service which is provided by IS-1. In order to provide this separation of duties the TCB must be able to uniquely identify individual users, therefore, WI-1 becomes an additional constraint.

WI-1 is both a direct and additional constraint in this case.

IM-2 to IM-3 **Constraint: CR-1, IS-1, *WI-1***

The same rationale as IM-1 applies, however, access mediation is not based upon the tag of the user. WI-1 is required only as an additional constraint.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Containment

IM-4 — Constraint: CR-1, IS-1, WI-1

The same rationale as IM-1 applies.

Rollback

IR-1 to IR-2 — Constraint: WI-1

This applies to all levels of the Rollback service. This service provides the ability to undo an action or a series of actions and return a protected object to a previous state. The TCB requires supporting functionality to provide this service. Specifically, IR-1 to IR-2 requires Identification and Authentication at WI-1 as a constraint. Rollback ensures that an automated means to allow authorized users to undo a defined set of operations exists. In order to provide this service each user must be uniquely identified to the TCB, therefore, WI-1 becomes a constraint.

Separation of Duties

IS-1 to IS-3 — Constraint: WI-1

This constraint applies to all levels of the Separation of Duties service. This service provides for the compartmentalization of responsibility. The TCB requires supporting functionality to provide this service. Specifically, IS-1 to IS-3 requires Identification and Authentication at WI-1 as a constraint.

The TCB must be able to identify administrative and nonadministrative user roles and their respective functions. In order to provide this separation of duties the TCB must be able to uniquely identify individual users, therefore, WI-1 becomes a constraint.

Containment

AC-1 to AC-3 — Constraint: IS-1, *WI-1*

This constraint applies to all levels of the Containment service. This service allows the TCB to control the use of services and resources by users. The TCB requires supporting functionality to provide this service. Specifically, AC-1 to AC-3 requires Separation of Duties at IS-1 which is further constrained by Identification and Authentication at WI-1.

Requests for changes to access mediation information must only be serviced by the TCB for administrators and users to whom the required authority has been delegated. In order to provide this service, the TCB must be able to identify administrative and nonadministrative user roles and their respective functions. IS-1 provides this service. The TCB must be able to uniquely identify individual users to separate the duties, therefore, WI-1 becomes a constraint.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Constraints

Fault Tolerance

AF-1 to AF-2 — Constraint: IS-1, AR-1

This constraint applies to all levels of the Fault Tolerance service. This service allows the TCB to ensure availability of the product after component failures. The TCB requires supporting functionality to provide this service. Specifically, AF-1 to AF-2 requires Robustness at AR-1 and Separation of Duties at IS-1 which is further constrained by Identification and Authentication at WI-1.

Fault Tolerance requires that a set of the product's components can be replaced without incurring a service discontinuity. In order for hot-replacement to occur the product must provide Robustness. Without Robustness the product would incur a service discontinuity. In order to provide this service, Robustness at AR-1 is required.

An administrator, or users to whom the required authority has been delegated are able to replace any protected component. In order to provide this service, the TCB must be able to identify administrative and nonadministrative user roles and their respective functions. IS-1 provides this service. The TCB must be able to uniquely identify individual users to separate the duties, therefore, WI-1 becomes a constraint.

Robustness

AR-1 to AR-3 — Constraint: IS-1, *WI-1*

This constraint applies to all levels of the Robustness service. This service allows the TCB to ensure availability of the product after component failures. The TCB requires supporting functionality to provide this service. Specifically, AR-1 to AR-3 requires Separation of Duties at IS-1 which is further constrained by Identification and Authentication at WI-1.

??????????????

The product notifies an administrator of the failure of any protected component. In order to provide this service, the TCB must be able to identify administrative and nonadministrative user roles and their respective functions. IS-1 provides this service. The TCB must be able to uniquely identify individual users to separate the duties, therefore, WI-1 becomes a constraint.

??????????????

Recovery

AY-1 to AY-3 — Constraint: IS-1, *WI-1*

This constraint applies to all levels of the Recovery service. This service allows the TCB to return to a known trusted state after a product failure or service discontinuity. The TCB requires supporting functionality to provide this service. Specifically, AY-1 to AY-3 requires Separation of Duties at IS-1 which is further constrained by Identification and Authentication at WI-1.

After a product failure or service discontinuity, the TCB enters a state where only administrators, and users to whom the required authority has been delegated, are capable of returning the product to normal operation. In order to provide this service, the TCB must be able to identify administrative and nonadministrative user roles and their respective functions. IS-1 provides this service. The TCB must be able to uniquely identify individual users to separate the duties, therefore, WI-1 becomes a constraint.

Audit

WA-1 — Constraint: WI-1

The TCB requires supporting functionality to allow monitoring of potentially suspicious activity on the product. Specifically, WA-1 requires Identification and Authentication at WI-1 as a constraint.

WA-1 requires the audit trail to contain sufficient information to recover the identity of the users involved in each audited evert. In order to provide this service, the TCB must be able to uniquely identify individual users and WI-1 provides this service.

WA-2 to WA-5 — Constraint: IS-1, WI-1

In addition to the rationale provided for WA-1, Separation of Duties at IS-1 is required.

Audit review tools shall be available to administrators, and users to whom the required authority has been delegated, to assist in the inspection of the audit trail. In order to provide this service, the TCB must be able to identify administrative and nonadministrative user roles and their respective functions. IS-1 provides this service. In order to provide this separation of duties the TCB must be able to uniquely identify individual users, therefore, WI-1 becomes an additional constraint.

WI-1 is both a direct and additional constraint in this case.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Trusted Path

Constraints

WT—1 to WT-2 Constraint: WI-1

This constraint applies to all levels of the Trusted Path service. This service provides the ability to ensure direct communication between the users and the TCB. The TCB requires supporting functionality to provide this service. Specifically, WT-1 to WT-2 requires Identification and Authentication at WI-2.

The trusted path is used for initial identification and authentication. In order to provide this service, each user must be uniquely identified to the TCB and the TCB must use a protected mechanism to authenticate each user before allowing that user to perform any other TCB-mediated action. WI-1 provides this service.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX C

Fundamentals

Introduction

To fully understand the Canadian Criteria, one must understand its fundamentals. Computer security has been well understood for more than a decade. Advances in networking, distributed systems, applications, et cetera have left some feeling that the basic premises behind computer security are insufficient; however, the problem does not lie with the fundamentals.

With object-oriented systems coming to the forefront, and with movement towards distributed applications, the monolithic system is close to obsolete. To address this change the National Computer Security Center (NCSC) has written two interpretations of the Orange Book: The Trusted Database Interpretation (TDI) and the Trusted Network Interpretation (TNI).

The Canadian System Security Centre (CSSC) set about to create definitive information security criteria to address present and future computer systems.

Scope

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given for illustration purposes only and are not the only acceptable solution in meeting the Criteria.

Perspectives

The Trusted Computer System Evaluation Criteria (TCSEC or Orange Book) uses the terms *subjects* to define active and *objects* to define passive entities within a product. In contrast, the Canadian Criteria define all entities within a product as *objects.* In the Canadian Criteria an object can be instantiated into one of three types: user objects, process objects, or passive objects. These are commonly referred to simply as *users, processes,* and *objects.*

The TCSEC notion of a *subject* can be defined as the combination of a *user object* and *process object,* as illustrated in Figure 10. However, where the TCSEC would define numerous subjects, the Canadian Criteria can have a single *user* controlling numerous *processes.*

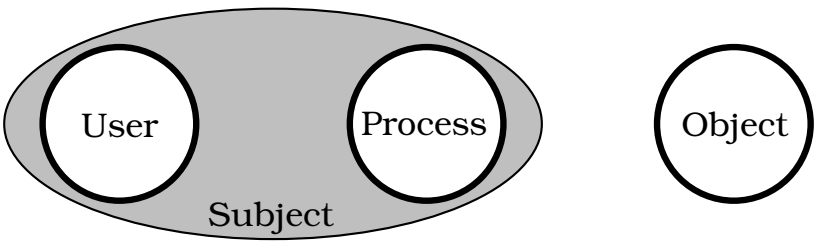

Figure 10: Trusted "Objects" in the TCSEC vs. Canadian Criteria

Objects

The Canadian Criteria defines all entities within the system as an *object.* Although many vendors will actually refer to a well defined subset of entities within their product as *"objects"*, for the purposes of this appendix, *object* refers to all objects.

A *user* is an abstraction of a physical user, the individual who is issuing commands to the product. This *user* is defined in terms of a security profile[13], which can be defined in terms of access rights, execution rights, privileges, et cetera, which is used by the product to associate a set of accessible *processes* and *objects*.

A *process* is an active *object.* The *user* activates an *object* and, upon successful activation the *object* becomes an active *process* associated with the *user* carrying all (or a subset) of the *user's* security profile.

Objects are passive entities upon which actions are taken (such as modification or disclosure). An *object* is a uniquely identifiable encapsulation of every "tagable" entity within a product, and includes resources, data, processes, and users. *Processes* and *users* address an *object* via the object's unique identifier.

All *objects* are tagged with access mediation information. The information is used by the trusted product to arbitrate access requests by a given *user* and *process* to an *object.*

Control Over Processes

Process's are objects which have been activated by a *user*[14] and having been activated define a domain of execution. This domain of execution includes all objects that the user and process may access — where the objects may include other processes and users.

Processes can run in two states: autonomous and nonautonomous. Autonomous *processes* have been activated by a *user* and have been sent into the background

[13] A security profile can contain role information, clearance information, etc. It is reflective of the product's security policy.

[14] The activating user can be a daemon or system ghost. These are administrative "users", although no physical individual is associated with them.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

to continue processing and do not necessarily require user interaction. Nonautonomous *processes* may require active user involvement and may imply that the *user* is interactively communicating with the *process.* Regardless of how the *process* is executing, the *process* is running on behalf of the *user* and is under the same restrictions.

The effective division of the TCSEC notion of *subject* into *user* and *process* is to ensure that access to *objects* can be restricted to specific *processes,* not just specific users.

Integrity requires that all attempts at modification of an object be verified by a reference monitor. Only authorized users, possibly via authorized processes, can modify an object. By restricting which *process* can manipulate which *objects* the product can provide enhanced auditing capabilities to ensure not only valid release of information, but valid manipulation of sensitive data. The finer granularity allows a vendor to create a product which can restrict access and modification not only to a given user but also to a given process. This ability allows the product to guarantee that modification or release of data is performed via controlled processes acting on behalf of authorized users.

The Reference Monitor

The traditional reference monitor has three attributes:

1. Always invoked, resulting in a barrier between accessing users and corresponding data;
2. Tamperproof, leading to domain isolation of the TCB; and
3. Small, leading to an easy to understand, minimally sized piece of code.

The NCSC has maintained that the reference monitor is *a* method by which one could *localize* security functionality, but not the *only* method. Many have taken the definition to imply that the reference monitor must be monolithic and interposed between subjects and objects. This interpositionary representation is held by neither the NCSC nor CSE. The reference monitor must be a well defined part of the TCB such that it is easily identifiable. The method and mechanism employed, if effective, is irrelevant.

The purpose of the reference monitor is to ensure that the flow of information between users, processes[15], and objects is mediated and ensured to be valid. The mechanism by which this is achieved can be varied. Appendix D discusses a small selection of possible mechanisms.

Although the Criteria refer to the reference monitor concept, CSE is willing to accept other methods of ensuring the mediation between users, processes, and

[15] Some products may combine the notion of users and processes, or have no users to speak of. In such cases, the reference monitor mediates the flow of information between the extant objects.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

objects. However, if a reference monitor is not used the onus is on the vendor to convince the evaluation authority of the validity of the new concept.

Classic View

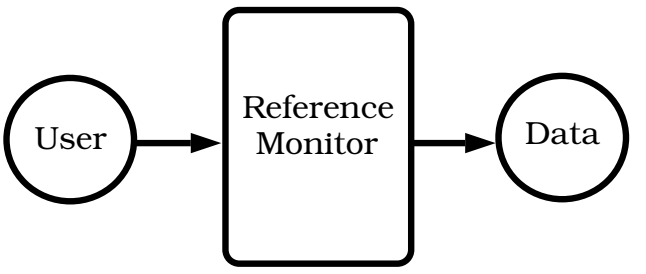

Figure 11: Classic View of a Reference Monitor

Figure 11 shows the classic view of the reference monitor: a barrier between user and data. This interpretation of the reference monitor has arisen from the wording of the Orange Book. The Orange Book states that the reference monitor enforces the security policy of the trusted computing base (TCB) and that in so doing all access to objects by subjects (users and processes in the CTCPEC) is to be monitored. The reference monitor allows or disallows access according to whether the request is authorized relative to access mediation information and tags (access controls and labels in the TCSEC) associated with the user and the data.

However, the reference monitor is a concept. In being a concept one must remember that one model does not necessarily best reflect the nature of the reference monitor. The primary task of the reference monitor is to ensure that the security policy is enforced. No where does it state that a single monitor must do the enforcing nor does it imply that the enforcement cannot be done by many "reference monitors" each associated with objects within the system.

To address the problems of distributed systems, object-oriented systems, and other non-monolithic systems, a reference monitor should be viewed as an encapsulator around an object[16]. Only valid access requests are passed through to the object. Therefore, information flowing out from an object towards another object is checked by each object's reference monitor for validity.

Further discussion of reference monitor concepts can be found in Appendix D.

[16] The encapsulation can be around the entire object space of the product.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Conclusion

This appendix offered the reader a general overview of the new terminology as used within the Canadian Criteria allowing the reader to compare the basic underlying premises of the Canadian Criteria to those of other criteria.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX D

Concepts

Introduction

This appendix addresses some of the more prominent concepts underlying computer security. The concepts are presented to illustrate the flexibility and acceptability of the concepts within the Canadian Criteria framework.

Scope

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are provided for illustration purposes only and do not represent the only acceptable solution to meeting the Criteria.

The Reference Monitor

As discussed in Appendix C, the reference monitor concept can trace its ancestry back to the Orange Book. The primary purpose of the reference monitor is to provide a known point of interaction for all transactions within a given product. For any protected object to be manipulated, the reference monitor **must** be invoked. It must also be tamperproof and small in size. Although the term "small" (or "minimalist") is relative, the absolute size of the reference monitor should not be the entire TCB.

This section covers various implementation viewpoints of the reference monitor. The monolithic, interpositionary view, though not discussed, is a valid implementation of the reference monitor.

Encapsulated View

A generalized form of the reference monitor concept is the encapsulated view, as illustrated in Figure 12, where every object is encapsulated by the reference monitor. The number of objects protected can vary from an individual object to the entire system. The former lends itself particularly well to distributed and object-oriented systems. Objects thus protected allow for the reference monitor to examine requests from various users, allowing only authorized requests for access to pass. The method by which this encapsulation occurs can vary: an interpositionary reference monitor separating users and objects; an envelope around each object, each object capable of discerning valid accesses; or an inherited trait from the parent object, possibly received at the object's create time.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

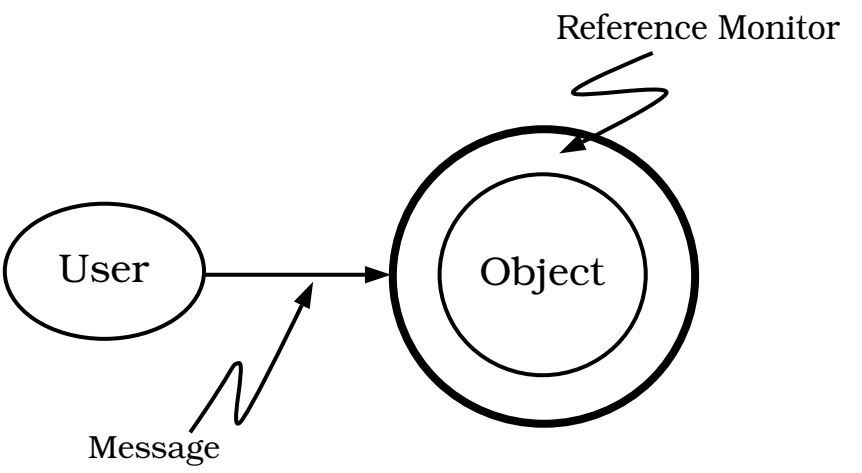

Figure 12: Reference Monitor As Encapsulator

The physical implementation may vary significantly: allowing for monitoring of messages between autonomous objects to interception of messages via the reference monitor acting as the message passing authority for the entire system.

Figure 13 has an underlying reference monitor which processes messages between a user and an object. Logically, the reference monitor can be ignored, in actual fact, however, the reference monitor is the mechanism by which the message gets propagated. When the reference monitor examines the message and deems the access request authorized, it passes the message on to the object.

DRAFT DRAFT DRAFT DRAFT DRAFT

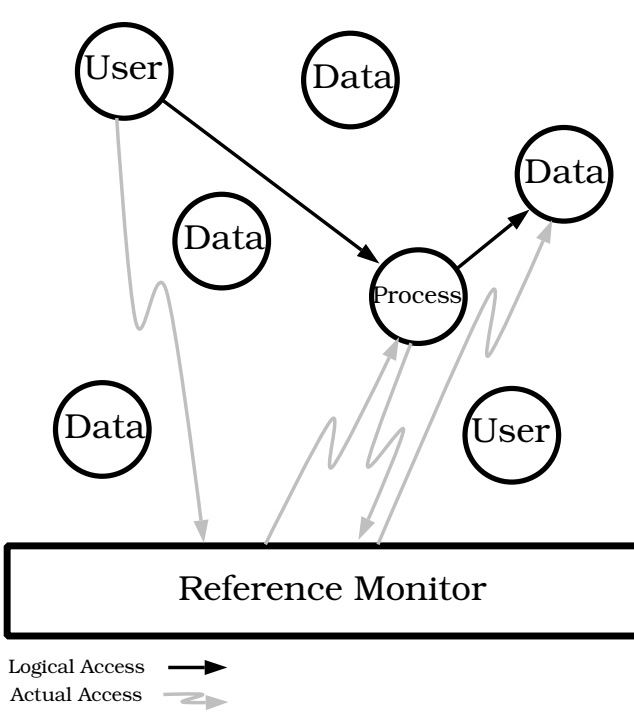

Figure 13: Entity Style (Message Passing) Reference Monitor 2

Figure 13 has a user attempting to access an object via a process. The user, in order to access the object, passes a message to the process, which is to request access on behalf of the user to the object. However, the reference monitor examines the message, attempts to ensure that it is a valid message for the target objects (the requested access by the user to the process, and the subsequent request by the user and process to the object, "data"), ensures that no security policy directives are being compromised, and then either rejects or allows the message. In actuality, this is what *typically* happens in a reference monitor, although on initial access attempt[17].

In an object-oriented[18] view the reference monitor can be viewed as the virtual machine[19]. The user and data are objects and the messages are object-oriented

[17] Most reference monitors enforce the security policy on initial access. The reasoning is that if the access was granted once, it will not have been revoked in the mean time. However, a Vendor may feel that the reference monitor should constantly reaffirm access rights. This is a decision left to the Vendor and the design/implementation team.

[18] This is, by no means, the only way to design an object-oriented reference monitor.

[19] A description of an artificial machine used by an interactive language, such as Smalltalk or Scheme. It can be viewed as the language's "kernel" and operating systems built on top of such languages typically allow access to the underlying language as an integral part of the operating system. Such languages usually have no requirement for an existing operating system, although coding of the virtual machine is simpler if an operating system exists.

messages which are handled by methods within each object. Security, when applied to this type of system, can be applied in a variety of locations. However, their impact is identical: a message either does or does not proceed to the object depending on the security policy.

Modularity

In current manifestations, modularity is identified with programs which are written piecemeal; each piece representing a logical operation or grouping of operations.

