2010 Second International Conference on Advances in System Testing and Validation Lifecycle

Unsteady Ground: Certification to Unstable Criteria

Joe Loughry
Computing Laboratory
University of Oxford
Oxford, UK
Email: joe.loughry@stx.ox.ac.uk

***Abstract*—Cross Domain Systems for handling classified information complicate the certification test and evaluation problem, because along with multiple data owners comes duplicate responsibility for residual risk. Over-reliance on independent verification and validation by certifiers and accreditors representing different government agencies is interpreted as conflating the principle of defence-in-depth with the practice of repeated verification and validation testing. Using real-world examples of successful and unsuccessful certification test and evaluation efforts to guide the development of a new communication tool for accreditors, this research aims to reduce time and cost wasted on unnecessary retesting of the same or similar security requirements during security test and evaluation in multi-level environments.**

***Keywords*—cross domain systems; certification and accreditation; security test and evaluation; certification test and evaluation;**

I. Introduction

What happens when a system that has been developed successfully under one particular set of security testing criteria suddenly finds itself needing to conform to a completely new set of rules? The question is not hypothetical; the same situation arises whenever a new certification scheme is adopted, *e.g.*, the Common Criteria (CC) throughout its various revisions, Health Insurance Portability and Accountability Act (HIPAA) and Sarbanes–Oxley laws in the United States, or as regularly as clockwork in the case of the peculiar species known as a Cross Domain System (CDS). This paper aims to show that a lack of communication amongst government certifying and accrediting agencies leads to unnecessary duplication of effort and multiplication of cost with no concomitant improvement in security.

The first part of this paper defines the term CDS and assurance requirements for security accreditation. Section II describes a particular security test and evaluation problem that is unique to CDS. Section III contains two real-world examples illustrating the occurrence of the problem in practice. Finally, Section IV describes progress towards a solution.

A. Characteristics of Cross Domain Systems

CDSs handle classified information. These systems are unlike other installations that process classified information in that they encounter new or changed security testing criteria all the time. Unlike safety-critical systems, whose certification criteria are stable, usually having been established by professional engineering organisations before being given the weight of law [1]–[4], the security certification criteria under which classified data processing systems are evaluated are ever-changing. By definition, CDS installations always span at least one boundary between security domains controlled by different data owners. Data owners do not trust one another with access to their data, hence the need for a controlled interface between security enclaves [5]. An elementary example of a CDS application is the one-way interface (sometimes called a *data diode*) allowing wire service news articles to flow into a classified intelligence-gathering system (Figure 1). The purpose of a CDS is separation of security domains. The data owner of a classified system worries about two potential threats: accidental leakage of classified information to the unclassified side, called a *spill*, and the potential for introducing malicious code into the classified system from an outside source.

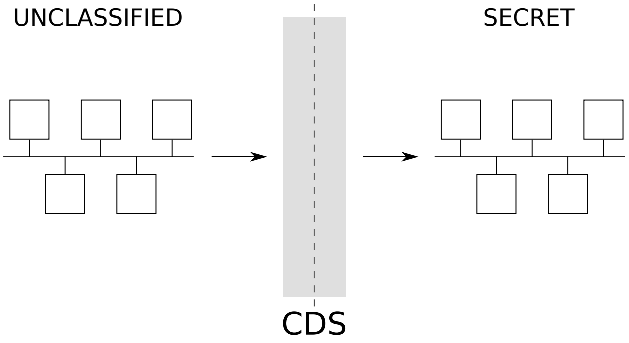


Figure 1: The simplest imaginable cross-domain system.

The example shown in Figure 1 is simplistic because it presupposes a unidirectional flow of information from low to high (unclassified to classified) and only two security domains. More realistic CDS installations cope with bidirectional flows—such as a web browser inside the security enclave that needs to be able to search unclassified databases outside. In practice, multi-directional information flow requirements are commonplace. Depending on the capability of the CDS, one system might handle scores of channels simultaneously, interconnecting many security domains at widely different classification levels. Internally, the CDS maintains separation of information by classification and source, routing inputs to outputs according to rule sets specified by the data owners. Advanced CDS systems have the capability to transform data in flight, whether by transliterating message formats or by

978-0-7695-4146-4/10 $26.00 © 2010 IEEE
DOI 10.1109/VALID.2010.21

sanitising classified information for release at a lower security level (Figure 2).

