Common Criteria Version 3.0 Update

The Common Criteria (ISO/IEC 14508) is undergoing its first major revision since being published as CC v2.1 in 1999 and CC v2.2 in 2004. Using the input of vendors and the experiences of those nations operating a CC certificate-producing scheme, the Common Criteria Recognition Arrangement nations developed a work plan and allocated resources to undertake the development of CC v3.0. The goals set forth by the CCRA nations were simple: CC v3.0 would eliminate redundant evaluation activities and reduce/eliminate those activities that contributed little to the final assurance of a product; clarify CC terminology to reduce misunderstandings; restructure and refocus the evaluation activities to those areas where product assurance would truly be gained; and add new CC requirements if needed.

CC v3.0 is currently on schedule to be released/posted for public comment in July 2005. Additionally, CC certificate-producing schemes will be encouraged to perform trial evaluations using the CC v3.0 with select evaluations for the purpose of fine-tuning the requirements prior to formal publication of CC v3.0 in 2006. As an integral companion to CC v3.0, the Common Evaluation Methodology (CEM) is also undergoing revision and is being released in parallel with the CC.

SUMMARY OF CHANGES

Part 1

Part 1 was updated to define and establish the use of consistent terminology for the entire CC standard and to reflect changes to the ASE/APE families.

Part 2

Part 2 defines and explains the terminology that is used to describe exactly what a Target of Evaluation (TOE) is supposed to do security-wise. The description of the TOE's security-behavior should be articulated to allow all interested parties (e.g. developers, consumers and evaluators) to have a common understanding of the security behavior of the TOE. This common understanding, together with a particular assurance level, establishes the value of a CC certificate for the consumer.

The vendor and scheme experiences with CC v2 demonstrated that the CC standard was too complicated, thus leading to a number of problems in writing and understanding Protection Profiles and Security Targets. The requirements were also written at different levels of specificity; some were so detailed as to be almost implementation-specific while others were very general. Therefore, Part 2 has been completely overhauled in CC v3.0. Complicated terms were broken down or removed; concepts were simplified and clarified; and the underlying paradigm has been made more uniform. Additionally, a number of problem areas within the CC have been removed.

CC v3.0 contains only 6 classes (reduced from 11), 45 families (reduced from 67), and is approximately 130 pages in length (reduced from 354).

Part 2 divides the security behavior of TOEs into five major, relatively independent, areas.

1. Internal security behavior - actions that occur internally in the TOE, such as access control
2. Connecting external entities to the TOE - identification, authentication and the like
3. Protecting communication between the TOE and connected external entities - maintaining confidentiality, integrity, non-repudiation, etc.
4. Security audit - logging of and responding to security-relevant events

5. Protection of the TSF - how the TOE protects itself against breakdown, physical attacks, resource exhaustion, etc.

It is anticipated that these changes will enable Protection Profiles and Security Targets to be easier to write and understand, more uniform in content, and less open to misinterpretation.

The requirements that were capable of being expressed using v2.2 can be expressed using v3.0, though perhaps by using different SFRs; a summary mapping of the Part 2 requirements of versions 2.2 and 3.0 can be found in Annex 5.

Part 3

As with Part 2, the changes to Part 3 are significant. Each of the changes was developed with the goal of improving the assurance of the TOE (and the product) with evaluation activities focused on only those areas that contribute to the assurance of a TOE. Many classes were consolidated and/or eliminated, while others were added to handle many of the evaluation difficulties encountered with the current CC v2.2. The classes, or groups thereof, are discussed as such below.

ASE/APE

In CC v2.2, ASE and APE contained numerous instances where elements were stated such that evaluation work was repeated by the evaluator for no assurance benefits. Also, insufficient guidance existed on determining the adequacy of Assumptions, Threats, Operational Security Policies (OSP), or Security Objectives statements. The consequences were that text found in the ST/PP could be determined to pass evaluation, yet proved to be useless for the end users (i.e. potential customers) in determining whether the TOE/product met their needs.