Extending this definition modularity becomes a group of "data" with its accompanying manipulation/access routines. This allows for data hiding and the ability of replacing or updating specific pieces of code without impacting the rest of the system.

In a trusted environment, modularity is divided into: i) the overall system and ii) the trusted computing base.

The Overall System

Modularity across the entire system demands that the system be implemented in a set of autonomous pieces. Each of these pieces, be they functions or programs, interact with other functions or programs via well defined input and output parameters.

The TCB

Even though the TCB may be a single program or hardware component, such as an operating system kernel or security card, the TCB must be structured so as to consist of modules, or programs/subprograms. Each of these modules are self-contained and require no additional data structures beyond their own scope[20]. Each module can pass information to other modules via a defined method of message passing (e.g., parameters, messages, IPC-style calls, etc.).

Figure 14 illustrates this view of modularity. Note that this figure illustrates both the modularity implemented by the overall system as well as the TCB.

[20] This may not be possible in all cases. The use of global values within a system, although frowned upon, is entirely acceptable given proper justification.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

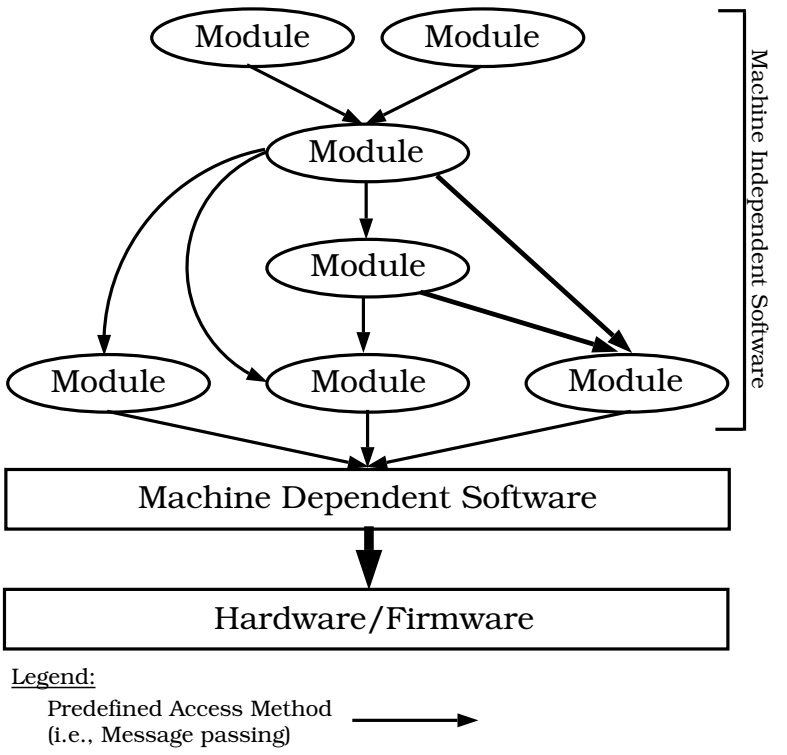

Figure 14: Modularity within A Trusted Environment 2

Although Figure 14 can be said to show “layering”, “layering” is not a required trait of a product. The Canadian Criteria places emphasis on modularity and data hiding rather than on layering. 3 4 5

By defining the modules, method of interaction, and the mechanisms used the vendor assures the evaluating authority that his product is modular. It is possible that, given a strong enough method of message passing, that the system can be evaluated piecemeal, module by module. Each module would then be evaluated against particular aspects of the Criteria. As each module is evaluated an overall rating would be assessed. 6 7 8 9 10 11

This form of an evaluation is known as a composable evaluation and implies that the system is, in fact, composed of numerous modules which are individually evaluated and evaluatable. Once each module is properly evaluated, the system requires an assurance rating on the mechanism used for message passing. If the strength of this mechanism can be assured to the same level as the individual modules, then the entire system receives the appropriate rating. 12 13 14 15 16 17

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Concepts

It is important to reiterate that the entire product, even if split across more than one physical machine, *must* implement a uniform security policy.

Networks & The Like

Extending the concept of modularization and composability still further we see that networks can be defined as a grouping of modules linked via a message passing scheme. This scheme must be well defined and implement a single security policy, as must the modules themselves. Each module is further subdivided into submodules. Each submodule, once evaluated, provides a rating for the entire module. As a final step, the entire set of modules defining the network must be evaluated within the context of the homogenous[21] network and its security policy.

The Canadian Criteria groups networks into either homogenous or non-homogenous (heterogenous) networks. A homogenous network is a network which may have numerous components, possibly from different vendors, but is designed to implement a single security policy across the network and each individual component within the network is designed and implemented to work with the network security policy. Each individual component must not, in any way, counteract, contradict, or compromise the security policy of the network.

A non-homogenous system, one running incompatible architectures and policies, would be unevaluatable since the issue of interconnection of the various components is an open research problem.

Other products which use a network as a base can be evaluated using a similar approach. Products such as distributed systems or distributed databases require a well defined security policy and a consistent use of the network facilities for their particular application. Further, the network capability must be shown to be an integral part of the product, not an extension outside the consideration of the evaluation.

Conclusion

The above discussion is not all-inclusive. Vendors are welcome to bring new ideas into CSE for examination. Concepts which are termed viable can be used within products which are to be entered into evaluation.

[21] Non-homogenous products can not be properly evaluated since this could result in two diametrically opposite systems being evaluated. Each of the two products may not have equivalent policies, interfaces, etc.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Tags

APPENDIX E A Guide to Object Mediation

Introduction

This appendix provides a general guide to understanding object mediation in the Canadian Criteria. The object mediation can be provided for the purposes of meeting the Confidentiality or Integrity criteria. This appendix provides guidance for understanding the concept of "tagging", and the difference between discretionary and mandatory mediation.

Scope

The guidelines established in this appendix apply to products under evaluation with the intention of meeting the Discretionary Confidentiality criteria (CD-1 to CD-4), Mandatory Confidentiality criteria (CM-1 to CM-4), Discretionary Integrity criteria (ID-1 to ID-4), and the Mandatory Integrity criteria (IM-1 to IM-4).

This appendix is intended as guidance only, and does not replace or supercede the requirements stated in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

Tags

The Discretionary and Mandatory Confidentiality criteria and the Discretionary and Mandatory Integrity criteria have been written to be as policy and mechanism independent as possible. However, while not requiring a specific implementation, the generic term **tag** is used to describe the implementation requirements for the product under evaluation. The Evaluation Authority can provide guidance on acceptable resource tags.

The Criteria do not specify the type of tag which can be used to satisfy the object mediation criteria. Rather, the term tag is used to indicate that some security attribute must be associated with users, processes, or objects. In fact, the object mediation services for discretionary and mandatory confidentiality, and discretionary and mandatory integrity could be identity-based, role-based, rule-based, or any combination of these.

In an identity-based service, the identity of the user, process, or the object will be used to determine whether access to an object is allowed. In a role-based service, roles are associated with users, processes, or objects. Access to an object is determined based on the current user role, the role associated with the current process, or the role required to access the object. In a rule based service,

users, processes, or objects are tagged (for example with a label). Rules are then established which define a relationship among the users, processes, and objects.

The "tagging" requirements as defined in the Canadian Criteria may be satisfied by any of the following:

- using a user identifier, a process identifier, or an object identifier;
- associating a role with a user, a process, or an object;
- associating a label with a user, a process, or an object (as defined in the TCSEC [2]);
- a cryptographic key which may be associated with a user, a process, or an object;
- a table which maintains a correspondence between a user, a process, or an object; or
- other approved (by the Evaluation Authority) mechanisms which allow users, processes, and objects to be "tagged".

Object mediation based on tags can be used to regulate the information flows within a product. An **information flow** exists within a product if there is some method of transferring information between users, processes, or objects. For example, if a user is capable of reading a file, then an information flow exists between the process reading the file and the object, and between the process displaying the file and the user. Similarly, if two processes are capable of sending messages to each other, then an information flow exists between the two processes.

Discretionary and Mandatory Mediation

The Criteria use the terms **discretionary** and **mandatory** when rating confidentiality and integrity object mediation services. A product which provides mandatory services must ensure that the information flow within the product is fixed by an administrator and cannot be changed over time by "general users[22]". On the other hand, a product which provides discretionary services allows "general users" to modify the information flow within the product.

The creation of additional information flows could be accomplished through: the modification of user, process, or object tags; the creation of new objects (includes copying existing objects); and the exportation and importation of objects. Therefore, a product which controls the creation of additional information flows can be rated against the Mandatory Confidentiality and Mandatory Integrity criteria while a product which cannot control the creation of addition information flows can be rated against the Discretionary Confidentiality and Discretionary Integrity criteria.

As an example, consider a product which has three users: Doe, Drake, and Admin. Admin has been assigned the task of security officer for the product

[22] For the purposes of this discussion, a "general user" is defined as any user of the product who has not been granted authorization to allow the modification of information flows by an authorized administrator.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

and has been granted all authorizations associated with that task. Doe and Drake have not been assigned any authorizations, and as such, are general users of the product.

Accuracy of Tags

In a product that enforces a mandatory object mediation policy (confidentiality or integrity), general users are not capable of changing an object's tag under any circumstance. In the above example, if the information flow policy created by Admin specifies that Doe and Drake cannot share information, then neither Doe nor Drake should be capable of modifying an object's tag such that the other user is capable of accessing the object.

In contrast, a product that enforces a discretionary object mediation policy (confidentiality or integrity) can allow general users (as defined in the policy) to modify an object's tag. For example, a product might allow the authorized user of an object (the authorized user) to modify its tag. Then, in the above example, Doe can modify the tag associated with Doesfile to which he is an authorized user, to modify the information flow such that Drake is now capable of accessing it.

Creation of New Objects

In a product that enforces a mandatory object mediation policy (confidentiality or integrity), a default or predefined tag must be associated with all objects created by all users. In the above example, if the information flow policy created by Admin specifies that Doe and Drake cannot share information, then neither Doe nor Drake should be able to create an object that is accessible by the other user. In addition, rules must exist for determining the tag that must be associated with copied objects, such as assigning a default or predefined tag to all copied objects, or assigning the most restrictive tag to the copied object. But nevertheless, if Doe copies an object which was not accessible to Drake, then

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX **F**

A Guide to Confidentiality

Introduction

This appendix provides a general guide to understanding Confidentiality in the Canadian Criteria. Confidentiality is the prevention of unauthorized disclosure of information. Confidentiality includes criteria for *Covert Channels*, *Discretionary Confidentiality*, *Mandatory Confidentiality* and *Object Reuse*. This appendix provides guidance on meeting the criteria and discusses considerations for designing mechanisms for these Confidentiality services.

Scope

The guidelines established in this appendix apply to products under evaluation with the intention of meeting the Covert Channel Analysis criteria (CC-1 to CC-3), Discretionary Confidentiality criteria (CD-1 to CD-4), Mandatory Confidentiality criteria (CM-1 to CM-4), and the Object Reuse criteria (CR-1).

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

Overview of Confidentiality

In all computer systems information flows in one of two basic directions: from the user into an object, or from an object out to a user. The paths taken can vary as can the input and output media. Confidentiality is, therefore, defined as the proper flow of information from an object to a user or a process.

Proper flow is defined as information flowing to an authorized user, perhaps via an authorized process. The information is tightly coupled to the user so as to ensure that the user is authorized to receive this information. The underlying principles against which the information flow is governed is defined by a Confidentiality Policy.

A complete security policy includes a confidentiality policy for the confidential services provided by the product. Furthermore, the approved policy will define the protected objects, the access permissions, and define the information flows in terms of mandatory and discretionary controls to users.

Covert Channels

Covert Channel Bandwidths

A product for which a covert channel analysis has been performed will be rated against the Covert Channel Analysis Criteria. The purpose of the covert channel analysis is to identify any covert channels that exist in the product. All covert channels must be identified in the analysis.

The Criteria do not impose any bandwidth restrictions on covert channels. A "reasonable" bandwidth restriction today may become too difficult to achieve in the future or may unnecessarily restrict advances in technology. Furthermore, any bandwidth restrictions imposed independent of the operational environment would always be arbitrary.

However, this does not imply that the bandwidth of covert channels should not be limited. In fact, the overall strength of the confidentiality mechanisms will depend directly upon the maximum bandwidth of any covert channels which remain in the product. For example, given an operational environment such that no covert channels with a bandwidth greater than 1 Kbs can be tolerated, a product built to meet the requirements of CD-4 and CM-4 is not useful if a 1 Mbs covert channel exists which is easily exploitable.

Therefore, a Vendor should determine, given the operational environment in which the product will be used, the maximum covert channel bandwidth acceptable to the products' users. The product should then be designed and built to ensure that this maximum covert channel bandwidth is not exceeded. The Evaluation Authority can provide guidance on acceptable covert channel bandwidths for different operational environments.

It should also be noted that other markets do impose restrictions on the maximum covert channel bandwidth acceptable in a product (e.g., U.S. and European restrictions on covert channel bandwidths, etc.). The Evaluation Authority can provide guidance to those Vendors which are interested in that market.

Storage and Timing Channels

The Criteria do not distinguish between covert storage channels and covert timing channels for the following reasons:

1. covert channels usually require an element of storage and of timing; and
2. there are no bandwidth restrictions on either type of covert channel, and therefore, there is no need to differentiate between covert timing channels and covert storage channels.

Vendors may or may not wish to separate their covert channel analysis into sections dealing with covert storage channels and covert timing channels. However, Vendors should differentiate between hardware, firmware, and software covert channels in their analysis.

Aggregate Covert Channels

As part of the covert channel analysis, Vendors are required to identify any aggregate covert channels that exist in their product. However, the Vendor is only required to document any aggregate covert channels which result from the same method of exploitation. Those aggregate covert channels which require different methods of exploitation are assumed to interfere with each other and as such, do not produce a combined covert channel bandwidth which is significantly higher than either of the separate bandwidths.

Meeting the Criteria

At the lowest level (CC-1), a Vendor is required to perform a covert channel analysis. All covert channels must be identified in the analysis, as well as the maximum bandwidth of each channel. Aggregate covert channels which result from the same method of exploitation must be included in the analysis. The channels and their bandwidths will be verified during the product evaluation.

At the second level (CC-2), the Vendor's product must be capable of auditing a subset or all of the covert channels (including aggregate covert channels) that were identified in the covert channel analysis. The Vendor must have the subset approved by the Evaluation Authority.

At the highest level (CC-3), the intent is to eliminate all covert channels. The covert channel analysis must show that an attempt was made to uncover all possible covert channels. Each identified covert channel must be identified and eliminated from the product. The Evaluation Authority will verify that the product does not contain any covert channels.

Discretionary Confidentiality

A product provides discretionary object mediation if the information flow within a product can be changed over time at the discretion of general users.

Appendix E provides additional information on tags, information flows, and the difference between mandatory and discretionary object mediation.

Discretionary Security Policy

The Discretionary Confidentiality Criteria do not require a specific security policy. In order to meet the Discretionary Confidentiality Criteria, a Vendor must provide a discretionary security policy. A security policy may be chosen to satisfy the Discretionary Confidentiality Criteria which satisfies the criteria and **the policy must be approved by the Evaluation Authority**. The security policy must include the following information:

- a list of the objects that can be protected;
- the access modes supported;
- rules for import and export of protected objects;
- the authorized users which are capable of modifying tags; and
- the discretionary protection mechanism used (including the rules used by the mechanism to mediate access to the protected objects).

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Meeting the Criteria

Discretionary confidentiality is the protection against disclosure of an object's information through an improper process or to an unauthorized user, where the processes and users authorized to receive information from an object are controlled by the authorized user. The authorized user, is a user within the product who possesses specific types of access to a particular object. An authorized user (possessing the correct access) can control disclosure of the information by changing the Access Mediation Information (AMI) to the object to grant another user access to that object.

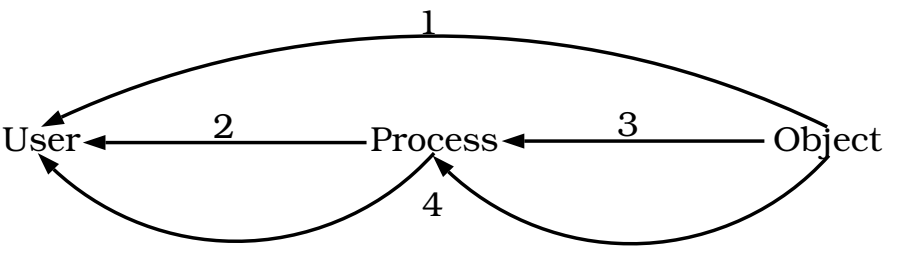

Figure 15: User-Process-Object Interrelationship for Object Mediation

The discretionary confidentiality ratings are given below with their corresponding relationship to Figure 15.

Rating	Path	Representation
CD–1	3	Partial or Full Access Mediation Information
CD–2	1 & 2	Partial Access Mediation Information
CD–3	1 & 2	Full Access Mediation Information
CD–4	4	Full Access Mediation Information

Table 5 Discretionary Confidentiality Ratings Summary.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

CD-1: Minimal Discretionary Confidentiality The weakest measure of ensuring non-disclosure is to place a sole restriction on the flow of information to a process. At CD–1, information flows from the object to only a select set of processes. Although there is no limitation on who the process may have been invoked by, and thereby who may be receiving the information, the TCB will restrict information flow to a fixed set of processes based on the object and process tag. Since the user is not aware of the information flow between objects

and processes, it is acceptable to provide a partial or a full representation of the access matrix of all process tags and protected object tags. The user who controls the domain to which the object belongs can modify the set of processes to which the object can pass information. Information is further protected by ensuring that objects retain their tags at all times, especially during import and export.

This lowest level of discretionary control is most useful in products requiring many users having access to information, but that there still be restrictions on how the information is accessed. Such products include databases and cryptomodules.

CD–2: Basic Discretionary Confidentiality The object's tag and user's tag contains information which is used by the TCB to mediate which users are allowed to gain access to the object's information. Further, invocation restrictions on processes indicating which users may invoke a particular process allow for a weak implicit control over the information path. At CD–2, the user, to whose domain the object belongs, is capable of placing restrictions on the object as to whom the TCB may disclose the object's information. Simultaneously, the user, to whose domain the object belongs, may place restrictions on those processes within his domain as to who may invoke the process. This allows for a weak, implicit link allowing a user to create a full map of all possible paths of disclosure from the object to the user.

The restriction on the control rights are at a coarse granularity, perhaps at the level restricting information flow to groups of users. An example of coarse control rights which would fit the definition of CD–2 is owner/group/world rights which indicate whether a particular user can receive information from an object, or invoke it, by checking whether they meet the coarse criteria set out by the restrictions within the owner/group/world access rights.

Protection bits such as those implemented in UNIX operating systems, would satisfy the requirements at CD–2 .

CD–3: Controlled Discretionary Confidentiality The primary difference between CD–2 and CD–3 is that CD–3 offers a finer granularity of control over which users can, and perhaps can not, receive information from an object or invoke a process. The controls which would fit into CD–3 would allow the authorized user, the user to whose domain the object belongs, to specify the access permissions assigned to the individual users and groups of users. It may be possible to include **and** exclude users from access to the list.

A full access control list (ACL) implementation would meet the CD–3 requirements.

CD–4: Advanced Discretionary Confidentiality CD-4 offers complete control of information flow within a trusted computer product. Each object, process, and user tag within the product is tagged with information which is used by the TCB to mediate the disclosure of information, but also which user/process pair is required in order for the TCB to permit access to the object's information.

Further, the ability of the TCB to mediate based upon the combination of the three tags, can restrict the disclosure of information to specific individuals. In this way, the object is guaranteed that the information being passed on is being sent to an appropriate user via an authorized process.

