APPENDICES 1

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.

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.

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.

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.

Integrity

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

Discretionary Integrity (CD)

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.

Mandatory Integrity (CM)

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.

Physical Integrity (IP)

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.

Rollback (IR)

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

Separation of Duties (IS)

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.

Self Testing (IT)

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.

Availability

Containment (AC)

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.

Fault Tolerance (AF)

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.

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.

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 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.

Mandatory Integrity

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.

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.

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.

Trusted Path

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.

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.*

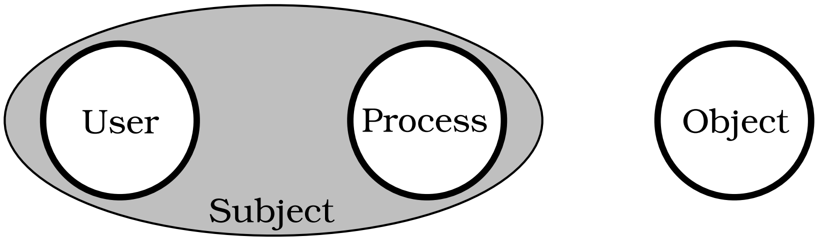

Figure 10: Trusted "Objects" in the TCSEC vs. Canadian Criteria

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.

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.

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

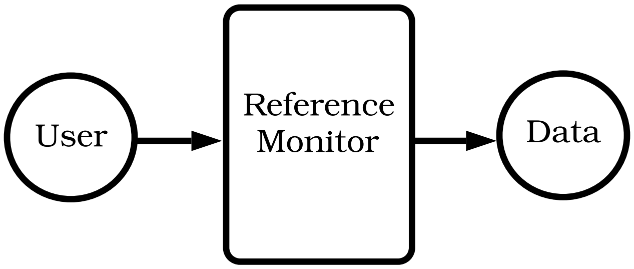

Figure 11: Classic View of a Reference Monitor

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.

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.

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.

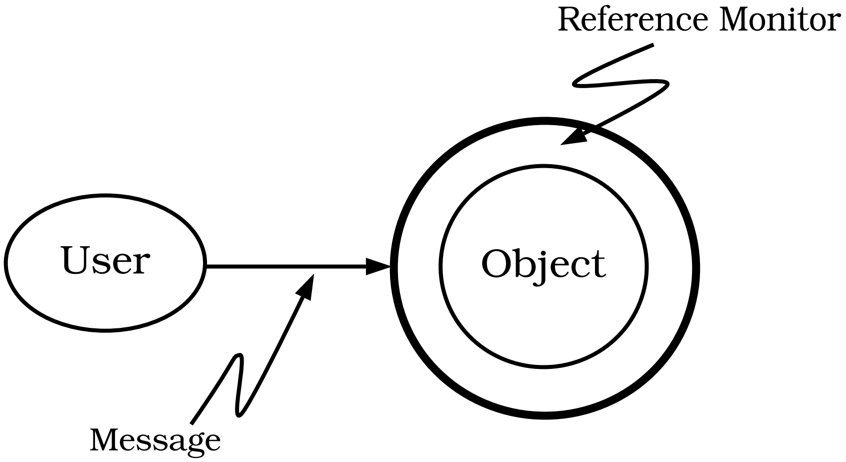

Figure 12: Reference Monitor As Encapsulator

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.

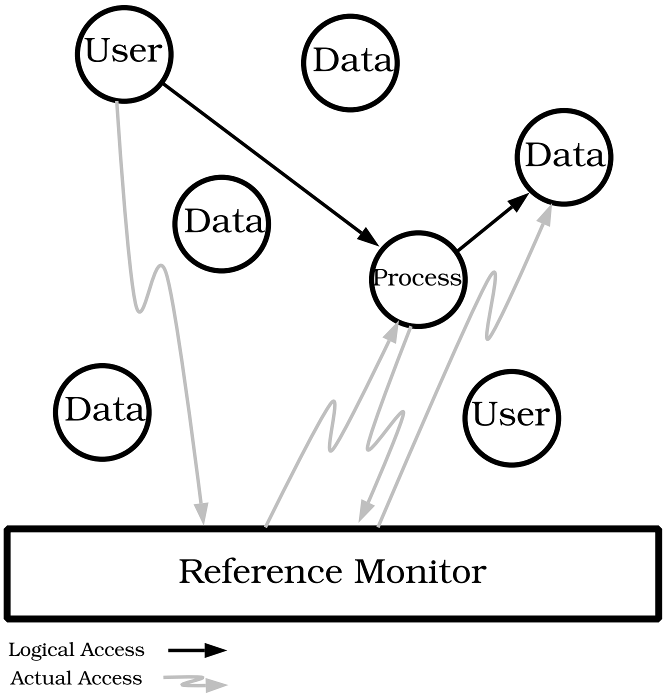

Figure 13: Entity Style (Message Passing) Reference Monitor

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.

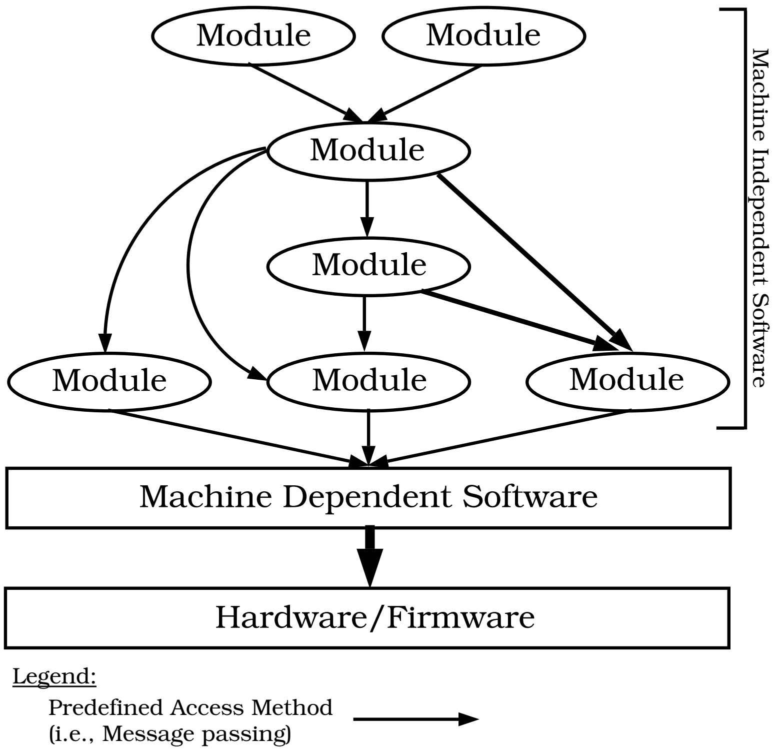

Figure 14: Modularity within A Trusted Environment

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.

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.

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.

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.

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.