The approach taken for the ASE/APE rewrite was to organize the descriptions to yield a useful resulting ST/PP, while streamlining the work in evaluating it. This rewrite provides descriptions of good Assumptions, Threats, Organisational Security Policies, and Security Objectives statements, as well as clarifying that the purpose of the TOE Summary Specification is to explain how the TOE meets its claimed security requirements.

A summary mapping of the APE requirements of versions 2.2 and 3.0 can be found in Annex 1.

ACM/ADO/AGD/ALC

The update to the ACM/ADO/AGD/ALC classes was basically a rearranging of the contents to have clear delineation of the purpose of each family. For example , the configuration management requirements addressed in ACM should be in place over the entire lifecycle of the TOE, which is in fact the subject of ALC; and the actions required by the administrator (which are described in AGD) might also include actions associated with the start-up of the TOE, which is part of ADO.

These four classes were therefore rearranged into two classes: ALC which addresses the requirements associated with the developer's site, and AGD which addresses all of the requirements associated with the user's/ customer's site.

A summary mapping of the ACM/ADO/ADG/ALC requirements of versions 2.2 and 3.0 can be found in Annex 2.

ADV

The problems being experienced by vendors and schemes with CC v2.1 ADV were varied. In some cases, the evaluation work required far exceeded the assurance gained (e.g. FSP.2 called for a *substantial* amount of work that far exceeded an EAL4 level of assurance). In some cases, the evaluation work was inefficient. In other cases, the evaluation work reflected a technology

bias that the CC purported not to have (the two levels of abstraction approach in HLD/LLD is infeasible for very complex TOEs and unnecessary for very simple TOEs). In other cases, the components were not granular enough to allow the assurance to track along with the EAL scale (FSP remained unchanged from EAL1 through EAL3). In still other cases, it was unclear what exactly the authors were talking about (*formal* low level design?). Additionally, some basic IT security principles were completely missing (the absence of an architecture argument meant that that all of the claimed security functions could be corrupted or bypassed, making them meaningless). And in all cases, there was no acknowledgement that some parts of the TSF are more critical and security-interesting than others, resulting in evaluation analysis being performed on all parts of the TOE, including those that no security professional would ever bother with, thereby expending vast amount of unnecessary effort and cost.

The ADV rewrite now reflects a reasonable scale of increasing assurance with a corresponding amount of work. A new family was created to address the need for an argument for a sound architecture. In some areas, simple modifications/patches to CC v 2.2 fixed the problems. In other cases, a complete rebuild was required. Due to the rebuild activities, an element-by-element comparison between CC v2.2 and CC v3.0 will not be possible; however a summary component level mapping of the ADV requirements can be found in Annex 4.

It is important to note that the ADV requirements[1] for CC v3.0 contain more text (explanatory front matter as well as elements in the components and, hence, more work units in the methodology), which might **erroneously** lead a reader to believe it contains more work for evaluators, developers and certifiers. The increase in text is due to the description of both the principles underlying security analysis as well as how to *save* effort in performing it.

ATE
ATE was updated only to reflect the new ADV and its terms (i.e. to explicitly link COV to the FSP requirement and to link DPT to the component- or module-level description in the TOE design).

AVA
AVA merged the Security of Function (SOF) analysis into the Vulnerability Analysis (VLA) family (to reflect there is no longer a separate SOF claim made in PPs/STs). It also merged the Misuse (MSU) analysis into the AGD family (because it simply extends the requirements of the quality of those documents). Finally, it created a new lowest level of vulnerability analysis, based upon public domain information. Note that vulnerability analysis now bears the tri-graph "VAN".

A summary mapping of the AVA requirements of versions 2.2 and 3.0 can be found in Annex 3.

ACO
A new class on Composition is currently being developed to address the issue that arises when a TOE includes a product that had, itself, been evaluated, such as a database running atop an evaluated operating system. Current understanding does not provide a means of combining the results if the two are evaluated separately; an evaluation must be performed upon the combination. However, because the operating system is already evaluated, there needs to be a means to leverage off the results of that evaluation. The new Composition class defines what needs to be done to accomplish this.