The control rights on the processes and objects are under the control of the authorized user, with each object existing within a particular user's domain.

An implementation of ACL's whereby it is possible to specify individual users, individual processes, and specific combinations of user/process pairs access to objects would meet CD–4 requirements.

Mandatory Confidentiality

A product provides mandatory object mediation if the information flow within a product is fixed by the administrator and cannot be changed over time by general users.

Appendix E provides additional information on tags, information flows, and the difference between mandatory and discretionary object mediation.

Mandatory Security Policy

The Mandatory Confidentiality Criteria do not require a specific security policy. In order to meet the Mandatory Confidentiality Criteria, a Vendor must provide a mandatory security policy which satisfies the criteria and **the policy must be approved by the Evaluation Authority**. The security policy must include the following information:

- a list of the objects that can be protected;
- rules for import and export of protected objects;
- the administrative roles which are capable of controlling the flow of information;
- rules for delegating authority to users to control the flow of information; and
- the mandatory protection mechanism used (including the rules used by the mechanism to mediate access to the protected objects).

Meeting the Criteria

Mandatory confidentiality is the protection against disclosure of an object's information through an improper process or to an unauthorized user, where the processes and users permitted to receive information from an object are controlled by the TCB enforcing a mandatory security policy. The mandatory security policy defines the access mediation information (AMI) within the product and the administrator which has control over the objects. Only this

administrator can set *delegate authority* to another user to be able to perform this specific function.

An object, as defined by the mandatory confidentiality policy, contains associated tags. Each tag represents a domain within which other objects attempting to receive information must reside. For example, if an object is tagged with a tag of "A", the process attempting access must be similarly tagged. Further, the user invoking the process, at the higher levels of mandatory confidentiality, must also reside within the same domain of objects (i.e., contain the same tag information). In this way the TCB can ensure that the user requesting information is authorized to receive the information.

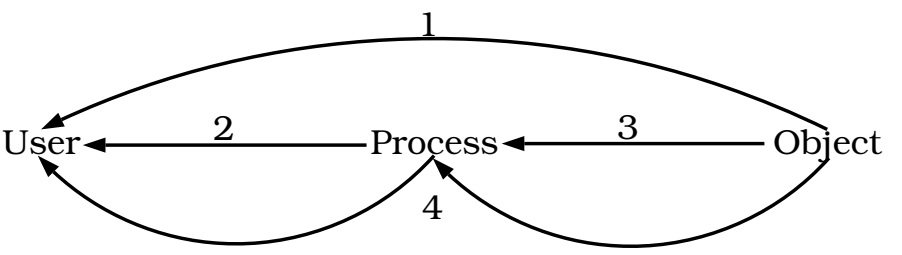

Figure 16: User-Process-Object Interrelationship for Object Mediation

The mandatory confidentiality ratings are given below with their corresponding relationship to Figure 16.

Rating	Path	Representation
CM–1	3	Subset of objects
CM–2	1 & 2	Subset of objects
CM–3	1 & 2	All objects
CM–4	4	All objects

Table 6 Mandatory Confidentiality Ratings Summary.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

CM-1: Minimal Mandatory Confidentiality The weakest measure of ensuring non-disclosure is to place a sole restriction on the flow of information to a process. At CM–1, the TCB will permit information to flow from an object to only a select set of processes. Although there is no limitation on who the process may have been invoked by, and thereby who may be receiving the

information, the TCB will restrict information flow from the object to a fixed set of processes. The restrictions create a domain, centered on the object, within which the processes and objects operate.

This lowest level of mandatory control is most useful in products requiring many users having access to information, but that there be restrictions on how the information is accessed.

CM–2: Basic Mandatory Confidentiality A select group of objects are tagged with specific domain information. User tags contain corresponding information regarding those domains which they can enter and thereby access information within those particular domains. Information is further protected by ensuring that objects retain their tags at all times, especially during import and export.

The object tag will contain restrictions as to the domain a user must be in prior to allowing information to be passed along. Further, invocation restrictions on processes are enforced by the TCB as to where particular processes can be invoked, and by whom. An authorized user may place users within appropriate domains so as to restrict invocation rights to processes. This allows for a weak, implicit link allowing the product to create domains within which all possible paths of disclosure are known.

The restriction on the control rights are at a coarse granularity, perhaps at the level restricting information flow to groups of users. An example of a coarse control right which would fit the definition of CM–2 is partitioned architectures which place users in particular domains. Each object is restricted to a particular domain to which a user is associated. Similarly, processes can be restricted to particular domains and must be invoked by users operating from a predefined domain.

CM–3: Controlled Mandatory Confidentiality This level requires all objects to be tagged with specific domain information. User tags contain corresponding information regarding those domains which they can enter and thereby access information within those particular domains. Information is further protected by ensuring that objects retain their tags at all times, especially during import and export.

The primary difference between CM–2 and CM–3 is that CM–3 must allow for a finer granularity of division among the domains within which a user is placed. It must be possible for the TCB to restrict information flow from **all** objects to any user or process except those within predefined domains and subdomains. Processes are similarly restricted to passing information only to those users which fall into the appropriate domains. All domains are defined by an authorized user, as defined by the mandatory security policy. Further, it may be possible to include and exclude particular users from specific domains and subdomains.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

CM–4: Advanced Mandatory Confidentiality CM-4 offers complete control of information flow within a trusted computer product. Each object within the product is tagged with information indicating not only which process may receive information, but also which user/process pairs are required in order for the TCB to permit information to flow from an object to a process or user.

These tags, in essence, create domains. Each domain is comprised of the following triple: [User/Process, Process/Object, User/Process/Object]. The TCB restricts information flow from a specific object based on this triple. Therefore, the product is able to guarantee that the information flows to the appropriate user via an authorized process all residing within a known, and authorized domain.

Object Reuse

A product provides object reuse if no information remains in a shared object and previous authorizations to the object are revoked before being assigned to a user or process[23]. This is applicable to initial assignment, allocation, or reallocation of the object from a previous user.

The criteria does not stipulate when the contents of an object should be cleared once a user releases it, as long as the previous user's information contained in the object is cleared by the time the next user gains access. Furthermore, previous authorizations and access to the object must be revoked to ensure that no user may gain access to a released object. Depending on the mechanism implemented, it is possible to clear the information in an object when released by a user or just before being reassigned to a user. However, if the latter method is implemented, the TCB is required to protect the object from any unauthorized access (while waiting in a pool of shared storage objects) until it is cleared upon reassignment.

It is acceptable to have a delayed release policy whereby if a user requests the same object which it previously released, it will be reallocated without being cleared as long as it has not been reallocated to any other user since it had been released.

Object Reuse may be implemented by encrypting the plaintext information and using cryptographic keys instead of clearing the information. As no other user can view the plaintext information without having the cryptographic key, the information is effectively unavailable to other users and processes because it is ciphertext. Encrypted representations of information will only be considered as unavailable if the encryption mechanism has been specifically approved for such an application by the Evaluation Authority and the Vendor meets the requirements of Appendix K.

[23] For clarity, *user* will be used instead of *user or process* in this section on Object Reuse.

Meeting the Criteria

Level CR-1 requires the Vendor to enforce an object reuse policy over all the product's storage objects. The Evaluation Authority recognizes that other methods may exist (such as the use of cryptography) to satisfy the Object Reuse Criteria, however, it must be approved by the Evaluation Authority in advance.

References

[1] Commission of the European Communities. *Information Technology Security Evaluation Criteria (ITSEC)*, Office For Official Publications of the European Communities, L-2985 Luxembourg, CD-71–91–502–EN-C, Version 1.2, June 1991.

[2] Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria*, DOD 5200.28–STD, December 1985.

[3] Morrie Gasser. *Building a Secure Computer System*, Van Nostrand Reinhold Company, New York, 1988.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX G

A Guide to Integrity

Introduction

This appendix provides a general guide to understanding Integrity in the Canadian Criteria. Integrity refers to both information integrity and product integrity. At the present time, the Canadian definition of Integrity is that of protecting against unauthorized modification. Information integrity is assumed to be a binary trait, an object either has retained its integrity, or it has not.

Scope

The guidelines established in this appendix apply to products under evaluation with the intention of meeting the Discretionary Integrity criteria (ID-1 to ID-4), Mandatory Integrity criteria (IM-1 to IM-4), Physical Integrity criteria (IP-1 to IP-4), Rollback Integrity criteria (IR-1 to IR-2), Separation of Duties Integrity criteria (IS-1 to IS-3), and the Self Testing Integrity criteria (IT-1 to IT-3).

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution to meeting the Criteria.

Overview of Integrity

In all computer systems information flows in one of two basic directions: from the user into an object, or from an object out to a user. The paths taken can vary as can the input and output media. Integrity is, therefore, defined as the proper flow of information from a user or process to an object.

Proper flow is defined as information flowing from an authorized user, perhaps via an authorized process to an object. The information is tightly coupled to the user so as to ensure that the user is authorized to write to the object. The underlying principles against which the information flow is governed are defined by an Integrity Policy.

A complete security policy includes an integrity policy for the integrity services provided by the product. Furthermore, the approved policy will define the protected objects, the access permissions, and the information flows in terms of mandatory and discretionary controls with respect to users.

Discretionary Integrity

Security Policy

A product provides discretionary object mediation if the information flow within a product can be changed over time at the discretion of general users.

Appendix E provides additional information on tags, information flows, and the difference between mandatory and discretionary object mediation.

The Discretionary Integrity Criteria do not require a specific security policy. In order to meet the Discretionary Integrity Criteria, a Vendor must provide an integrity security policy. A security policy may be chosen to satisfy the Discretionary Integrity Criteria which satisfies the criteria and **the policy must be approved by the Evaluation Authority**. The security policy must include the following information:

- a list of the objects protected;
- the access modes supported;
- rules for import and export of protected objects;
- the authorized users which are capable of modifying tags; and
- the discretionary protection service used (including the rules used by the service to mediate access to the protected objects).

Meeting the Criteria

Discretionary integrity is the protection against modification of an object by an unauthorized user[24] or process Every object "belongs" to a single user or group of users. This user (or group of users) has the right to modify the access mediation informtion (AMI) of the object to allow or disallow access to other users, groups of users, or processes.

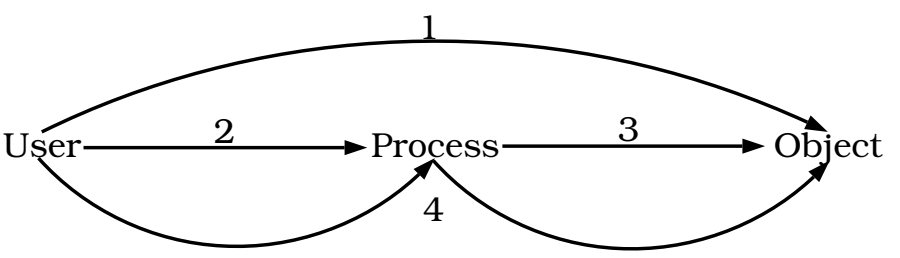

Figure 17: User-Process-Object Interrelationship for Object Mediation

The discretionary integrity ratings are given below with their corresponding relationship to Figure 17.

[24] The authorized user, is a user within the product who possesses specific types of access to a particular object.

DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Discretionary Integrity

Rating	Path	Representation
ID–1	1 & 2	Full or Partial Access Mediation Information
ID–2	3	Partial Access Mediation Information
ID–3	3	Full Access Mediation Information
ID–4	4	Full Access Mediation Information

Table 7: Discretionary Integrity Ratings Summary.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

ID-1: Minimal Discretionary Integrity The object tag will contain restrictions as to the users which are allowed to modify a particular object. Further, invocation restrictions on processes indicating which users may invoke a particular process allows for weak control over the information path. Information is further protected by ensuring that objects retain their tags at all times, especially during import and export.

At ID–1, the owner to whose domain the object belongs, is capable of placing restrictions on the object as to which users are allowed to modify the object. Processes are similarily controlled with the owner capable of restricting who can and can not invoke processes within the user's domain. This allows for a weak, implicit link allowing a user to create a partial or full map of possible paths of modification of the object by a user.

The restriction on the control rights are at a coarse granularity, perhaps at the level restricting information flow from groups of users. An example of a coarse control right which could possibly satisfy the definition of ID–1 is owner/group/world rights which indicate whether a particular user can modify an object, or invoke it, by checking whether they meet the coarse criteria set out by the restrictions within the owner/group/world access rights.

ID–2: Basic Discretionary Integrity A stronger method of ensuring unauthorized modification is to place a restriction on the object as to what process, or group of processes, can modify the object. At ID–2, the TCB will allow the object to be modified by only a select set of processes. Although there is no limitation on who the process may have been invoked by, and thereby who may be modifying the information, the TCB will be able to restrict the information flow from a fixed set of processes. The user who controls the domain to which the object belongs can modify the set of processes by which the object can be modified.

Protection bits such as those implemented in UNIX operating systems, could possibly satisfy the requirements at ID–2 .

ID–3: Controlled Discretionary Integrity The primary difference between ID–2 and ID–3 is that this level offers a finer granularity of control over which processes can, and perhaps can not, modify an object. The TCB is able to restrict the information flow from **all** processes.

A full access control list (ACL) implementation could possibly satisfy the ID–3 requirements.

ID–4: Advanced Discretionary Integrity ID-4 offers complete control of information flow within a trusted computer product. Each user, process, and object within the product is tagged with information indicating not only which process may modify information, but also which user/process pair is required in order for the TCB to permit the object to be modified. Further, the ability of the TCB to mediate based upon the combination of the three tags, can restrict the modification of information to specific users. This guarantees that the object modification is by an authorized user via an authorized process.

The control rights on the processes and objects are under the control of the authorized user of the specific object, with each object existing within a particular user's domain.

An implementation of ACL's whereby it is possible to specify individual users, individual processes, and specific combinations of user/process pairs access to objects could possibly satisfy CD–4 requirements.

Mandatory Integrity

A product provides mandatory object mediation if the information flow within a product is fixed by the administrator and cannot be changed over time by general users.

Appendix E provides additional information on tags, information flows, and the difference between mandatory and discretionary object mediation.

Security Policy

The Mandatory Integrity criteria do not require a specific security policy. Instead, any security policy may be used which satisfies the mandatory integrity criteria **provided that the policy has been approved by the Evaluation Authority**. The security policy must include the following information:

- a list of the objects protected;
- rules for import and export of protected objects;
- the administrative roles which are capable of controlling the flow of information;
- rules for delegating authority to users to control the flow of information; and

Meeting the Criteria

- the mandatory protection mechanism used (including the rules used by the mechanism to mediate access to the protected objects).

Mandatory integrity is the protection against modification of an object's information through an improper process or by an unauthorized user, where the processes and users permitted to modify information are controlled by the TCB enforcing an integrity security policy. The mandatory security policy defines the access mediation information (AMI) within the product and the administrator which has control over the objects. Only this administrator can set *delegate authority* to another user to have control over the objects.

An object, as defined by the mandatory integrity policy, contains associated tags. Each tag represents a domain within which other objects available for modification must reside. For example, if an object is tagged with a tag of "A", the process attempting modification access must be similarly tagged. Further, the user invoking the process, at the higher levels of mandatory integrity, must also reside within the same domain of objects (i.e., contain the same tag information). In this way, the TCB can ensure that the user attempting modification of an object is so authorized.

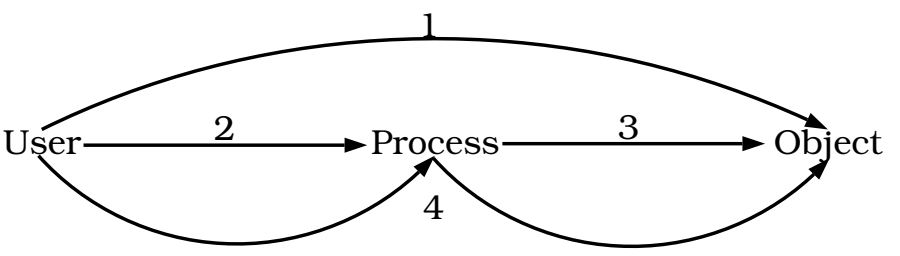

Figure 18: User-Process-Object Interrelationship for Object Mediation

The mandatory integrity ratings are given below with their corresponding relationship to Figure 18.

Rating	Path	Representation
IM–1	1 & 2	Subset of objects
IM–2	3	Subset of objects
IM–3	3	All objects
IM–4	4	All objects

Table 8 Mandatory Integrity Ratings Summary.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

IM-1: Minimal Mandatory Integrity A select group of objects are tagged with specific domain information. User tags contain corresponding information regarding which domains can be entered and thereby modify information within those particular domains. Information is further protected by ensuring that objects retain their tags at all times, especially during import and export.

The object will contain restrictions as to the domain a user must be in prior to allowing information to be modified. Further, invocation restrictions on processes are enforced by the TCB as to where particular processes can be invoked, and by whom. An authorized user may place users within appropriate domains so as to restrict invocation rights to processes. This allows for a weak, implicit link allowing the product to create domains within which all possible paths of modification are known.

The restriction on the control rights are at a coarse granularity, perhaps at the level restricting information modification to groups of users. An example of a coarse control right which could possibly satisfy the definition of IM–1 is partitioned architectures which place users in particular domains. Each object is restricted to a particular domain to which a user is associated. Similarly, processes can be restricted to particular domains and must be invoked by users operating from a predefined domain.

IM–2: Basic Mandatory Integrity A stronger method of ensuring unauthorized modification does not take place is to set a restriction on the object as to what process, or group of processes, can modify the object. At IM–2, a select group of objects are tagged with specific domain information and the TCB will permit an object to be modified by only a select set of processes. Although there is no limitation on who the process may have been invoked by, and thereby who may be modifying the information, the TCB will restrict the information flow from a fixed set of processes. The user who controls the domain to which the object belongs can modify the set of processes which can modify the object.

The select set of processes can only be modified by an authorized user as defined by the mandatory integrity policy.

IM–3: Complete Mandatory Integrity This level requires **all** objects to be tagged with specific domain information. An increase in the strength of the service is provided at IM-3 by having the TCB to restrict information flow from all processes from modifying an object within predefined domains and subdomains. All domains are defined by an authorized user, as defined by the mandatory security policy. Further, it may be possible to include and exclude particular users from specific domains and subdomains.

IM–4: Advanced Mandatory Integrity IM-4 offers complete control of information flow within a trusted computer product. Each object within the product is tagged with information indicating not only which process may modify information, but also which user/process pairs are required in order for the object to be modified.

These tags, in essence, create domains. Each domain is comprised of the following triple: [User/Process, Process/Object, User/Process/Object]. The TCB restricts information flow to a specific object based on this triple. Thus, the product is able to guarantee that the information being modified is being modified by an authorized user via an authorized process all residing within a known, and authorized domain.

The control rights on the processes and objects are under the control of a privileged user(s) as defined by the mandatory integrity policy.

Physical Integrity

Waiting for the IP criteria to be modified.
Verify once the IP criteria is fixed.

A product provides physical integrity if the product can indicate or restrict physical access to the TCB.

Security Policy

The Physical Integrity Criteria do not require a specific security policy. Instead, any security policy may be used which satisfies the Physical Integrity Criteria **provided that the policy has been approved by the Evaluation Authority**. The security policy must include the following information:

- a defined set of the product's components to which it applies;
- the normal operating environmental conditions;
- the protection administered in case of physical attack on the product or its components;
- the physical perimeter of the TCB;
- the tamper evident mechanisms;
- the tamper resistant mechanisms; and
- the tamper response mechanisms.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT

Meeting the Criteria

Physical Integrity services ensure the product's components and perimeter integrity against the threat of unauthorized use, physical modification, physical access, and extreme environments. The TCB protects the products integrity by passive or active means in response to the threat occurrence regardless if it is due to human error or not. The protection varies due to the level of protection built in to the product. Physical integrity compromise is defined to be unauthorized use of, physical access to, or physical modification of the protected components.

Physical Integrity services utilize mechanisms to indicate, restrict, and respond to physical access to the TCB depending on the level of service provided. A tamper evident mechanism is a passive device to indicate tampering which provides an audit of the occurrence, whereas a tamper response mechanism actively announces the event occurrence. A tamper resistant mechanism attempts to thwart the threat occurrence. These mechanisms are further explained below.

Tamper evident mechanisms are passive devices which will indicate the occurrence of a tamper event. A user is required to actively inspect the device to verify if it indicates an event has occurred. The device providing the protection is considered to be part of the TCB and can be considered a type of audit mechanism which indicates an event occurrence. Some examples of tamper evident devices are break-away labels, conformal coating (opaque tamper evident material), and bleeding paint.

By contrast, a tamper response mechanism initiates an action in response to the detection of a threat occurrence (physical integrity breach). The action may render the product inoperative, remove a part of the product from service, activate an annunciator (auditable or visual indicator), or zeroize the keys (in the case of a cryptographic module). The response mechanism is considered to be part of the TCB depending on the implementation. The response mechanism may generate an audit entry. An example of a tamper response mechanisms are hard opaque potting material (e.g., a hard opaque epoxy).

Tamper resistent mechanisms are built in such a way as to make it very difficult or almost impossible to defeat. This is to prevent a compromise by making it too difficult and time consuming to defeat. This can be used to provide the TCB time to respond with some corrective action to protect the product and the internal components. It is advisable to include a tamper response mechanism to take these corrective measures as no protection is absolute. For example, a potting compound used to encapsulate a cryptographic module Integrated Circuit (IC) will prevent probing and indicate tampering via scratches. Furthermore, the compound also provides a tamper response which damages the IC beyond repair if physical access is gained.

The physical integrity criteria is designed to be a metric to enable evaluation of the physical security implemented in products. These issues are more critical when the products incorporate cryptographic modules. The physical integrity

criteria was designed to be similar to FIPS 140–1 Physical Security[25] to address the physical security of cryptographic modules. The physical integrity criteria address FIPS 140–1 Physical Security starting with Level 2 for tamper evident counter measures at IP-1. The remaining FIPS 140–1 levels 3 and 4 correspond to IP-2 & IP-4 respectively.

IP-1: Basic Physical Integrity At IP-1, the TCB protects the physical perimeter by the use of tamper evident mechanisms to provide evidence of physical integrity compromise of the protected components. This level provides an audit trail of the unauthorized integrity compromise to inform the security officer of a possible perimeter breach. This prevents the security breach from going unnoticed and the officer, once aware, can prevent further integrity compromise. Note that this auditable action may only be noticed after the threat occurred.

This level only provides a record of this event and does not actively notify the security operator or take any action (this is a passive level).

IP-2: Intermediate Physical Integrity IP-2 provides real time detection by providing a response mechanism to the unauthorized attempt. The physical perimeter and its covers and openings are required to be protected by tamper response mechanisms to announce the detection of physical integrity breach of the protected components during an unauthorized attempt.

IP-3: Advanced Physical Integrity IP-3 requires, in addition to the requirements at IP-2, the TCB to protect the product and the components within, from extreme environmental conditions which would compromise the physical integrity of the TCB. An environmental extreme is any condition (temperature, humidity, voltage, current, frequency, EMI/EMC susceptibility, ionizing radiation, phase imbalance) which deviates from the established operating norms and tolerances.

Electronic devices and circuitry are designed to operate within a particular range of environmental conditions. If the devices or circuitry are operated outside of this range, their correct operation is not guaranteed. Therefore, any deliberate or accidental deviation outside of the specified normal operating range can cause erratic operation or failure of the electronic devices or circuitry within a product that can compromise the security of the product.

Environmental failure protection (EFP) mechanisms and environmental failure testing (EFT) are two different ways of addressing this threat to protect the product. EFP mechanisms are required to monitor and respond to conditions which deviate from the normal conditions while EFT are required to demonstrate

[25] The integrity appendix demonstrates how it is possible to interpret products meeting FIPS 140–13 and are intended as guidance only.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

that varying the environmental parameter outside of the product's norms give a reasonable guarantee that environmental conditions or fluctuations will not result in compromise of product security.

If a condition is to be determined to be outside of the product's normal operating range, then an EFP mechanism should:

1. shut down the product to prevent it from operating outside of the normal operating range;
2. immediately zeroize all plaintext cryptographic keys and other unprotected critical security parameters to prevent compromise; or
3. take corrective action such as employing an environmental regulator (temperature conditioner, humidity conditioner, or power conditioner).

IP-4: Complete Physical Integrity IP-4 requires, in addition to the requirements at IP-3, complete physical protection by enforcing the physical integrity policy over all of the products components. All components within the physical perimeter shall be protected by tamper resistant and tamper response mechanisms to prevent physical integrity compromise.

Rollback

A product provides rollback if the product can undo the last action or the last series of actions and effectively rollback to a known previous state.

Security Policy

The Rollback Criteria do not require a specific security policy. Instead, any security policy may be used which satisfies the Rollback Criteria **provided that the policy has been approved by the Evaluation Authority**. The security policy must include the following information:

- a defined set of protected objects for rollback;
- a defined set of operations for rollback;
- a specific authorized users; and
- the predefined period of time applicable for rollback.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

Meeting the Criteria

Rollback services provide the ability to return to a known valid (integrity preserved) previous state. This enables the user or process to recover from an error or a series of errors and return to a known valid state. The recover operation involves undoing the last error or series of errors over a predefined period of time.

A predefined period of time and a maximum number of operations dictate how many operations (commands) may be undone. These limits are to be specified in the TFM. The period of time applicable to operations permissible from rollback may be defined as the time between saves of an editor, for example.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Rollback

Rollback is a versatile service which enables users to recover from human error, product error (e.g., the editor dies, environmental failure, etc.), and to maintain integrity in databases and transaction based applications as shown below by enabling:

1. a user to undo a series of errors to the last known error free state;
2. a user to undo a series of operations to return to the last known state of the editor after it died;
3. a confused process to undo a series of operations and return to a known valid state; and
4. a database to undo the last operation if the record entry can not be completely updated.

All rollback requests must be audited. Rollback is not applicable to the audit log. The audit log is to maintain a record of all operations including initial operations and the rollback operations. This is to say that operations rolled back do not erase the audit trail of the original operations. The trail will include the initial operations and the operations carried out to undo the initial operations.

For rollback to be acceptable, the time to rollback actions required should be no greater than the original request. The time an operation takes to be done or undone should be the similar if not the same. For example, deleting a line in a file using an editor should require no more time than it took to paste it. Furthermore, the actual effort expended by the TCB to rollback an action may be longer than the original action itself. However, after initiating rollback, the user should be able to continue without undue delay (i.e., a distributed DBMS waiting for all the entries to be updated should not result in a long delay for a user).

The use of backup media is unacceptable as a rollback implementation since the process requires considerable time to retrieve the file and is usually a manual operation. Although, there are some automated file back up processes, it still requires more time than the initial operation to retrieve the file from the disk for the user.

IR-1: Restricted Rollback IR–1 provides an automated means for authorized users or processes to rollback a set of operations on protected objects over a predefined period of time.

IR-2: Advanced Rollback The primary difference between IR–1 and IR–2 is that this level offers the ability to rollback all operations.

Separation of Duties

Verify once the IS criteria is fixed.

A product provides separation of duties if there are multiple roles defining distinct functions.

Security Policy

The Separation of Duties Criteria do not require a specific security policy. Instead, any security policy may be used which satisfies the Separation of Duties Criteria **provided that the policy has been approved by the Evaluation Authority**. The security policy must:

- identify administrative and non administrative roles;
- list the roles and their respective functions; and
- define an explicit action which a user must perform in order to assume an authorized role.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

Meeting the Criteria

Separation of duties services provide a measure of information integrity by reducing the likelihood of an unauthorized action being taken, or limiting the effects of an unauthorized action if it does occur. In their simplest form, separation of duties services rated by the Criteria separate general users from administrative responsibilities — including product resources, information belonging to other users, and administrator-controlled security attributes.

The Criteria use the term *role* to denote the granularity of separation of duties required. There are two basic types of roles:

1. the basic separation of administrative duties performed by administrators from those duties which are performed by general users; and
2. the further separation of a role (administrator or user) into two or more strict, task-oriented functions.

The level of separation of duties achieved depends on the granularity of separation between user and administrative responsibilities.

Complete separation of duties is provided in those products which, in addition to separate administrator roles, provide the ability to define roles for users. For example, a bank would require distinct role definitions for bank tellers, loan officers, and clerks. Different definitions would also be required on a military supply system for order clerks, shipping and receiving personnel, and combat unit personnel.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Separation of Duties

Creating an audit administrator, account administrator, and operator roles (for example) will limit potential damage to only the information to which that particular administrator has been authorized. Finer grained separation services can be used to restrict the capabilities of a corrupt administrator to very specific duties thus limiting potential damage.

Separation of duties services must be supported by identification and authentication. This supporting functionality is expressed in the Criteria as a constraint to (at a minimum) WI-1 for separation of duties at all levels. A user is required to take an explicit action in order to assume a role for which they are authorized. This explicit action may be, for example, an identification and authentication event (i.e., login) meeting WI-1 or higher, the issuing of a command requesting recognition from the TCB, or the use of a badge and magnetic strip reader identifying the holder as an authorized user.

IS-1: Basic Separation of Duties At IS-1, the TCB must enforce separation of duties at the granularity of administrative functions versus general user functions. This limits the potential for unauthorized actions by general users against product resources or security attributes. This is the most basic form of separation of duties, but it provides for effective limits on the actions of general users.

Products having one administrator and all others defined as general users, as implemented in the UNIX operating system, could possibly satisfy the requirements at IS–1.

IS-2: Administrative Separation of Duties At IS-2, the product must support two or more distinct administrator roles with specific sets of administrative duties. One of these role definitions is required to be that of a security administrator (the security administrator role may be divided into an audit administrator role and an account administrator role). The other role is to be a non security administrator

The security administrator role must be defined such that security related duties can be performed by the security administrator only. The security administrator role definition must also limit the inclusion of non-security relevant functions to those which allow the security administrator role to be performed effectively. While not specifically required by the Criteria for other roles, the spirit of these measures should be followed in defining other non-security roles. It should be noted that roles need not be mutually exclusive, as certain functions or commands will be required by both an administrator and a user, or by different administrative roles, etc.

Products which have multiple administrative roles, each of which are authorized to perform a limited set of duties, could possibly satisfy these requirements.

IS-3: Privilege-based Separation of Duties The primary difference between IS–2 and IS–3 is that this level requires multiple distinct user roles to be defined for general users. This allows for applications requiring privileges for users or applications as warranted.

Products which define multiple user roles such as those implemented in the HP MPE operating system, could possibly satisfy these requirements.

Self Testing

A product provides Self Testing if the product can validate the correct operation of the TCB.

Security Policy

The Self Testing Criteria do not require a specific security policy. Instead, any security policy may be used which satisfies the Self Testing Criteria **provided that the policy has been approved by the Evaluation Authority**. The security policy must include:

- a complete description of the self test services;
- when the tests can be performed;
- what functions will the tests validate;
- how the tests are initiated; and
- what happens if the tests detect a corrupted TCB.

With each increase in rating, a similar increase in level of service provided occurs. Below is an explanative summary of each level of service.

Meeting the Criteria

Self Testing services provide a means of validating the correct operation of the TCB. Correct operation of the TCB ensures there will not be a loss of integrity due to product operation. The initial state of the product must be in a known, valid integrity state to process information. Otherwise, any information processed will have its integrity compromised.

If the TCB malfunctions or is tampered with, the self testing service will notify the administrator that the integrity of the product has been compromised or take corrective action to ensure that no further compromise occurs. A compromised product (or a product suspected of being compromised) must protect information from being processed from possible or further integrity compromise. Users already logged on are to be prevented continuing to process information and new users are to be denied logon access to ensure that no further integrity compromise occurs.

The critical functions of the product require testing to ensure a valid integrity state. The critical functions and the testing procedures are to be described in the Trusted Facility Manual (TFM).

Testing procedures can be initiated by administrative users according to procedures described in the TFM. However, automated testing initiated by the TCB increases the products integrity as human error is reduced. Furthermore, the

DRAFT

ability to periodically test the product operation ensures the products integrity after initial power-up.

IT-1: Basic Self Testing At IT-1, the product must include basic self testing services which can validate the correct operation of the TCB. There are no specific requirements as to what functions the tests are to cover or when it will be initiated as long as they are described clearly in the Trusted Facility Manual. The self testing service is under user control as to when it is initiated.

Products having a self test service which is under administrative control, could possibly satisfy the requirements at IT–1.

IT-2: Intermediate Self Testing This level differs from IT-1 as the TCB, at initial start-up, is capable of initiating the self tests to validate the correct operation of its critical functions. The critical functions validated by these tests must be described in the Trusted Facility Manual.

This level ensures that the product's integrity is preserved before allowing the user to proceed.

Products which initiate self tests on power–up or hard reset satisfy the requirements at IT–2.

IT-3: Advanced Self Testing The primary difference between IT–2 and IT–3 is that at IT–3 the TCB can initiate self tests during normal product operation in addition to initial start-up. The critical functions validated by these tests must be described in the Trusted Facility Manual.

Products which initiate self tests periodically during normal product operation in addition to when the power is switched on satisfy the requirements at IT–3.

References

[1] Gasser, Morrie. 1988. Building a Secure Computer System. *Van Nostrand Reinhold, 1988*

[2] Roskos, J.E., Welke, S. R., Boone, J.M., Mayfield, T. July 1990. *Integrity in the Department of Defense Computer Systems: 43–44.* Institute for Defense Analysis

[3] U.S. Department of Commerce, National Institute of Standards and Technology, *Security Requirements for Cryptographic Modules, (DRAFT), Federal Information Processing Standard Publication 140–1, February 10, 1993*.

[4] Canadian NITSIM 10/90 July 27, 1990[26]

[5] See reference # 5 in crypto appendix[27]

[26] Fix this

[27] Fix this

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX H A Guide to Availability

Introduction

This appendix presents a discussion of availability, some existing models, and the considerations taken in developing the Availability Criteria. Availability includes criteria for *Containment*, *Robustness*, and *Recovery* services. The services defined by the Availability Criteria are elementary in nature. As the Availability Criteria mature, they will expand with additional services.

In discussing availability there is a definite requirement to distinguish between reliability, the failure of a product to operate as originally constructed, and denial of service, the inability of the product to provide a specified service.

Availability Control Objective

Control objectives provide a basis for developing criteria requirements. The availability control objective was specified in Version 1 of the Canadian Criteria:

> *Products that are used to process or handle classified or other sensitive information must assure the availability of services or data according to the product security policy. Data availability requires that the data be (a) accessible at the expected or specified location; (b) presented in the required or specified form and (c) accessible within an expected or specified time interval.*

Scope

The guidelines established in this appendix apply to products under evaluation with the intention of meeting the Containment service (AC-1 to AC-4), the Robustness service (AR-1 to AR-3), and the Recovery service (AY-1 to AY-3).

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Availability Criteria. Examples are given for illustration purposes only and do not represent the only acceptable method of meeting the Availability Criteria.

Requirements for Availability

Availability is defined as the ability to access a specific resource within a specific time frame as defined by the product's specification. This can be defined in terms of a maximum waiting time[28] (MWT) or a finite waiting time[29] (FWT) by which the user should be granted access. Given a period of time, Δ, which is an "average" estimated time for the system to respond to a request, all requests falling outside Δ are logged as failed access attempts.

[28] Maximum waiting time is defined as the maximum time a user would wait to receive the requested resource. This is a quantitative value.

[29] Finite waiting time is defined as a finite, though unbounded, length of time a user would wait to receive the requested resource. This is not a quantitative value.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Policies & Issues

The greatest problem in writing a set of availability criteria is the lack of documented policies currently in use. As a result, there is a need to survey the literature for policies, analyze the common elements of these policies, and identify the fundamental principles, or control objectives, which define a set of policy enforcement mechanisms.

The German Criteria [3] and Harmonized Criteria [4] define a set of requirements for a system to exhibit some level of availability. Unfortunately, the requirements are poorly defined and vague.

Various authors [1, 2, 5, 6, 7] have examined availability. From these sources one can distill a set of common terms which define basic availability. These terms are defined in Table 9. These terms are grouped into the three services

Availability Terminology	Service
User Authorizations/Allocation to Resources	Containment
Scope of Control	Containment
Access Restrictions	Containment
Recovery	Recovery
Robustness	Robustness
Reliability	Robustness
Fault Tolerance	Fault Tolerance
Granularity of Policy	*(Vendor Issue)*

Table 9 Common Availability Policy Terminology Mapped to the Availability Criteria

as found in the Availability Criteria. Although the services of the Availability Criteria are far from complete, they do represent a minimal set of availability services which vendors can meet with present technology.

Models

There is a definite lack of availability models in the current literature. This is directly related to the lack of documented policies, both corporate and governmental. Those models found in the literature are briefly discussed below.

The common thread found throughout the models presented is that of containment (denial of service), the telephony model being the exception.

Amoroso Model

Amoroso [5] proposes a policy model *prevent(p,c)* for availability based on subject priority and object criticality. This policy is intended to control the degree to which a set of subjects can prevent other subjects from obtaining a critical service. Each subject has an associated attribute called priority and each object has a similar attribute called criticality. Subjects may be users, processes, or some other entity. Objects may be files, directories, records, etc. Subjects have roles (e.g. user, administrator, etc.) and objects are partitioned into criticality classes.

An example of the policy applied to System V/MLS is given in [5].

Amoroso states that the *prevent(p,c)* policy provides a means for evaluating potential denial of service in secure computing systems. Amoroso further states that the System V/MLS example demonstrates that conformance to the *prevent(p,c)* policy does not require any unreasonable mechanisms and that mandatory access controls (MAC) were useful with respect to the policy.

The Amoroso model is an example of a model which conforms to the reference monitor approach and one that primarily addresses containment.

Yu-Gligor Model

Yu and Gligor [1, 2] presented a formal specification and verification method for the prevention of denial of service in the absence of failures and integrity violations. Their discussion includes a formal service model for the prevention of denial of service and uses a specification method based on temporal logic. A finite waiting time (FWT) policy is adopted which consists of three different but mutually-related specifications: fairness policies, simultaneity policies, and user agreements. The fairness policy states that a user will not be blocked forever within a service if that user has many opportunities to make progress. The simultaneity policy states that a user will eventually have all the opportunities needed to make progress within a service provided that a user agreement of that service can be satisfied. The user agreement policy states that invocation sequences of multiple service users must be constrained by analysis of possible invocation sequences that may be generated by any user and specification of the allowed user operations, their order, and the required service invocation sequence that must be obeyed by users outside of the service.

CP-6 Quota System

The CP-6 Operating System [7], developed by Honeywell, as the logical successor to Xerox's CP-V, began development in the late 1970s. CP-V, a highly advanced, online system, traces its ancestry back to the first time sharing systems developed at Berkeley.

The quotas introduced by CP-6 were grouped into *Front End Quotas, User Authorization File Quotas,* and *Resource Limit Quotas.* Tables 10,12, and 11 present the three types of quotas and a brief description of each.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

All of these quotas allowed CP-6 to ensure the availability of resources for all users. By intelligently dividing the resources, the system administrator is capable of ensuring that the system is available most of the time. A determined hacker can still attack the system and make some resources temporarily unavailable. However, the accounting of the system would list this user as being "obnoxious" and would issue a warning on the console. Repeated attempts to subvert the system would result in the system forcibly logging the user off. Repeated offences would cause the system to deactivate the account.