CEM
The new CEM is presented according to class/family/component, to reflect the structure of the CC, rather than by EAL, as was done in v2.1. Methodology is provided for all components up through EAL5 (and higher, for cases where such were available)[2].

[1] Other than the new ARC class, which is new to v3.0
[2] Despite the presence of added methodology, mutual recognition is still only for those components up through EAL4.

Table 1: APE requirements

The following table provides a mapping between the APE/ASE requirements of v2.2 and v3.0. The contents as defined in Figure B.1 (v2.2) and Figure 5 (v3.0) of Part 1 have been put into outline format, and then mapped.

v3.0	v2.2
1. PP Introduction A. PP reference	1. PP Introduction A. PP identification B. PP overview
B. TOE overview	2. TOE Description
2. Conformance Claims A. CC Conformance Claim B. PP Claim C. Package Claim	[As defined in section 5.4, as modified (and re-titled "Conformance Results") by interpretation CCIMB-0008]
3. Security Problem Definition A. Threats B. OSPs C. Assumptions	3. TOE Security Environment A. Assumptions B. Threats C. OSPs
4. Security Objectives A. SOs for the TOE	4. Security Objectives A. SOs for the TOE
B. SOs for the development environment C. SOs for the operational environment	B. SOs for the environment
D. Security Objectives rationale	(8A, below)
5. Extended Components Definition	(5A1 and 5A2, below: the explicit reqs)
6. Security Requirements A. SFRs for the TOE (including SFO claim)	5. IT Security Requirements A. TOE Security Requirements 1. TOE SFRs [Part 2 *and* explicit reqs] [no SOF claim – see also Table 3]
B. SARs for the TOE	2. TOE SARs [Part 3 *and* explicit reqs]
C. Security requirements rationale	(8B, below)
(requirements for environment are now optional)	B. Security Requirements for IT environment
(no separate App notes section; these can be put into Intro)	7. PP Application Notes
	8. Rationale A. Security Objectives Rationale
	B. Security Requirements Rationale

Table 2: ACM/ADO/ALC/AGD requirements

CC v2.2	CC v3.0
ACM_SCP – what is tracked by CM	ALC_CMS – scope of CM: what is covered
ACM_CAP – a CM system; its capabilities; maintenance of CIs	ALC_CMC – capabilities of CM system (including whether automated)
ACM_AUT – automated CM	
ALC_DVS – developer security	ALC_DVS – developer security
ALC_FLR – flaw remediation	ALC_FLR – flaw remediation
ALC_LCD – lifecycle development	ALC_LCD – lifecycle development
ALC_TAT – tools and techniques	ALC_TAT – tools and techniques
ADO_DEL – delivery procedures (at both developer's site and user's site)	ALC_DEL – delivery procedures (at the developer's site) [user-side moved to AGD_PRE]
ADO_IGS – installation, generation, start-up procedures (at both developer's site and user's site)	AGD_PRE – preparation of TOE at the user's site: User-side delivery procedures (receipt); User-side start-up procedures; [developer-side start-up procedures moved to ALC_CMC]; subject to misuse analysis (formerly AVA_MSU) – see Table 3
AGD_ADM	AGD_OPE – operation: guidance on how to operate the TOE, aimed at humans that interact with it; subject to misuse analysis (formerly AVA_MSU) – see Table 3
AGD_USR	

Table 3: AVA requirements

CC v2.2	CC v3.0
AVA_CCA – covert channel analysis	Covert channel analysis moved into VLA, as part of vulnerability analysis (it applies only to TOEs enforcing information-flow-like policies)
AVA_MSU – misuse analysis: how might the documentation be misinterpreted in a way that leads to insecure use?	Misuse analysis moved into the AGD families that address the documents subject to such analysis – see Table 2
AVA_SOF – strength of function analysis: how strong are the permutational/probabilistic mechanisms?	SOF analysis moved into VLA, as part of vulnerability analysis; no more SOF claim made
AVA_VLA – vulnerability analysis	AVA_VAN – vulnerability analysis: include SOF analysis and perhaps covert channel analysis as part of VAN; also look to public domain sources of vulnerabilities.