Quota	Description
Logon Timeout	The maximum amount of time before a logon attempt is logged out, in seconds.
Read Timeout	The maximum amount of time before a logged in session is terminated, in seconds, due to lack of user input.

Table 10: Front End Quotas

Quota	Description
LO	The number of allowed printed output pages generated by this user during a job or session.
MaxPrio	Maximum priority that the user may assign a given job.
PDis	The number of granules (1k words) of permanent disk that the user may generate.
TDis	The number of granules (1k words) of temporary disk that the job or session may generate.

Table 11: Resource Limit Quotas

Quota	Description
Max Charges	The maximum charges (calculated in CPU dollars) that the user may charge for a fixed time period, normally a week or month. Once this limit is reached, the user is no longer allowed online access during the current fixed time period.

Table 12: User Authorization File Quotas (Continued . . .)

Quota	Description
Expire	The maximum length of time a file can reside on disk before the file is expired and removed. By default this is set to infinity, for obvious reasons.
Max Memory	Specifies the maximum memory (RAM) that a user may request.
Output Prio	Specifies the default priority of the user's output.
PrioB	Specifies the Base execution priority in the various modes (batch, online, etc). This is used by the scheduler to allocate quanta of time to a user relative to other users on the system.
Quan	Specifies a change in the quantum of time to be assigned to an execution period for this user's job for a specific mode (batch, online, etc.). The value is set in milliseconds. Any positive value is **added** to the default quanta calculated from the prio within the scheduler and used for CPU time.
Time Max	The maximum amount of CPU time allocated for this user.
Resources	Limits are imposed on how many various physical resources may be allocated to a user "exclusive". These devices include tape units, disks, and line printers.
Disk Quota	The user is allocated a specific amount of permanent disk space. This space is specified on a specific volume. Allocating more than the given amount is strictly prohibited.
BatNum	The maximum number of batch jobs this user may have in the queue at any one time

Table 12: User Authorization File Quotas

Telephone System

The telephony industry, as well as many networks [6], use the following methods to ensure near 100% availability.

Perceived Availability Perceived availability is the reaction by the network, or telephony system, to an error with an appropriate action to correct the error and ensure "100% availability". The network attains perceived availability by ensuring that mean time to recover (MTTREC), and mean time to repair (MTTR) are kept to a minimum while the mean time between failures (MTBF) is kept to a maximum.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

For the mean time to recover (MTTREC) to be minimized, the telephony industry uses redundant components which are capable of assuming network functions without perceptible delay. MTTREC is closely coupled with the mean time to repair (MTTREP). It is essential that when a component fails, that the system know how the component failed and where the failed component *is*. This ensures that the MTTREP is kept to a minimum. With telephony systems and many modern networks, boards and subsystems are removed from a live, online system. Users of the system never realize that a repair has taken place.

This is accomplished in two ways: i), failures are reduced to the maximum extent possible and; ii) the failures are kept as isolated as possible.

Conditioning The carrier must ensure the quality of the lines used for telephony or network applications. For this, various vendors supply conditioning on their lines. The lines are examined in various ways and thresholds are established. The vendor then limits the use of the lines to these specific thresholds. This ensures that transmission loss is kept to a minimum thus minimizing the error recovery techniques required.

Testing and Monitoring The majority of communications networks have a network management centre (NMC). This centre's job is to ensure the reliability and efficient operation of the network. The NMC takes care of the day-to-day preventative maintenance operations. A wide variety of tools are used to diagnose and correct network problems/faults to provide a perceived availability of 100%.

Common Elements Of The Models

All of the models restrict resources and require a central monitor. This falls in line with the trusted computing base concept and the underlying reference monitor concept. Applying the common elements of the models to an operating system, Table 13provides a list of resources which would be controlled by the *availability reference monitor*[30].

[30] The *availability reference monitor* would be an integral part of the product's reference monitor.

Resource	Limiting Factor
disk space	Kilobytes or Megabytes
sessions	Max online and batch
	Max idle time
	Max "system" errors
	Max online time/da
output	Max output per session
login	Max attempts
	Max timeout
CPU	Max CPU time/day
Priority	Max/Min priority allowed

Table 13: Generic Quotas

Further availability services could be achieved in hardware. Some simple forms would be:

- Redundancy
- Hot board replacing
- Physical restrictions to system (i.e., a multiprocessor system would allow a user to log into only a specific set of ports and only use a specific CPU).

Availability Reference Monitor

The availability reference monitor enforces a specified policy for allocating resources to authorized users. As with the reference monitor concept, the availability reference monitor must be:

- tamper proof;
- **always** invoked;
- designed provably complete; and
- noncircumventable.

Fault Tolerance

A standard method of ensuring availability of a component or system is to introduce a high level of fault tolerance into the product. This allows the product to withstand and correct errors up to a predefined threshold. Once this threshold is breached, the product is considered to have failed.

Hardware fault tolerance techniques could be applied to software. The availability monitor could contain thresholds (fault tolerance levels) for such things as failed login attempts, CPU usage, disk space usage, etc. Prior to these thresholds being reached, the system would consider itself in a "fault tolerant state". Once this threshold had been passed, the system would consider itself in a "failure state" and would log all applications and users currently on the system and all resource/service allocations. Normal thresholds might be:

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

0 – 80% usage	Normal system utilization	2
80 – 90% usage	Fault tolerant state	3
90 – 100% usage	Failure state	4

The numbers above represent normal system response thresholds. Once a system breaches each of the above percentages, the response time of the system (or a particular component) diminishes noticeably.

Containment

Security Policy

The Containment criteria do not require a specific policy. The security policy can be any policy which satisfies the containment criteria **provided that the policy has been approved by the Evaluation Authority.** The security policy must include the following information:

- a list of objects to be protected;
- the mechanism implementing the policy along with supporting administrative tools;
- the granularity of the security policy with respect to the object space of the product;
- the means of containment per object type (i.e., quotas, restricted terminal access, etc.); and
- the failure states and the resultant actions of the trusted product.

Meeting the Criteria

Containment can most easily be viewed as the restricted access to the resources available on a given product. The most common method of containment is the use of quotas. All protected objects within the product (e.g., disk space, logon sessions, cpu time, etc.) must be identified and controlled by the reference monitor[31].

AC-1: Quotas The weakest form of containment is simple quotas. Its purpose is to place upon a user a restriction as to the maximum amount of any given resource that that user can obtain. With this level of service there is no guarantee that a user cannot hoard resources thus restricting another user from accessing those same resources.

[31] The reference monitor concept should be used unless the vendor can produce evidence that another method is equally valid. The Evaluation Authority is open to alternate methods of mediation beyond the reference monitor concept.

AC-2: Denial of Service This is a stronger form of quotas. These quotas allow the restrictions to be applied in such a way as to ensure that no one user can hoard the product's resources. Although intelligently administered quotas of type AC-1 could provide a similar result, AC-2 adds to this by allowing the product to allocate *less* than the user's maximum quota thus insuring another user access to those shared resources.

An example would be the capability of the system to restrict the number of simultaneous batch jobs to a number less than the user's quota if the system were heavily used at a given time.

AC-3: Resource Restrictions This form of containment is a special case in which users are grouped by the administer so as to define prioritized groups to the product. Thus, the product could enter a state whereby only the highest priority users could have access to the system at the expense of other users.

An example of such a case would be a computer used by the military. During a time of war, nonessential personnel would be disallowed logon access to the product thus ensuring maximum compute power for the task at hand. Other uses can be envisioned, such as products which enter degraded service due to hardware failures but retain essential services at the expense of nonessential services.

Fault Tolerance

Security Policy

The Fault Tolerance criteria do not require a specific policy. The security policy can be any policy which satisfies the robustness criteria **provided that the policy has been approved by the Evaluation Authority.** The security policy must include the following information:

- a list of components deemed fault tolerant;
- the mechanism or mechanisms employed; and
- the failure states and the resultant actions of the trusted product.

Meeting the Criteria

Fault Tolerance, as defined by the criteria, is the ability of the product to continue operation while specific components are replaced. The security policy must define which components of the product fall under the fault tolerance criteria. The fault tolerance criteria do **not** apply solely to hardware components.

AF-1: Limited Hot Replacement The vendor must indicate which components of the product are to be listed under these criteria. A service discontinuity within the product must not during the replacement of any of the listed components. The product shall be able to withstand limited failures of components with a possible degradation of service. The failure of a protected component shall never allow the trusted product to enter a state whereby its security policy would become compromised.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

A Guide to Availability

AF-2: Hot Replacement AF-2 requires that **all** of the product's components be replaceable while the system is active. The replacement of any component within the product will not result in a service discontinuity. The product shall be able to withstand limited failures of components with a possible degradation of service. This shall hold true for all components. If a failed component enters a degraded level of service, this shall be indicated by the vendor. The failure of a protected component shall never allow the trusted product to enter a state whereby its security policy would become compromised.

It is well understood by the Evaluation Authority that there may be some components within the trusted product that are not fault tolerant. These shall be so indicated.

Robustness
Security Policy

The Robustness criteria do not require a specific policy. The security policy can be any policy which satisfies the robustness criteria **provided that the policy has been approved by the Evaluation Authority.** The security policy must include the following information:

- a list of components to be protected;
- the mechanism employed; and
- the failure states and the resultant actions of the trusted product.

Meeting the Criteria

Robustness is the ability of the product to withstand component failures. This service is also known as fault tolerance in some circles. The security policy must define which components of the product fall under the robustness criteria. The robustness criteria do **not** apply solely to hardware components.

AR-1: Reliability under Limited Failure The vendor must indicate which components of the product are to be listed under these criteria. Any components which are listed should have a threshold defined below which it shall offer complete service. If the component exceeds its given threshold, the administrator shall be notified. The product will, at worst, enter a state of degraded service. Exceeding the threshold of a given component shall never allow the trusted product to enter a state whereby its security policy would become compromised.

AR-2: Reliability with Degraded Service AR-2 requires that **all** of the components be protected. If a component exceeds its given threshold, the administrator shall be notified. The product will, at worst, enter a state of degraded service. Exceeding the threshold of a given component shall never allow the trusted product to enter a state whereby its security policy would become compromised.

It is well understood by the Evaluation Authority that there may be some components within the trusted product that are not robust. These should be so indicated.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Recovery

AR-3: Reliability with Full Service The vendor must protect all components. No component that exceeds its given threshold shall cause a loss of service nor a service degradation. Exceeding a component's threshold shall result in notification being sent to the administrator. Exceeding the threshold of a given component shall never all the trusted product to enter a state whereby its security policy would become compromised.

The Recovery criteria do not require a specific policy. The security policy can be any policy which satisfies the recovery criteria **provided that the policy has been approved by the Evaluation Authority.** The security policy must indicate the method of recovery for the trusted product.

There are three types of recovery: manual, automated, and selective.

Manual recovery requires the intervention of the administrator or operator.

Automated is fully automated but if the product cannot return to a trusted state, the product will await operator intervention.

Selective recovery will allow for the trusted product to return to a trusted state but at a possibly degraded level of service. If the product does enter a degraded level of service, for whatever reason, the administrator shall be notified immediately. The components causing the degraded service shall be placed offline. These components shall be logged as offline and the administrator so notified.

All recoveries shall be logged and the administrator notified as to the time and cause.

References

[1] Gligor, Virgil, D; "A Note on the Denial of Service Problem", Proceedings of the 1983 Symposium on Security and Privacy, pp. 139-149.

[2] Yu, Che-Fn, and Gligor, Virgil, D; "A Formal Specification and Verification Method for the Prevention of Denial of Service", Proceedings of the 1988 Symposium on Security and Privacy, pp 187-202.

[3] *IT-Security Criteria: Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems*, 1st Version 1989, German Information Security Agency (GISA).

[4] *Information Technology Security Evaluation Criteria (ITSEC), Harmonised Criteria of France, Germany, the Netherlands, and the United Kingdom*, June 1991, Version 1.2 (Provisional).

[5] Amoroso, E., "A Policy Model for Denial of Service". Proceedings of the Computer Security Foundations Workshop III, Franconia, New Hampshire, June 12–14, 1990.

[6] Black, Uyless D., *Data Communications, Networks and Distributed Processing*. Reston Publishing Co. (Prentice-Hall), Reston, Virginia. 1983.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

[7] *The CP-6 System Manager Handbook.* CP-6 Operating System Version C00. Honeywell, 1985.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX **I**

A Guide to Accountability

Introduction

This appendix provides a general guide to understanding Accountability in the Canadian Criteria. Accountability is the attribution of responsibility for an action to the proper individual. Accountability includes criteria for *Audit*, *Identification & Authentication*, and *Trusted Path* services. This appendix provides guidance on meeting the criteria for these Accountability services, and discusses considerations for their proper use and administration.

Scope

The guidelines established in this appendix apply to products under evaluation with the intention of meeting the Audit criteria (WA-1 to WA-4) or the Identification & Authentication criteria (WI-1 to WI-2).

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

Audit

The Audit component of Accountability is responsible for monitoring and detection of real or potential security policy violations. In addition, audit services monitor the use of authorized privileges and act as a deterrent against potential security policy violations. Auditing is the process of recognizing, recording and analyzing security relevant actions and events.

The criteria rate monitoring and detection services based on the granularity of the audit mechanism, the depth of audit file analysis possible, and the degree to which active monitoring is used. These services detect real or potential violations of the product security policy and monitor user activity.

General Concerns

Effective Auditing. Only authorized users are permitted to use the audit review or analysis tools, set options, or select events for the event recording mechanism. The authorized user will usually be a system administrator or security administrator. In environments where a higher degree of separation of duties and least privilege is required, audit administrators may be used to perform all security auditing tasks exclusively.

The administrator responsible for the setting of auditable event options should be aware of the alternatives in capturing information from failed login attempts. If

authentication information (e.g., a password) is captured in the audit record, there exists some possibility for compromise if the reason for failure was an unknown login identifier but the password was correct. The capture of login identifier information is potentially compromising as well, since a valid password may be typed inadvertently, instead of the login identifier. It is the responsibility of the administrator to consider the security needs of the system and determine the appropriate level of information to be captured.

Physical Storage of Audit Data. The choice of the physical medium used for the storage of audit data should be take into consideration the operational usage or performance load, and the required audit granularity. Any forwarding of audit data (e.g., for off-line processing) must be done in a manner which will ensure the security of the audit data. Consideration should also be given to planning an archive for audit data.

Notification to the administrator should be provided when audit data files are approaching maximum capacity. Sufficient advance warning should be given if any manual intervention to perform maintenance is required. The TCB must prevent the performance of any auditable actions when audit files are full, so that no action can escape auditing. This may involve causing a shutdown when audit data files are filled. The shutdown should be graceful with the appropriate alarm given, and an audit event (e.g., "audit data files full — system shutdown") record written.

Meeting the Criteria

Audit services are rated according to their ability to detect real or potential violations of the product security policy and monitor user activity. "After-the-fact" detection methods and mechanisms are accepted at WA–1 via basic auditing and the use of audit review tools, while at WA-4, "active" monitoring in the form detailed auditing and real-time intrusion detection is required.

Audit Granularity. The audit mechanism is viewed as a basic building block of monitoring and detection. The Criteria rate audit mechanisms based on their granularity (e.g., number and/or types of auditable events). However, a fixed set of auditable events is not prescribed for any of the audit levels. It is left to the Evaluation Authority to approve an acceptable set of auditable events for a specific product's Security policy and a specific Audit level, and to ensure consistency between evaluations.

For the purpose of the criteria, a **security relevant event** is defined as an event associated with or derived from one of the TCB's security services, or other events which, while not directly related to a security policy-enforcing mechanism, may provide insight into a potential or real security policy violation.

There are two granularities defined for audit mechanisms: **basic auditing** and **detailed auditing**. Basic auditing is defined as the recording of those security relevant events which are directly attributable to security as defined in the product

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

security policy and as implemented in the security services of the TCB. Basic auditing is required as part of WA-1 and WA-2.

Detailed auditing is defined as the recording of those security relevant events which go beyond obvious attempts at violating the product security policy and those events which require analysis, perhaps as an aggregate, for a security policy violation to be detected. Detailed auditing is required at levels WA-3 and WA-4.

Audit File Analysis. Tools are required at all Audit levels to assist in examining audit data. **Audit review tools** are required at WA-1 and WA-2 to complement and support basic auditing, by assisting the administrator in recognizing real or potential security policy violations in the audit trail. At a minimum, review tools must provide the administrator with the ability to view audit data (e.g., more, grep, Emacs for audit data stored on-line; a hardcopy audit trail may be acceptable for some products).

Audit analysis tools are required for WA-3 and WA-4 audit mechanisms in support of detailed auditing. Analysis tools are those which permit more complex scrutiny of the audit trail in order to identify possible security policy violations. Tools meeting the WA-3 or WA-4 requirements must, at a minimum, provide the administrator with the ability to and manipulate audit data (e.g., "sort by file name," SQL-like query).

Selection of Audit Events. The TCB should permit the administrator to select the auditable events to be recorded by the audit mechanism. This may be achieved either through "pre-selection" or "post-selection."

Pre-selection of audit events involves the designation at system initialization of a subset of the product's possible auditable events for collection and recording in the audit trail. By using pre-selection, the administrator may reduce the number of audit events captured, and in turn, the size of the final audit trail. This fine-tuning allows the administrator to capture only those events which are deemed to be of interest, making the job of reviewing the audit trail more manageable. The disadvantage of pre-selection of audit events, however, is that those events not pre-selected cannot be recovered at a later date if the administrator notices unexpected and potentially suspicious activity. Pre-selection of audit events must be controlled by the TCB.

Post-selection of audit events requires the collection of a more comprehensive set of auditable events in the audit trail. Specific audit events are queried, or post-selected, from an already recorded audit trail by the administrator. The advantage of post-selection lies in the flexibility of potential "after-the-fact" analysis. The administrator is not required to focus the search for potential security violations prior to system initialization. The TCB is required to control the interface to the filtering or query mechanism. That is, the TCB must be

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

responsible for accepting administrator requests and inputs, for performing any data compression or expansion, and for returning the results of a request to the administrator. The mechanism used to perform the querying or filtering may not necessarily be part of the TCB.

Active Monitoring. Active monitoring includes alarms triggered by violations (e.g., of privilege) and imminent violations (e.g., numerous login failures) of security policy beginning at WA-2, and real-time intrusion detection analysis beginning at WA-4. Each of these monitoring and detection mechanisms provide an active response for a real or potential security policy violation.

Intrusion detection systems meeting WA-4 must provide real-time analysis capability. Off-line analysis which is performed after the fact is not acceptable at WA-4.

Auditable Events

Figures 19 through 23 list potential auditable events for functional classes of mechanisms providing Confidentiality, Integrity, Availability, or Accountability. The Evaluation Authority is responsible for approving a suitable minimum list of auditable events for the product under evaluation.

CM	set (modify) tag information for user (process, object) set (modify) designation of export/import medium as multi-tag or single-tag set (modify) printable tag
CD	set (modify) tag information for user (process, object) set (modify) designation of export/import medium as multi-tag or single-tag add (remove) user to (from) a discretionary group
CC	exploitation of network protocol information (e.g., header information) covert channel exploitation of frequency of transmission covert channel exploitation of potential covert channel

Figure 19: Confidentiality-related Auditable Events

IM	set (modify) tag information for user (process, object)	2
	set (modify) designation of export/import medium as multi-tag or single-tag	3 4
	set (modify) printable tag	5
ID	set (modify) tag information for user (process, object)	6
	set (modify) designation of export/import medium as multi-tag or single-tag	7 8
	add (remove) user to (from) a discretionary group	9
IS	add (remove) a user	10
	modify user attributes	11
	modify user security attributes	12
IT	alarm condition raised from self test	13

Figure 20: Integrity-related Auditable Events 14

AC	set (modify) resource limits	15
	resource limits exceeded	16
AR	component failure detected	17
AY	system failure detected	18
	automated recovery initiated	19

Figure 21: Availability-related Auditable Events 20

WA	audit event selection (deselection)	21
	audit files 100% (n%) full	22
	audit facility shut down	23
	audit files emptied	24
	defined threshold for failures exceeded	25
	(imminent) security policy violation detected	26
WI	successful (failed) login attempt	27
	authentication information modified	28

Figure 22: Accountability-related Auditable Events 29

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Misc	system date, time changed
	production of printed output
	object creation, deletion
	object opened, closed
	network connection (connection-less association) between two hosts attempted
	network configuration change - removal (addition) of host

Figure 23: Miscellaneous Auditable Events

Identification & Authentication

Authentication

As a fundamental component of Accountability, identification & authentication serves a dual function. First, identification & authentication login services ensure that all individuals gaining access to a computer system are authorized to do so. Secondly, authenticated user identities are required to support or enforce other types of policies. Identification & authentication is also more commonly known as "login."

Proper authentication must be performed by the TCB using a mechanism which requires an individual to provide unique information known only to the real user. Three types of authentication can be used to verify identities:

- something you *know*;
- something you *have*;
- something you *are*.

These may also be used in combination for increased effectiveness against specific threats of unauthorized access. The use of two or more types of authentication provides additional protection if one of these is defeated through, for example, guessing, spoofing, duplication or theft. The use of two mechanisms can dramatically increase the work effort necessary to defeat the identification & authentication service and gain unauthorized access.

"Something you know." Knowledge of a password, a Personal Identification Number (PIN) or other private information are examples of "something you know." This type of authentication can be taken anywhere and is easy to guard. However, the effectiveness of "something you know" is limited by its ease of duplication: it may be relatively easy to learn or guess private information, and no special equipment or capabilities are required to duplicate "something you know."

Passwords are the most widely used authentication mechanism because of their simplicity and their ease of reset if compromised. Passwords are an attractive authentication alternative because the owner is always in full possession; however, the use of passwords requires vigilance to avoid compromise from duplication.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Whatever the implementation, the space of potentially valid authentication data must be sufficiently large to discourage duplication.

"Something you have." Physical objects such as a smart card, a magnetic strip card, a challenge-response generator, electronic keys, or a physical cryptographic ignition key are examples of "something you have." The main advantage of this type of authentication is the difficulty or expense associated with duplication. Physical authentication devices are typically made using special equipment or processes which are not generally available, thereby deterring attempts at forgery.

The effectiveness of a physical object as authentication is constrained by possession; that is, it is not joined to its owner. Loss of authentication devices may be a potential problem for compromise. However, in most cases it may be sufficient to detect only the loss of such a device.

"Something you are." Biometrical measurements such as a fingerprint, retinal or DNA patterns, or hand geometry are examples of "something you are." Given that the goal of authentication is verifying the identity of a user, "something you are" is a direct test of that identity, and as such should provide much stronger authentication than the other two types of authentication.

The major obstacle in using "something you are" is the cost of developing economical devices which can provide acceptable reliability (i.e., can entirely distinguish between two individuals). Further, the use of these relatively expensive authentication techniques does not guarantee unerring performance. Failure rates may be unacceptable for some fingerprint, retinal or voice scanning equipment. Identical twins would not be detected by DNA analysis (and perhaps other physical measurements).

Meeting the Criteria

Each individual desiring access to the trusted product must be identified by the TCB as an authorized user. Identification is initiated by the individual requesting access; the claimed user identity of the individual must be verified by the TCB authentication mechanism(s) before further access to system resources is permitted.

The use of a single type of authentication mechanism is sufficient to satisfy WI-1. This single mechanism must implement one of the three types of authentication discussed above.

WI-2 requires the use of two or more different types of authentication mechanisms. For example, "something you know" may be used in conjunction with "something you have." Whatever the combination, two different *types* of authentication must be used – the use of two instances of the same type does not qualify for WI-2. For example, a password-based mechanism (something you know) and a PIN mechanism (also something you know) cannot be used

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

in combination to satisfy the WI-2 requirements. If more than one authentication mechanism is implemented, the individual requesting access must pass both authentication tests before being granted access.

References

[1] National Computer Security Center. *A Guide to Understanding Audit in Trusted Systems*, NCSC-TG-001 Version 2, 1 June 1988.

[2] National Computer Security Center. *A Guide to Understanding Identification and Authentication in Trusted Systems*, NCSC-TG-017 Version 1, 1 September 1991.

APPENDIX J

A Guide to Assurance

Introduction

This appendix provides a general guide to understanding the Assurance Criteria. The Assurance Criteria include requirements for Architecture, Development Environment, Development Evidence, Security Manuals, Security Testing, and Trusted Distribution & Generation. This appendix discusses assurance issues involved in the design, implementation, and evaluation of trusted products, and provides guidance on meeting the Assurance Criteria requirements.

Scope

The guidelines established in this appendix apply to products under evaluation at all levels of Assurance (T-1 to T-7).

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

Architecture

Development Environment

Life Cycle Process

The development environment of the Vendor is required to include a supporting life cycle process and a configuration management system. These requirements should not force an undue burden on the Vendor. They do, however, provide assurance to the Evaluation Team that the development environment and the product under evaluation is sufficiently controlled.

The Vendor is required to define the life cycle process under which the product is being developed. Given that this process will evolve during the product's life cycle, the Vendor is required to document the known details of each phase and update the document during its evolution. Each phase and its milestone requirements (i.e., requirements to be met before continuing to the next development phase), the use of life-cycle methodology, structured design techniques, and software standards (e.g., coding and in-line documentation standards) must be documented as part of the life cycle process.

The Criteria also requires that programming languages and compilers used during implementation be **well-defined**. A well-defined programming language in this context includes any language meeting an International Standards Organization (ISO) or comparable standard; or, one which is in wide commercial or military use; or, one for which a sufficient language definition can be submitted to the Evaluation Authority for approval (similar extensions to this definition may be added by the Evaluation Authority on a case-by-case basis). In addition, the use

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

of implementation-dependent or hardware-dependent options in the programming language or compiler must be documented. Depending on the assurance level, the development environment may include measures for physical, technical, procedural and personnel security, and these must also be documented.

Configuration Management

Configuration management is a necessary and beneficial part of any development effort, especially for trusted products. Unfortunately, the feared costs rather than the benefits of configuration management are usually identified. Further, the benefits of configuration management may be obvious in a large development project, but less so in a project of smaller size or lesser complexity. However, the benefits of configuration management are the same for both small and large projects; it is only the tools and the level of formality that differ.

The requirements of the Assurance Criteria are intended to be as flexible as possible. The Vendor is encouraged to develop and fine-tune an appropriate configuration management system which best reflects both the complexity of the product and the size of the Vendor's development organization. The level of change authority, the possible use of automated tools, and the appropriate level of formality in procedures and audits are defined by the Vendor (and approved by the Evaluation Team) so as to be as compatible with the rest of the development environment as possible.

Configuration management provides designers and developers with the methods and tools to identify the product at any time during its life cycle, to control changes to the product and to monitor the status of the product or any of its components. Configuration management evidence can also be used as input into the quality assurance process.

The Criteria requirements do not force an undue burden on the Vendor, while providing assurance to the Evaluation Team that the product under evaluation is sufficiently controlled.

Basic Principles. Configuration management entails four primary tasks: identification, change control, status accounting, and audit. Each of these tasks plays a necessary role in an effective configuration management system.

The product must be identified in terms of its configuration items: hardware, firmware, software, and product documentation (e.g., functional specification, architectural and detailed design, testing documents) Each configuration item is assigned a unique and meaningful name that it will retain through the product life cycle. Project-wide conventions for entitling, labeling, numbering and cataloguing configuration items must be used to ensure consistent identification. Special attention should be paid to conventions for software items (e.g., to reflect whether the item is source or object code). Version information must also be maintained so that like-named items remain unique over multiple versions and releases.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

The size of configuration items will necessarily vary according to their complexity and anticipated frequency of change. By carefully selecting the size of individual configuration items, the configuration management system can better isolate those items that will change often from those that will not, isolate those items that are security relevant from those that are not, and group small components into a single configuration item to reduce the overall number of configuration items. The effectiveness of change control is driven by the appropriate designation of configuration items: a balance must be maintained between managing a large number of small configuration items and grouping too many components into one item.

The main function of configuration management is the control of the product and all changes to it. Configuration change control must be exerted over the product during all stages of the product's life cycle. A change control process and a set of procedures must be defined so that proposed changes are submitted to a change control authority for approval. It can then ensured that items are frozen when necessary, and that the effects of proposed changes are assessed before being authorized. In addition to procedures, the Vendor's organizational bodies responsible for configuration management (e.g., a Configuration Control Board; design, development, testing and management staff) must have specific roles and responsibilities defined for them in the control process.

The status accounting task of configuration management involves the recording of status information on configuration items. This includes both original definitions of the item and any changes made to the item (e.g., enhancements, bug fixes). By maintaining a record of each configuration item, the current status of each item can be made available to affected staff, and historical data may be compiled for use during configuration audits.

Configuration auditing is performed by conducting reviews and audits of status accounting records (for example) to ensure that the Vendor's configuration management system is functioning as intended. Further refinement and fine-tuning of the configuration management process can be made possible by first conducting configuration audits. These reviews and audits also ensure that organizational standards, policies and procedures are followed throughout the configuration management system.

Planning A Configuration Management System. A Configuration Management Plan allows the Vendor to define the organization's concept of configuration management and the ways in which the Vendor will address the configuration management tasks. It is through the configuration management plan that the Vendor can develop a system that best complements the development process.

It is important that all procedures, and the roles and responsibilities of all staff involved in configuration management be defined and documented. A written

document prevents disagreement and confusion, and assists in the training of new staff. Procedures are not required to be complex, but they must be documented. The aim is not to stifle the development process, but rather to provide standard methods of operation and allow developers to address real technical problems.

It is important to ensure that an appropriate level of change control authority is used throughout the project. Authority must be defined so that it is both appropriate and timely. Approval authority for changes should be adjusted according to the phase of the development cycle, the nature of the change, and the scope of the change. Depending on these, the appropriate authority may be a designer, a programmer, a design review committee, or the project manager. The approval process and level of approval authority should not become so cumbersome that it seriously inhibits development.

The use of computer-based tools can enhance the effectiveness and timeliness of configuration management tasks. Tools can provide automated software library control for source and object libraries, as well as serve as an archive for different product versions. Tools are useful in performing on-line tracking of configuration item status and changes. Database management systems can also be used for their querying and reporting capabilities.

The configuration plan should address configuration status accounting and define the information to be recorded. The necessary types of configuration audits (and their objectives and frequencies) also need to be defined.

Cost and time delays can be minimized with the application of configuration management techniques and the appropriate selection and use of tools and procedures. These techniques must be adapted and fine-tuned for a project and an organization's development process.

Meeting the Criteria. All products must be maintained under a configuration management system which can effectively control all changes to the product. For the purposes of the Canadian Criteria, a configuration management system consists of both technical and procedural measures. The product's hardware, software, firmware, documentation, associated test suites, etc. must be maintained under the configuration management system. The configuration management requirements increase at T-1, T-4 and T-6 levels of Assurance.

At T-1 to T-3, the Vendor must plan and implement a basic configuration management system which is capable of identifying the evaluated product, controlling changes to the product, and maintaining a history of changes to the product. The configuration management system must have a technical or procedural means of associating documentation with hardware, software, firmware and test suites as required. This mapping may be performed by reference (from one to the other). At these levels of assurance, the configuration management system is not required to use computer-based tools; manual procedures are acceptable.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

At T-4 and T-5, the configuration management system must also be able to generate a version of the TCB from source code, compare it to another version of the TCB, and log any differences. A means for tracking reported bugs or other problems must be included as part of the configuration management system. This mechanism must be able to track a bug or problem report for a given configuration item to its resolution. At these levels of assurance, the configuration management system is required to use computer-based tools to satisfy the criteria; manual procedures are also expected to be used to supplement the automated system.

At T-6 and T-7, the configuration management system must also maintain control over all tools (e.g., programming languages, compilers, runtime libraries, etc.) used in the development of the trusted product.

Development Evidence

The Development Evidence requirements are grouped under three major life-cycle milestones: Functional Specification, Architectural Design, and Detailed Design. Each section details the appropriate style of specification and level of detail required, and the level of formality required for mapping.

Terminology

Specification Style. Three types of specification style are used in the Criteria: "informal," "semiformal" and "formal." The functional specification, architectural design specification, and detailed design specification will be written using one or more of these styles. Therefore, each of these styles will be instantiated in a different fashion (e.g., high level data flows, function calls, design diagrams, input/output) depending upon the intent of the specification.

An **informal** specification is written as prose in natural language. Natural language is used here as meaning communication in any commonly spoken tongue (e.g., English, French). An informal specification written in natural language is not subject to any notational or special restrictions other than those required as ordinary conventions for that language (e.g., grammar and syntax). While no notational restrictions apply, the informal specification is also required to provide defined meanings for terms which are used in a context other than that accepted by normal usage.

A **semiformal** specification is written in a restricted syntax language and is accompanied by supporting explanatory (informal) prose. The restricted syntax language may be a natural language (e.g., restricted sentence structure and keywords with special meanings), or it may be diagrammatic (e.g., data-flow diagrams, state transition diagrams, entity-relationship diagrams, data structure diagrams, and process or program structure diagrams). Whether based on diagrams or natural language, a set of conventions must be supplied (as explanatory informal prose) to define the restrictions placed on the syntax.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

A Guide to Assurance

Structured design methods such as the Yourdon Structured Method [1], Structured Analysis and Design Technique [19], Structured Systems Analysis and Design Method [17], and the Jackson Structured Design [10] and Jackson Structured Programming [11] methods can also provide a basis for semiformal specifications.

A **formal** specification is written in a formal notation based upon well-established mathematical concepts, and is supported by explanatory (informal) text. The concepts are used to define the syntax and semantics of the notation, and the proof rules supporting logical reasoning. Critical security properties must be expressible in the formal specification language. The critical security properties must be shown to be satisfied by the formal specification. Where hierarchical levels of specification exist, it must be possible to demonstrate that each level maintains the properties established for the previous level.

The syntactic and semantic rules supporting a formal notation used in a security policy model shall define how to recognize constructs unambiguously and determine their meaning. Where proof rules are used (at T-6 and T-7, the formal language must have an associated proof theory) to support logical reasoning, there shall be evidence that it is impossible to derive contradictions. All rules supporting the notation shall be defined or referenced. All constructs used in a formal specification shall be completely described by the supporting rules. The formal notation shall allow the specification of both the effect of a function and all exceptional or error conditions associated with that function.

Example formal notations are Verdi [18], VDM [12], Z [20], Ina Jo [21], the Gypsy Specification Language [5], and the ISO protocol specification languages LOTOS [9] and Estelle [8]. The use of constructs from predicate (or other) logic and set theory as a formal notation is acceptable, provided that the conventions (supporting rules) are documented or referenced (as set out above).

Level of Detail. At each level, development evidence is required to be submitted to the Evaluation Team. The expected level or amount of detail is characterized in the Criteria by the use of “state,” “describe” or “explain” in detailing the requirement.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Functional Specification	An outline of the Security Policy and of the functionality of the product
Architectural Design	High level software, firmware and hardware structure diagrams (e.g., Hierarchical Software Diagrams, block diagrams) and a brief statement of the elements at each level
Detailed Design	An itemization of programs with name, inputs, outputs, purpose

Figure 24: "Stated" Evidence

When the Vendor is required to **state** product evidence, the submitted document must provide sufficient detail to provide basic understanding. Stated evidence for a Functional Specification, an Architectural Design and a Detailed Design is illustrated in Figure 24.

Functional Specification	A detailed account of the Security Policy and the functionality of the product; an itemization of the abstract rules or mechanisms in the Security Policy Model used to enforce the Security Policy
Architectural Design	High level software, firmware and hardware structure diagrams (e.g., Hierarchical Structure Diagrams, block diagrams) and an itemization of the programs and procedures
Detailed Design	An itemization of programs and procedures with name, inputs, outputs, purpose; brief description of important data structures and algorithms

Figure 25: "Described" Evidence

When the Vendor is required to **describe** product evidence, the submitted document must provide enough detail so that its meaning is evident to the evaluators and justification is usually not necessary. Described evidence for a Functional Specification, an Architectural Design and a Detailed Design is illustrated in Figure 25.

Functional Specification	A detailed account of the Security Policy and the functionality of the product; an itemization of the abstract rules or mechanisms in the Security Policy Model used to enforce the Security Policy
Architectural Design	High level software, firmware and hardware structure diagrams (e.g., Hierarchical Structure Diagrams, block diagrams) and an itemization of the programs and procedures, together with an explanation of their interaction
Detailed Design	An itemization of programs and procedures with name, inputs, outputs, purpose; detailed explanation of important data structures and algorithms; if the validity of an algorithm is not evident to the Evaluation Team, the Vendor is required to provide a justification of its sufficiency

Figure 26: "Explained" Evidence

When the Vendor is required to **explain** product evidence, the submitted document must provide enough detail so that its meaning is evident to the evaluators, and the burden of proof is placed on the vendor. Explained evidence for a Functional Specification, an Architectural Design and a Detailed Design is illustrated in Figure 26.

Mapping Requirements. Since the Functional Specification, Architectural Design and Detailed Design represent milestones in the development of the product, significant assurance can be gained by ensuring that the product can be traced through each phase of development. The Assurance Criteria, therefore, include mapping requirements to ensure a complete mapping between these development milestones. The level of effort required to satisfy the mapping requirements increases with the Assurance levels. The mapping requirements use the terms "trace," "demonstrate" or "prove" to characterize the necessary effort.

When the Vendor is required to **trace** the complete mapping between development milestones, only a basic correspondence is required between elements of each document. Pointers and references to other documents are acceptable when a complete mapping is traced.

When the Vendor is required to **demonstrate** the complete mapping between development milestones, a correspondence is required between more detailed elements of each document. Pointers and references to other documents are acceptable. The burden of proof is placed on the Vendor when a complete mapping is demonstrated.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Functional Specification

When the Vendor is required to **prove** the complete mapping between development milestones, a correspondence is required between more detailed elements of each document. Pointers and references to other documents are acceptable. The correspondence between elements is now required to be expressed formally. The burden of proof is placed on the Vendor when a complete mapping is proven.

The Functional Specification is a functional description (i.e., free of any Architectural or Detailed Design information) of the security services provided by the product. It consists of a **Security Policy**, and beginning at T-2, a **Security Policy Model**. The Security Policy must describe the product's security services in terms of Criteria functional classes (and potential levels if possible). The Security Policy Model provides a mathematical model of the Security Policy.

For example, a Security Policy could be defined under headings describing security services meeting the following:

- Mandatory Confidentiality: CM-1
- Object Reuse: CR-1
- Separation of Duties: IS-1
- Identification & Authentication: WI-1
- Audit: WA-1.

The Security Policy Model would then provide a mathematical model for each of the five components of this Security Policy.

Also, it should be noted that for policies including separation of duties at IS-1 or higher, all security administration duties (for CM-1, WI-1, and WA-1 in the example above) may be modelled either distinctly under separation of duties, or within each of the model parts for the other security services (i.e., CM-1, WI-1, and WA-1).