Table 4: ADV requirements

CC v2.2	CC v3.0
Families Addressing Decomposition of TSF	
[no v2.2 equivalent]	FSP.1 allege security-enforcing interfaces
[no v2.2 equivalent]	FSP.2 describe security-enforcing interfaces
	FSP.3 describe security-relevant interfaces
FSP.1 describe all interfaces	
	FSP.4 give all details of all interfaces except indirect error messages
FSP.2 give all details of all interfaces	
FSP.3 give semiformal presentation	FSP.5 give semiformal presentation (including indirect error messages)
FSP.4 give formal presentation	FSP.6 give formal presentation
HLD – high level description of TSF, regardless of assurance level	TDS – a high-level description of TSF at low assurance levels, migrating toward a more detailed description as assurance increases
LLD – low-level description of TSF, regardless of assurance level	
IMP.1 implementation subset provided and examined	IMP.1 entire implementation available; subset examined
IMP.2 entire implementation provided and examined	
IMP.3 structured implementation	[deleted: covered by INT]
[no v2.2 equivalent]	IMP.2 assurance that source yields object
RCR.1 informal correspondence RCR.2 semiformal correspondence RCR.3 formal correspondence	[correspondence is distributed through families: each representation must demonstrate correspondence to the previous one.]
Families Addressing Understandability and Soundness of TSF	
[no v.2.2 equivalent]	ARC.1 explain architectural soundness (in terms of details provided in other evidence)
[no v2.2 equivalent]	INT.1 modularity of particular TSF subset
INT.1 modularity of TSF	INT.2 modularity of TSF
INT.2 reduced of complexity: layers	INT.3 reduced of complexity: layers
INT.3 minimal complexity	INT.4 minimal complexity
SPM.1 informal model	[deleted; the informal SPM is the collection of Objectives in the ST]
SPM.2 semiformal model	[deleted; the semiformal SPM is the collection of SFRs in the ST]
SPM.3 formal model	SPM.1 formal model

Table 5: Part 2 requirements

CC v2.2	CC v3.0
FAU – Security Audit	FAU – Audit
ARP: security audit automatic response	ARP: security audit automatic response
GEN: security audit data generation	GEN: security audit data generation
SAA: security audit analysis	SAA: security audit analysis
SAR: security audit review	[ability to review is protected under FDP_ACC]
SEL: security audit event selection	[ability to select is protected under FDP_ACC]
STG: security audit event storage	[audit data is protected under FDP_ACC]
FCO - Communication	FCO – Communication Protection
	AED: availability of exported data
[Old FDP_ETC and FDP_UCT]	CED: confidentiality of exported data
[Old FDP_ITC and FDP_UCT]	CID: confidentiality of imported data
[Old FDP_ETC]	ETC: export to outside TSF control
[Old FDP_ETC and FDP_UIT]	IED: integrity of exported data
[Old FDP_ITC and FDP_UIT]	IID: integrity of imported data
[Old FDP_ITC]	ITC: import from outside TSF control
NRO: non-repudiation of origin NRR: non-repudiation of receipt	NRE: non-repudiation of exported data NRI: non-repudiation of imported data
	TED: translation of exported data
	TID: translation of imported data
[Old FDP_ETC and FFP_UNO]	UNE: unobservability of export
FCS – Cryptographic Support	[removed; crypto is a *means* of achieving requirements: for protection of data in transit (FCO), for protection of data at rest (FDP), etc.]
CKM: cryptographic key management	
COP: cryptographic operation	
FDP – User Data Protection	FDP – Data Protection and Privacy
ACC: access control policy	ACC: access control
ACF: access control functions	Integrated into FDP_ACC
DAU: data authentication	Ownership of data is a security attribute
ETC: export to outside TSF control	Moved to FCO
IFC: information flow policy	Integrated into FDP_ACC
IFF: information flow functions	Integrated into FDP_ACC
	ISA: intialisation of security attributes
ITC: import from outside TSF control	Moved to FCO
ITT: internal TOE transfer	[removed; applies only to distributed TOEs]
[Old FMT_MSA]	MSA: mgmt of security attributes
RIP: residual information protection	Moved to FPT
ROL: rollback	ROL: rollback
SDI: stored data integrity	[rules for protecting data integrity are defined under FDP_ACC; reacting to integrity errors is defined by FPT_TST]
UCT: inter-TSF user data confidentiality transfer protection	[removed: applies only to distributed TOEs]
UIT: inter-TSF user data integrity transfer protection	[removed: applies only to distributed TOEs]
[Old FPR_UNL]	UNL: unlinkability
[Old FPR_UNO]	UNO: unobservability
FIA – Identification & Authentication	FIA – Identification, Authentication, and Binding
AFL: authentication failures	AFL: authentication failures
ATD: user attribute definition	Moved to FDP_ISA
	LOB: lock-out of bindings
SOS: specification of secrets	QAD: quality of authentication data
	SUA: subject/TSF authentication