Beginning at T-2, the Vendor is also required to map the components of the Security Policy to the Security Policy Model, and vice-versa. That is, the mapping must show that the Security Policy is completely represented in the Security Policy Model, and that the Security Policy Model models the entire Security Policy.

Architectural Design

The Architectural Design is a high-level or top level design that further defines the Functional Specification of the product. For example, block diagrams, high level procedure calls or program names are acceptable as part of an Architectural Design.

As in the case of the Functional Specification, beginning at T-2 the mapping requirements for the Architectural Design are intended to show that the Architectural Design completely implements the Security Policy Model and only the Security Policy Model.

Detailed Design

The Detailed Design is a low level or detailed description of the product. For example, pseudo-code or other low level design technique can be used. The Detailed Design should be of sufficient detail to allow full implementation of the product using only the Detailed Design document.

At most assurance levels, the Detailed Design must be completely mapped to the Implementation, and also to the Security Policy Model and Architectural Design at higher assurance levels.

Security Manuals

The Vendor is required to provide a Security Features User's Guide and a Trusted Facility Manual for the product as part of the evaluation. These manuals must address security issues for general users and administrators. The information contained in these manuals is dependent upon by the Security Policy of the evaluated product. These manuals should, therefore, address features, options, cautions and concerns relevant to the specific Security Policy enforced by the product.

Documentation describing the security features of the evaluated product must be provided for all product users by the Vendor. This documentation must be directed both at the general user and at the user responsible for security administration.

The Security Features User's Guide is required to describe to the general user the full functionality of the security services offered by the evaluated product. The Trusted Facility Manual provides assistance to the user responsible for security administration in the proper maintenance of security.

No specific format is required for these manuals. The manual requirement may be satisfied if it is contained in either a separate document or a separate section or chapter of an existing manual addressed to the general user (or administrator). It is required, however, that these manuals be presented as physically separate documents, as they are intended for different audiences.

Security Testing

Security testing is a key element of the Assurance Criteria. Four levels of security testing have been defined: **basic**, **detailed**, **extensive**, and **thorough** security testing. The level and scope of effort required to satisfy each type of security testing is defined below.

Basic security testing is required for T-1 Assurance and is defined by the following:

- The search for flaws shall be limited to those which are obvious to the Evaluation Team.
- Design documentation and knowledge of common flaws may be used as the basis for flaw hypothesis testing.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Detailed security testing is required for T-2 Assurance and is defined by the following:

- The search for flaws shall **include all design and implementation flaws, using limited penetration testing**.
- Design documentation and knowledge of common flaws may be used as the basis for flaw hypothesis testing.

Extensive security testing is required for T-3 Assurance and is defined by the following:

- The search for flaws shall include all design and implementation flaws, using limited penetration testing.
- Design documentation**, source code and object code** and knowledge of common flaws may be used as the basis for flaw hypothesis testing.

Thorough security testing is required for T-4 to T-7 Assurance and is defined by the following:

- The search for flaws shall include all design and implementation flaws, using **thorough** penetration testing.
- Design documentation, source code and object code and knowledge of common flaws may be used as the basis for flaw hypothesis testing.

Security testing beginning at T-4 should ensure that the TCB implementation works as intended by the Detailed Design. Beginning at T-6, the demonstrated mapping from the Architectural Design and Detailed Design to the source code is suggested for use as a basis for planning penetration testing.

Trusted Distribution & Generation

It is important that a product not be tampered with during its distribution and delivery to the customer. As well, TCB generation, start-up and initialization must be done in a secure manner to ensure that the product begins in a secure state.

References

[1] A.J. Bowles. *A Note on Yourdon Structured Method* In *ACM SIGSOFT Software Engineering Notes*, Vol. 15 No.2, April 1990, p.27.

[2] J.K. Buckle. *Software Configuration Management*, Macmillan Education, 1982.

[3] Commission of the European Communities. *Information Technology Security Evaluation Criteria (ITSEC)*, Office For Official Publications of the European Communities, L-2985 Luxembourg, CD-71–91–502–EN-C, Version 1.2, June 1991.

[4] Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28–STD, December 1985.

[5] D.I. Good, R.L. Akers, L.M. Smith. *Report on Gypsy 2.05*, report ICSCA-CMP-48, University of Texas at Austin, February 1986.

[6] IEEE. *ANSI/IEEE STD 828–1983 Software Configuration Management Plans* In *Software Engineering Standards*. The Institute of Electrical and Electronic Engineers Inc., 3rd edition, 1989.

[7] IEEE. *ANSI/IEEE STD 1042–1987 Guide to Software Configuration Management* In *Software Engineering Standards*. The Institute of Electrical and Electronic Engineers Inc., 3rd edition, 1989.

[8] ISO. *Estelle: A Formal Description Technique Based on an Extended State Transition Model*, DIS 9704, July 1987.

[9] ISO. *ISO/IEC-JTC1/SC21/WG1/FDT/C Information Processing Systems – Open Systems Interconnection – LOTOS, a Formal Description Technique Based on the Temporal Ordering of Observational Behaviour*, IS 8807, February 1989.

[10] M.A. Jackson. *System Development*, Prentice-Hall International, 1983.

[11] M.A. Jackson. *Principles of Program Design*, Academic Press, 1975.

[12] C.B. Jones. *Systematic Software Development Using VDM*, Prentice-Hall International, 1990.

[13] National Computer Security Center. *A Guide to Understanding Configuration Management in Trusted Systems*, NCSC-TG-006 Version 1, 28 March 1988.

[14] National Computer Security Center. *A Guide to Understanding Design Documentation in Trusted Systems*, NCSC-TG-007 Version 1, 2 October 1988.

[15] National Computer Security Center. *A Guide to Understanding Trusted Distribution in Trusted Systems*, NCSC-TG-008 Version 1, 15 December 1988.

[16] National Computer Security Center. *A Guide to Writing the Security Features User's Guide For Trusted Systems*, NCSC-TG-026 Version 1, 1 September 1991.

[17] National Computing Centre Limited. *The SSADM Manual*, ISBN 085–012–527–X, Manchester, UK, 1989.

[18] Odyssey Research Associates, Inc. *Reference Manual for the Language Verdi*, Technical Report TR-90–5429–09, 265 Carling Ave., Ottawa, ON K1S 2E1, Canada, February 1990.

[19] SofTech, Inc. *An Introduction to SADT Structured Analysis and Design Technique*, Report 9022–78R, 460 Totten Pond Road, Waltham MA 02154, USA, November 1976.

[20] J.M. Spivey. *The Z Notation: A Reference Manual*, Prentice-Hall International 1989.

[21] Unisys Corporation. *The Ina Jo Specification Language Reference Manual*, Culver City CA, USA, 1989.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

APPENDIX K

Implementing Services via Cryptography

Introduction

This appendix provides guidance on the use of cryptography in the implementation of specific services. Actual requirements as to the manufacture and implementation of cryptographic devices can be found in the document *Cryptographic Modules.* The *Cryptographic Modules*[32] [6] document is a companion to the Canadian Criteria and discusses cryptographic issues in trusted products.

Cryptographic modules are intended to provide the following classes of service: encryption; the generation of digital signatures; key distribution; key generation; and/or key management. These services may then be used by the TCB to implement various security services. A discussion of which services can be implemented via cryptography is provided.

A cryptographic module is considered to be a trusted component which works in conjunction with a TCB. The integration of the cryptographic module and the TCB are discussed in *Cryptographic Modules* [6], and are briefly outlined in the "Integration" section of this appendix; implementation details are deferred to the *Cryptographic Modules* document.

Scope

The guidelines established in this appendix apply to products under evaluation with the intention of using cryptographic means to implement a service, or services, found in the: Confidentiality Criteria, Integrity Criteria, Availability Criteria, or Accountability Criteria.

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

[32] *Cryptographic Modules* uses *Security Requirements for Cryptographic Modules* (FIPS 140–1) [1] as a base.

Export Controls

If cryptography is to be used for secure communications the development could require endorsement under the Cryptographic Endorsement and Assessment Program (CEAP). The developer is encouraged to contact the TPEP program manager to discuss the possibility of a combined rating under the TPEP and endorsement under the CEAP with a single submission.

Depending on the cryptography used, export control restrictions may be placed on the product. This would have to be explored separately by contacting the Department of External Affairs:

Department of External Affairs
International Trade Canada
Export Control Division (ESE)
Lester B. Pearson Building
125 Sussex Drive, C-4
Ottawa, Canada
K1A 0G2
Telephone: (613) 996–2387

Advice concerning the technical aspects relating to export may be acquired by contacting the Communications Security Establishment, Director, Evaluation and Engineering Group:

Communications Security Establishment
Director, Evaluation and Engineering Group
P.O. Box 9703 Terminal
Ottawa, Canada
K1G 3Z4
Telephone: (613) 991–7174

If the vendor intends to use cryptographic components and modules approved for applications within the Commercial COMSEC Endorsement Program (CCEP) the product development becomes classified and is therefore not covered by this appendix nor by the *Cryptographic Modules* document. The vendor should contact the TPEP Program Manager to discuss the full ramifications of such a product development.

Scope

Cryptography can be used to implement the services and corresponding levels of service found in Table 14. Table 14 represents those levels which CSE acknowledges as being implementable via cryptographic means. They do not, necessarily, represent the complete set of services which can be implemented via cryptography.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Service	Applicable Levels
Discretionary Confidentiality	CD-1 to CD-3
Mandatory Confidentiality	CM-1 to CM-2
Object Reuse	CR-1
Identification & Authentication	WI-1 to WI-3
Trusted Path	WT-1 to WT-2
Discretionary Integrity	ID-1 to ID-3
Mandatory Integrity	IM-1 to IM-2
Separation of Duties	IS-3

Table 14 Levels Affected by Cryptography

Although cryptography can be used to implement the security functions listed above, the use of cryptography alone may not be sufficient to fully meet the criteria.

Confidential export is the term used in this appendix to denote the encapsulation of sensitive data by encryption so that it may be transmitted or stored on electronic media which would otherwise be unsuitable for sensitive data. File encryption is an example of confidential export when it is desired that the device containing the encrypted data be treated as nonsensitive. Implementing confidential export entails additional requirements on the integration of the cryptographic modules. These requirements are designed to ensure that sensitive information not "leak" onto nonsensitive media.

Security Policy

By the very nature of cryptography, a cryptographic module provides its services by using private information to transform data so that the transformation cannot be reversed without some knowledge of the private information. Therefore, the secure operation of a cryptographic module depends on the confidentiality and the integrity of this private information. The secure operation of encryption further requires that the cryptographic module prevent the disclosure of sensitive information except to authorized objects. These considerations lead to the following high-level functional security objectives for the cryptographic module:

1. To prevent the unauthorized disclosure of the non-public contents of the cryptographic module.
2. To prevent the unauthorized disclosure of secret cryptographic keys and other critical security parameters.
3. To employ cryptographic algorithms approved for protecting designated information.
4. To prevent unauthorized operations on or the unauthorized use of the cryptographic module.

5. To protect cryptographic keys against unauthorized substitution, insertion, deletion and use.
6. To ensure the proper operation of the cryptographic module.
7. To prevent the unauthorized and undetected modification of the cryptographic module.
8. To detect errors in the operation of the cryptographic module and to prevent the compromise of sensitive data and critical security parameters as a result of those errors.
9. To provide indications of the operational state of the cryptographic module.

Discussion of the implementation details of the above objectives can be found in *Cryptographic Modules.*

Implementing Services

Outlined below are the services that can be implemented via cryptographic means. They do not represent a complete set. It may be possible for other services to be implemented via cryptography, however, the Vendor must provide sufficient evidence to convince the evaluation authority as to the legitimacy of the claims. The methods described should not be taken as the only method of implementing the indicated services.

The implementation of a service via cryptography may not be complete without supporting software within the TCB. The Evaluation Authority will notify the vendor if an implementation of a service via cryptography alone is insufficient. Two services which may require supporting TCB software are discretionary confidentiality and discretionary integrity.

Discretionary Confidentiality

Implementing discretionary confidentiality via cryptography is much more difficult than implementing mandatory confidentiality. The reason for this is simple: a users must be able to modify their own environment allowing or disallowing other users, groups of users, or processes access to objects in their domain.

Simply passing the cryptographic key to various users does not suffice as once a key is given to another user, it can never be retracted. Discretionary controls must provide the user with the ability to add or remove permissions. The TCB should be written to allow for the owner of an object to list those users (or processes) who can access an object. Figure 27 illustrates a case where the object is actually a front for the information being requested. The “object proper” is encrypted using the “key” stored in “Tag Information”. For the TCB to grant a user access to the object it checks whether that user is in the tag information list. If the user is, then the key is used to unlock the “object proper”. Otherwise, the TCB refuses to grant access to the object. With cryptography, the “object proper” need not be protected by the TCB *but* the “tag information” must be protected. It can be protected by encryption as well, perhaps via an unknown (except to the TCB) key.

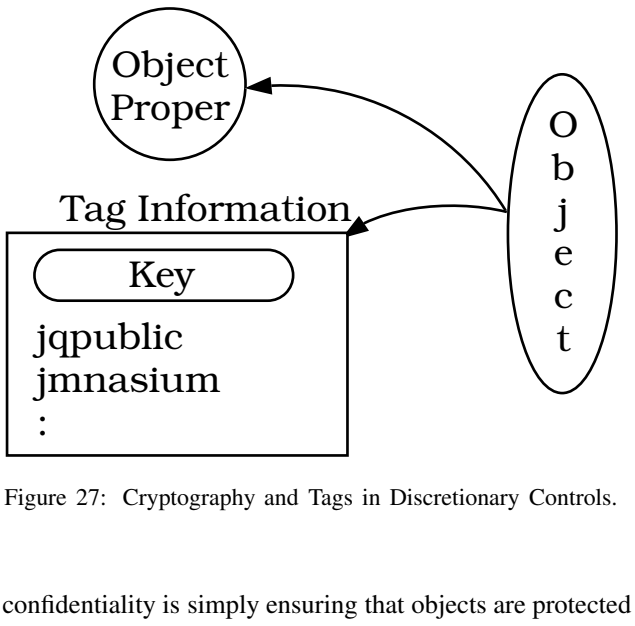

Figure 27: Cryptography and Tags in Discretionary Controls.

Mandatory Confidentiality

Mandatory confidentiality is simply ensuring that objects are protected according to a mandatory policy. A product which creates separately encrypted areas for use by users at different levels of sensitivity, ensuring that data from one group remains unreadable to another, is capable of implementing mandatory confidentiality. Further, providing each user with a unique key by which data is encrypted allows for the creation of user domains which meets the lower service level requirements for mandatory confidentiality.

Object Reuse

Object reuse can be implemented by having the cryptographic device within the TCB encrypt, with a protected system generated key, objects prior to reallocation. The use of cryptography for object reuse is dependant upon the sensitivity of the information being stored. The Evaluation Authority reserves the right to restrict the use of a product which uses cryptography for object reuse to specific applications. To handle certain sensitive of information or ranges of sensitive information may require the use of high grade cryptographic modules.

Identification & Authentication

A cryptographic device can be used for both identification and authentication. On many computer systems, authentication is already carried out by cryptographic means. However, identification can also be accomplished by using an offline device to identify *and* authenticate oneself to the TCB.

Cryptographic means could also be employed to uniquely identify data as belonging to a specific user, so long as the private key employed by the user remained secret.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Trusted Path

Trusted path is the guarantee that the TCB is talking directly to the user and that the user is talking directly to the TCB. By use of cryptography, one can guarantee that no-one will be able to spoof or forge messages from the user to the TCB. Message authentication codes are one possible implementation mechanism that can be used to guarantee that the user and TCB are speaking with one another. The issue of replay remains a concern, however.

Discretionary Integrity

This is very similar to discretionary confidentiality, above. However, the onus would be placed on ensuring that appropriate processes are used in the manipulation of objects.

Mandatory Integrity

As with mandatory confidentiality, mandatory integrity could be implemented by ensuring that an object can only be "opened" by a process or user/process pair having the appropriate keys.

In the case of IM-1, the user object would have access to keys capable of decrypting the process and object. For IM-2, the process must have a key which can unlock passive objects. Even though anyone can start up the process, the passive object containing the data can only be manipulated by the process with the proper decryption key.

A simple examination of object header information by the TCB would allow the TCB to determine whether or not the decrypt was successful.

Separation of Duties

Cryptographic means can be used to ensure separation of duties. If a user requires a set of privileges to perform a specific task, those privileges can be defined as a series of keys, each of which must be used prior to allowing the user to access information for a specific role.

For example, the roles *operator, security officer,* and *system administrator* could be defined in terms of privileges. Commands associated with a given role could only be activated by decrypting the command with the keys found in the associated privileges. Thus, $Operator_{console}(command)$ would decrypt *command* with the *console* key for the operator. If the security officer attempted such an access, it would be denied if the security officer did not have the *console* privilege. Similarly, for the operator to perform a backup operation it may require the following: $Operator_{backup}(Operator_{console}(command))$. If the operator was not granted backup privilege, the operation would fail. The advantage of such a scheme is that the *command* could not be successfully executed until completely decrypted by using all the required keys. The number of keys (and associated privileges) may be unknown for greater security.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Integration

This appendix distinguishes four methods by which a cryptographic module can work in conjunction with a TCB (see Table 15 and Figures 28 and 29): inline stand-alone, inline embedded, offline stand-alone, offline embedded.

A cryptographic module is considered to be *inline* if the export media can be accessed only through the cipher text interface of the cryptographic module; for example, a cryptographic module which implements its own communications controller would be considered inline for devices connected to that controller. A cryptographic module is considered to be *offline* if the cipher text interface communicates directly with the TCB.

A cryptographic module is considered to be *stand-alone* if it provides its own physical security and if all security critical control information (e.g. identification and authentication) is input by an interface which is external to the host; if all security critical status information (e.g., audit and alarms) is output by an interface external to the host; and if key data is input only through an interface external to the host. Critical interaction between a user and a stand alone cryptographic modules are not mediated by a host system. A cryptographic module is *embedded* if it relies on the (host) TCB to mediate interaction with a user for critical portions of the control and status interface.

	Stand-Alone	Embedded
Inline	• Critical portions of the control and status interface do not depend on host • Cipher text interface connected directly to external media	• Critical portions of the control and status interface depend on host • Cipher text interface connected directly to external media
Offline	• Critical portions of the control and status interface do not depend on host • Cipher text interface connected to (host) TCB	• Critical portions of the control and status interface depend on host • Cipher text interface connected to (host) TCB

Table 15 Configurations of Cryptographic Modules within a Host

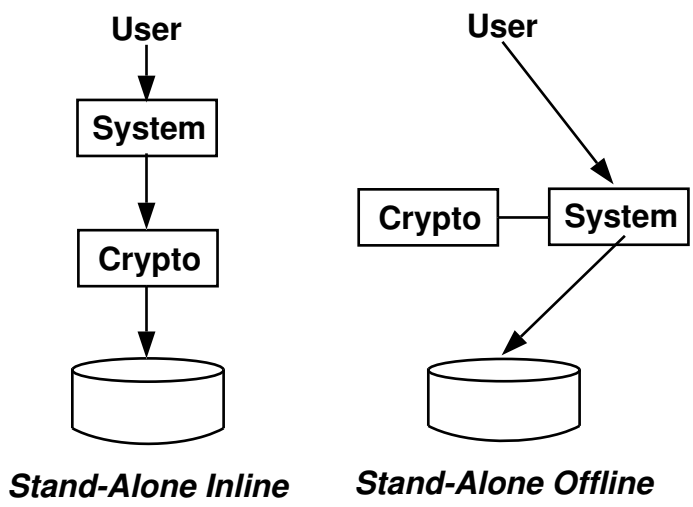

Figure 28: Stand-Alone Cryptographic Modules

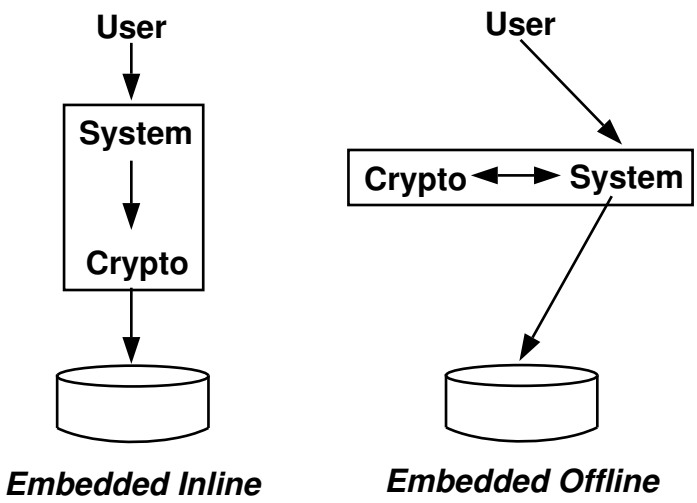

Figure 29: Embedded Cryptographic Modules

The inline/offline distinction is relevant only if confidential export is implemented. In this case, sensitive (unencrypted) data on the host must be prevented from "leaking" to the export media. Inline cryptographic modules achieve this by physically separating the host from the export media. Offline modules must, however, rely on the host to maintain this separation. A fuller discussion of

confidential and nonconfidential export, along with descriptions of valid implementations and requirements, can be found in *Cryptographic Modules* [6] document.