	TBR: TSF binding rules
	TIN: TSF information
	TOB: termination of bindings
UAU: user authentication	UAU: user authentication
UID: user identification	UID: user identification
	URE: user registration
USB: user-subjects binding	USB: user-subjects binding
FMT – Security Management	Integrated into FDP
MOF: management of functions in TSF	[mgmt functions protected under FDP_ACC]
MSA: management of security attributes	Moved to FDP
MTD: management of TSF data	[TSF data protected under FDP_ACC]
REV: revocation	Revoking attributes covered by FDP_MSA; revoking ability covered by FDP_ACC
SAE: security attribute expiration	Integrated into FDP_MSA
SMR: security management roles	[access to roles protected under FDP_ACC]
FPR – Privacy	Integrated into FDP
ANO: anonymity	FIA_URE, FIA_UID and FIA_USB
PSE: pseudonymity	FIA_URE, FIA_UID and FIA_USB
UNL: unlinkability	Moved to FDP
UNO: unobservability	Moved to FDP
FPT – Protection of the TSF	FPT – Protection of the TSF
AMT: underlying abstract machine test	Covered by FPT_TOU
FLS: fail secure	FLS: fail secure
[Old FRU_FLT]	FLT: fault tolerance
ITA: availability of exported TSF data	Moved to FCO_AED
ITC: confidentiality of exported TSF data	Moved to FCO_CED
ITI: integrity of exported TSF data	Moved to FCO_IED
ITT: internal TOE TSF data transfer	[removed: is an implied requirement]
PHP: TSF physical protection	PHP: TSF physical protection
[Old FRU_PRS]	PRI: priority
RCV: trusted recovery	RCV: trusted recovery
RPL: replay detection	Covered by FCO_IED, FCO_IID
RVM: reference mediation	Covered by ADV_ARC
SEP: domain separation	
SSP: state synchrony protocol	[removed: is an implied requirement]
STM: timestamps	Moved to FMI_TIM
TDC: inter-TSF TSF data consistency	[removed: is an implied requirement]
[Old FPT_RIP]	RIP: residual information protection
[Old FRU_RSA]	RSA: resource allocation
TRC: internal TOE TSF data replication consistency	[removed: is an implied requirement]
	TOU: testing of users
TST: TSF self test	TST: TSF self test
FRU – Resource Utilisation	Integrated into FPT
FLT: fault tolerance	Moved to FPT
PRS: priority of service	Moved to FPT_PRI
RSA: resource allocation	Moved to FPT
FTA – TOE Access	Integrated into FIA
LSA: limitation on scope of selectable attributes	Integrated into FDP_ISA
MCS: limitation on multiple concurrent sessions	Integrated into FIA_TBR
SSL: session locking	

TAB: TOE access banners	Integrated into FIA_TIN and FIA_TBR
TAH: TOE access history	
TSE: TOE session establishment	
FTP – Trusted Path/Channel	Integrated into FCO
ITC: Inter-TSF trusted channel	
TRP: trusted path	
	FMI - Miscellaneous
	RND: random number generation
[Old FPT_STM]	TIM: time stamps
	CHO: choice