References

[1] U.S. Department of Commerce, National Institute of Standards and Technology. *Security Requirements for Cryptographic Modules (Draft)*, Federal Information Processing Standards Publication FIPS PUB 140–1, June 1992

[2] U.S. Department of Commerce, National Institute of Standards and Technology. *General Security Requirements for Equipment Using the Data Encryption Standard*, Federal Information Processing Standards Publication FIPS PUB 1027, April 1982

[3] Communications Security Establishment, *Federal Standard 1027: General Security Requirements for Equipment Using the Data Encryption Standard*, CID 09/11(1) April 1985

[4] National Computer Security Center. *Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG 005 version-1, July 1987

[5] Communications Security Establishment, *Cryptographic Endorsement and Assessment Program Standards*, NITSM 10/90, July 27, 1990

[6] Communications Security Establishment, *Cryptographic Modules.* July 1993

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT

Introduction 1

APPENDIX L Government Security Policy and Standards 2 3

Introduction

The Security Policy of the Government of Canada was issued in 1986 in order to ensure consistent and appropriate protection for government information and assets. There were a number of factors contributing to the need for a security policy at this time, including the absence of comprehensive policy, concern about over-classification of documents, and excessive numbers of security clearances. Perhaps the key factor was the enactment of the **Access to Information Act** and the **Privacy Act** in 1983. In prescribing types of information to be withheld from public release, these Acts pointed to a need for consequent policy and standards on the protection of such information. 4 5 6 7 8 9 10 11

In the ensuing years, the policy has evolved, influenced in part by the experience of the departments in implementing the policy. As well, the changing management philosophy of government has led to a more flexible approach, with more discretion to departments. Reflecting these factors, the policy has been updated and certain sections clarified. Security standards have been developed and appended to the policy. 12 13 14 15 16 17

The effective Security Policy was issued in February 1991. It consists of a foreword that describes an accountability framework, the policy, and guidelines on classifying and designating information, roles and responsibilities of certain departments and agencies with respect to security, and definitions. The four appendices contain security standards: 18 19 20 21 22

1. **Appendix A:** Security organization and administration 23
2. **Appendix B:** Physical security 24
3. **Appendix C:** Information technology security 25
4. **Appendix D:** Personnel screening standards 26

Objective & Scope

The objective of the Security Policy is to ensure that all classified or designated information and assets of the federal government are safeguarded in an appropriate manner.

Classified information and assets are those that are sensitive in the national interest, and relate to the defence, economic, foreign policy and similar interests of Canada.

Designated information, on the other hand, is information that is not in the national interest but requires safeguarding for other reasons. This category covers personal, proprietary and third-party information held by the government. Its inadvertent release could cause injury to individuals or private sector entities.

The policy applies to government institutions and the private sector entities with whom they contract.

The policy encourages a systems approach, integrating all areas of security in the design of a security program: personnel, physical, administrative, as well as information technology. The latter area incorporates communications-electronics security (COMSEC), computer security and network or system security.

Security Policy Considerations

Designing a security program involves consideration of three security concerns: confidentiality, integrity and availability. Confidentiality concerns the level of sensitivity of information or assets; that is, whether it is classified or designated, and vulnerable to unauthorized disclosure. Integrity concerns the accuracy and completeness of information, and the threat to integrity is modification. Finally, availability relates to the requirement that information or assets be available on a continuing basis or when required, and safeguards for availability counter the threats of interruption and destruction.

Determining what to protect, then, includes considerations of confidentiality, integrity and availability of information and assets, including information technology systems and personnel. Selecting safeguards is based on the threats to the information and assets.

In the area of information technology security, and in particular as it relates to the Canadian Trusted Computer Product Evaluation Criteria (Canadian Criteria), confidentiality, integrity and availability may all be concerns to be addressed with appropriate safeguards. In addition, accountability and assurance are addressed in the Criteria.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Accountability, Risk, and Guidance

Applying the Policy (In Brief)

As described in the foreword to the policy, Deputy Heads are accountable for security in their departments, again reflecting current government management philosophy. Accordingly, the policy allows considerable flexibility to departments in designing their own security systems. Departmental discretion in application is essential to ensure that only appropriate safeguards are implemented, that a security system is designed that addresses the unique circumstances of the department or the particular application.

The mechanism that allows government institutions flexibility in designing a security system is threat and risk assessment. Only by identifying the sensitivity, importance and value of information or assets, by assessing the threats to the information or assets, and making a risk management decision, will the most cost-effective security system be designed.

Selecting security safeguards for a computer system or network normally requires the assistance of security specialists. The departmental security office can provide such assistance.

There are also lead security agencies to assist in complex and specialized system design. The Communications Security Establishment (CSE) and the EDP Security Branch of the Royal Canadian Mounted Police (RCMP) provide advice and guidance to departments. The Canadian Criteria is an example of guidance in the form of technical standards.

Applying the Policy (In Brief)

Following are certain key requirements of the Security Policy and Standards that apply in the system security environment.

1. Conduct Threat and Risk Assessment (Policy, §2(b); Standards, A, §7, and C, §3):

   a. Inventory of Information and Assets:
   What are the confidentiality, integrity, and availability concerns of the system?
   b. Threat Assessment:
   What are the threats to the inventory of information and assets?
   c. Risk Assessment:
   Are existing or proposed security safeguards adequate?
   What is the likelihood of the threat occurring, and what is the potential injury in event of the threat being realized?
   d. Recommendations:
   Based on the first three steps, what should be done?

2. Apply Security Standards (Policy, §3; Standards, C):

   Identify information technology security standards application to the particular system environment.

Sections 7, EDP Security, 8, COMSEC, and 9, System Interconnection, are particularly important to the application of the Canadian Criteria.

Note the following from §9:
Institutions are required to conduct a threat and risk assessment of system interconnections, identifying the confidentiality, integrity and availability requirements of all components and the system or network itself.

3. Monitoring (Policy, p. 7; Standards, C, §11):

   a. Note the section on Monitoring in the policy, and expanded in the Standards, C, §11:
   Institutions, other than the Department of National Defence, are required to:

   - *have their classified or designated EDP systems inspected by the Security Evaluation and Inspection Team (SEIT) of the RCMP in accordance with technical-level standards. They must also consult the SEIT when major changes are planned in their information processing systems;*
   - *advise the SEIT within six months of receiving the SEIT report of their plan to deal with identified problems and thereafter provide annual progress reports.*

APPENDIX M Security Functionality Profiles

Introduction

The US TCSEC and the European ITSEC have "functional groupings" or "functional classes". The US have the familiar C1, C2, B1, B2, and B3 (A1 is an improvement in **assurance** but no additional functionality is added). The ITSEC has functionality groupings called F-C1, F-C2, F-B1, F-B2, and F-B3. These correspond directly to the US classes[33]. Further, the ITSEC contains classes called F-IN (integrity), F-AV (availability), F-DI (data integrity on networks), F-DC (data confidentiality — designed for encryption which is not covered under the standard functional classes derived from the US TCSEC), and F-DX (data integrity and confidentiality on networked systems) to cover some of the more pressing issues in computer security[34]. These ten functionality classes correspond to what the ITSEC calls "example functionality classes".

The first five example classes of the ITSEC are hierarchical in nature and are designed to allow for inclusions between and above. They were defined to meet functionality mapping requirements with the US TCSEC. The final five example classes, however, are not hierarchical and can be added to other functional classes (unlike the first five). Therefore, it is quite possible to create an F-[B1, IN, AV] machine.

Although rigid systems like the US TCSEC's functional classes have shortcomings, functional classes which can have trusted services added and/or deleted are undesirable as they can lead to multiple definitions of, say, a "C2+" system. The solution is to allow for a set of services which when grouped under a set of rules (constraints) create a profile of a product, hence the Canadian Criteria's "Security Functionality Profiles".

These security functionality profiles (profiles for short) are logical groupings under the constraints of the Criteria. Minor variations are described by version letters. Major changes require a new profile.

---

[33] For further information please see the TCSEC.

[34] For further information please see the ITSEC.

DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Scope

Equivalency & Other Criteria

Security Functionality Profiles

This appendix is intended as guidance only, and does not replace or supercede the requirements expressed in the Criteria. Examples are given as illustration only and are not the only acceptable solution in meeting the Criteria.

The Canadian Criteria has three founding premises:

1. functionality and assurance are distinct;
2. functionality can be described in terms of services, hierarchically divided into levels of service; and
3. security policy can be divorced from functionality and assurance.

Other criteria, most notably the US TCSEC, use a different approach. The US TCSEC is divided into "classes" where each class is actually a grouping of functionality, assurance, and security policy.

The Canadian Criteria does not provide a facility for the creation of Orange Book-style classes. However, to assist in determining the correspondence between products evaluated against the Canadian Criteria to those evaluated against the TCSEC, there exist *security functionality profiles.*

A security functionality profile can be used to define the *functionality* present in a US TCSEC class in terms of CTCPEC services. The assurance and security policy evident within the TCSEC class can *not* be represented in a security functionality profile.

Three reasons exist for this: First, the CTCPEC does not require any given security policy. Therefore, a vendor can create a product to a given profile using *any* security policy. Second, the CTCPEC assurance levels are not equivalent to the TCSEC. In many ways, the CTCPEC assurance levels are *more* stringent than the base[35] TCSEC assurance requirements. And third, TCSEC evaluations result in products rated to a "low water mark". Functionality above and beyond a given class, although examined, does not add to the rating received.

The profiles presented in this appendix can be used as a guide by which functionality between products can be compared. Therefore, profiles can be seen as "one-way mappings" from various criteria into CTCPEC terms. A "B1–equivalent" profile does not imply that all (or any) products evaluated to the TCSEC B1 class *exactly match* this profile. However, this profile does provide a mapping of the *functionality* of the B1 class *as written in the TCSEC* into CTCPEC terms.

There is no means by which a profile can be deemed equivalent (or equal) to a TCSEC class, since a profile *does not* contain security policy information nor assurance requirements. No-one, other than US authorized evaluation facilities,

[35] Base is used here in terms of the assurance requirements for the vendor as stipulated in the actual TCSEC. "Criteria creep" — the additional requirements that have crept into lower level classes from higher level classes — do not reflect the requirements as laid out in the TCSEC classes.

can create a B1 evaluated product and place it on the US Evaluated Products List (EPL).

Creation of Profiles

The Canadian Criteria presents four criteria whose constituent parts are known as services. Each service is divided into levels of service, each representing an increase in functionality, granularity of service, or strength of protection. Each service level has an associated constraint list. This constraint list is used to create logical profiles. They do not imply that the created profile is necessarily correct. The correctness of a profile must be verified by the evaluation authority.

Unlike the Federal Criteria or the ITSEC, the Canadian Criteria does not require a "dependency analysis" to be performed against the services selected. The constraint lists provided are sufficient control to indicate the minimal set of required services for a particular grouping. If a vendor's selection meets, at a minimum, the constraints *and* the selection of services is approved by the evaluation authority, the profile can be registered as a valid functionality security profile. There remains a small possibility that due to a particular configuration or implementation, a higher level of a given constraint may be required to satisfy the evaluation authority. The constraints remain a minimum acceptable limit.

These profiles can be used to define products which are currently under development and undergoing evaluation at CSE, have completed evaluation at CSE, or are considered useful configurations of products evaluated in either the United States or in the European Community.

What follows is a description of what a profile contains as well as a set of profiles which correspond[36] to the TCSEC ratings of C2 – B3.

Profile Semantics

A profile contains three constituent parts: The word "profile", a numeric identifier, and a version letter. The numeric identifier is unique for each grouping of services but does **not** define an assurance level (Assurance Criteria) to the grouping; nor does the numeric identifier indicate any hierarchy, either implied or otherwise. The version letter indicates modifications of the standard profile, such as inclusion of a new service or minor strengthening of a particular service. For example, the strengthening of the audit service would result in a version change. However, multiple changes to a profile, such as audit and the

[36] Note that profiles which are mapped are **not** considered to be 100% accurate in describing all systems which have achieved the foreign rating. Some products, for example those attaining a rating of B1 against the US TCSEC, may have services beyond those specified in the profile. The reader must be aware of the "low-water mark" method employed by the US evaluation teams. Products which are in common usage which contain extra services and that are mapped from the US TCSEC or other foreign criteria may be given versioned profiles. Please see "profile Semantics" for further information.

inclusion of integrity functionality would result in a new profile. Figures 30 and 31 illustrate.

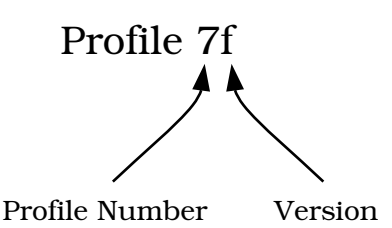

Figure 30: Profile Header Example.

For major modifications of a given profile, a new profile will be defined with references to the old profile. Figure 31 gives an example of a profile.

Profile 3
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Discretionary	CD-2
	Mandatory	CM-1
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
Availability	Recovery	AY-1
Accountability	I & A	WI-3
	Audit Level	WA-2
	Trusted Path	WT-1

Profile 3a
Modification of Profile 3

Functionality	Division/Mechanism	Level
Confidentiality	**Discretionary**	**CD-3**
	Mandatory	CM-1
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
Availability	Recovery	AY-1
Accountability	I & A	WI-3
	Audit Level	WA-2
	Trusted Path	WT-1

Figure 31: Profile Examples.

Predefined Profiles

The TCSEC Profiles

Below are profiles which represent functionality service groupings equivalent to the functionality of the U.S. TCSEC classes. The remaining profiles correspond to functionality groups commonly requested by various clients within government and industry as well as profiles which reflect some of the interpretations of the TCSEC which exist in the U.S.

These profiles provide one-way mappings for US evaluated products to be defined in CTCPEC terms. A product developed to a TCSEC-equivalent profile *does not* meet the requirements for the TCSEC class because a TCSEC class is *more* than a collection of services. Profiles *do not* include the assurance level nor the security policy.

Caveat: The profiles listed are as accurate as possible. The profiles are presented for guidance purposes only. Creation of a product to one of the profiles does not result in a product whose rating is equal to a US rating. The profiles, being devoid of assurance and security policy, present a guide to what is within a given class. Only US approved evaluation authorities are capable of granting US ratings.

Profile 1
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Discretionary	CD-2
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
	Self Testing	IT-1
Accountability	I & A	WI-2
	Audit Level	WA-2

Table 16: C2 Equivalent Profile.

Profile 2
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Discretionary	CD-2
	Mandatory	CM-2
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
	Self Testing	IT-1
Accountability	I & A	WI-2
	Audit Level	WA-2

Table 17: B1 Equivalent Profile.

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

Profile 3
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Covert Channels	CC-1
	Discretionary	CD-2
	Mandatory	CM-3
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-2
	Self Testing	IT-1
Accountability	I & A	WI-2
	Audit Level	WA-2
	Trusted Path	WT-1

Table 18: B2 Equivalent Profile.

Profile 4
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Covert Channels	CC-1
	Discretionary	CD-3
	Mandatory	CM-3
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-2
	Self Testing	IT-1
Availability	Recovery	AY-1
Accountability	I & A	WI-2
	Audit Level	WA-3
	Trusted Path	WT-2

Table 19: B3 Equivalent Profile.

Subsystem Profiles

The following profiles correspond to various subsystems either currently produced or in production. Most subsystems offer a form of partitioning as their major method of protection. This is evident from the profiles which show a heavy leaning towards mandatory profile controls. Please note that mandatory controls refer to the use of tagged information and do not necessitate the use of a particular policy.

Profile 5
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Mandatory	CM-1
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
Accountability	I & A	WI-2
	Audit Level	WA-2

Table 20: Standard Partition Subsystem

Profile 6
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Mandatory	CM-1
	Object Reuse	CR-1
Integrity	Discretionary	ID-1
	Separation of Duties	IS-1
Accountability	I & A	WI-3
	Audit Level	WA-2

Table 21: Strong Identification Subsystem

Profile 7
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Discretionary	CD-2
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
Accountability	I & A	WI-2
	Audit Level	WA-2

Table 22: Discretionary Based Subsystem

Service Specific Architectures

Many institutions and corporations, both government and private, have indicated interest in secure systems. However, the predefined systems of the TCSEC or the functionality classes of the ITSEC do not meet their requirements[37]. Provided below are three commonly requested security functionality profiles.

[37] It is true that the ITSEC allows one to add or remove pieces from their basic functionality classes. However, there are no predefined functionality components nor is there a methodology described for the creation of modified functionality classes.

Profile 8
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
Accountability	I & A	WI-2
	Audit Level	WA-4

Table 23: Audit Architecture

Profile 9
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Covert Channels	CC-1
	Mandatory	CM-3
	Object Reuse	CR-1
Integrity	Separation of Duties	IS-1
Accountability	I & A	WI-2
	Audit Level	WA-3
	Trusted Path	WT-1

Table 24: Partitioned Architecture

Profile 10
Base Profile

Functionality	Division/Mechanism	Level
Confidentiality	Discretionary	CD-2
	Mandatory	CM-3
	Object Reuse	CR-1
Availability	Recovery	AY-1
Integrity	Separation of Duties	IS-1
	Self-Test	IT-2
Accountability	I & A	WI-2
	Audit Level	WA-2
	Trusted Path	WT-2

Table 25: Comparted Mode Workstation Architecture

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

The Infinite Nature of Profiles

The basic premise behind the profile is that there is no fixed set of profiles which are "correct". If a grouping of services (functionality) meets a customer's requirements and does not break any of the Criteria's constraints then that system is a viable profile.

The fact that a profile is not listed above does not preclude that profile from being valid nor, for that matter, from being better than the profiles listed. It must be remembered that the above profiles are simply those most commonly used at the present time or those which reflect systems currently under evaluation or those types of systems which various institutions and corporations have shown interest.

Although the example profiles are provided they are not meant to be seen as what vendors must create as far as trusted systems go. The number of profiles the Canadian Criteria allow is enormous and space is limited. CSE holds final approval for any and all profiles, including those listed above. The given profiles are simply provided for guidance and do not necessarily imply that they are correct or are acceptable in Canada. As can be seen from profile 1, not all profiles which are valid elsewhere are valid in Canada.