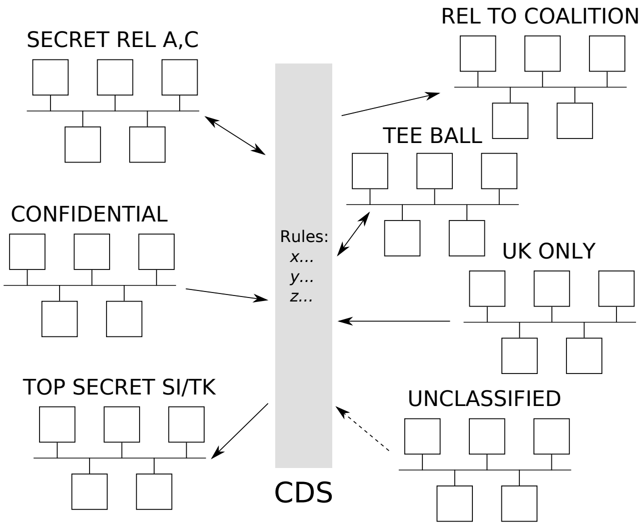


Figure 2: A more realistic CDS scenario with complex rules.

B. *How Risk Reduction is Done and Measured*

As an irreducible result of inherent complexity, CDS applications present a uniquely high risk of failure whilst at the same time having the potential to cause a great deal of damage to national security. Consequently, they are developed and tested with utmost care. It begins with formalised requirement and specification reviews, proceeding through software development accompanied by systems and security engineering, vetting of programme personnel, design and code inspections, automated static analysis of source code, unit and system level testing, configuration management of software and hardware, documentation, training of installers and operators, audits, physical security, and process improvement [6], [7].

Once software development is complete, the Certification Test and Evaluation (CT&E) phase begins. In the case of CDS applications having field-alterable rules, CT&E is performed using a representative set of processing rules designed to exercise the capabilities of the system. In practice, the software developer (being the one most familiar with the system) creates an initial set of tests and expected results based on Factory Acceptance Test (FAT) procedures and test coverage analysis. Then a different organisation is engaged to use those procedures to perform Independent Verification and Validation (IV&V) and penetration testing on the system. At each stage in the certification process, *findings*—deviations from expected results—are reported to the developer and the certifier (Table I).

Table I: Categorisation of findings according to severity.

Category	Relative Severity
I	These are show-stoppers.
II	Less severe than a Cat I finding but must be corrected before CT&E can proceed to completion
III	Do not necessarily prevent certification; fix may be deferred up to 180 days after Interim Approval to Operate (IATO)
IV	Minor problems, sometimes deferred indefinitely

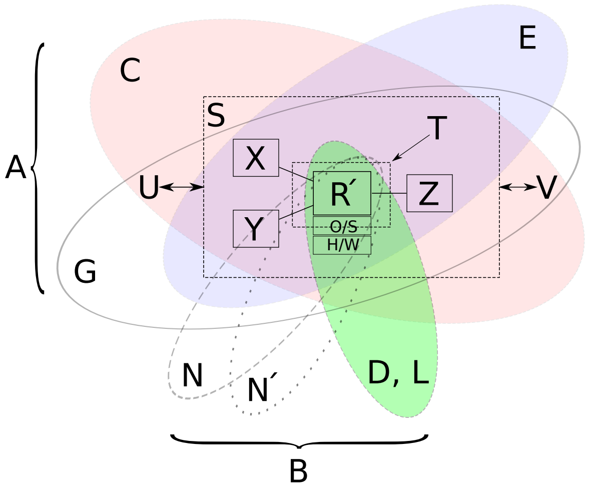
After CT&E, each instance of the CDS needs to be installed and accredited for a particular use in a particular location. After the initial site survey, trained installers configure rule sets in coordination with all of the data owners involved and set up the system in the location where it is to be used, though it is not connected to all of the network endpoints yet. At this time, site operations personnel, system administrators, and security officers are trained on the new system. Before the CDS is allowed to connect for the first time, it needs to be tested one more time in a process known as Security Test and Evaluation (ST&E) for accreditation. Since the ultimate purpose of a CDS is to reduce residual risk to a level acceptable by the data owner(s), in each case there is a Designated Approving Authority (DAA)—a person delegated by the Principal Approving Authority (PAA) of the data owner formally to accept responsibility for residual risk in the operation of the CDS. IV&V contractors assist the DAA with ST&E by exercising the CDS through test procedures specific to the site until the DAA is satisfied and agrees to accept responsibility for the residual risk. The DAA then issues an Approval to Connect (ATC) and allows the system to operate. The formal Approval to Operate (ATO) is for a limited time—no more than three years—and contingent on security-relevant changes not being made to the system without approval.

The purpose of these four phases—trusted software development, CT&E, trusted delivery, and ST&E—in combination is to reduce the amount of risk to a level acceptable by the data owner. The goal of the DAA is always to minimise residual risk.

II. The Problem

All of the aforementioned steps are necessary to guarantee the level of assurance needed by a CDS. But it is at this point we argue that the process loses coherence. It was earlier mentioned that data owners typically do not trust other data owners, at least not completely. With multiple data owners come multiple DAAs. With multiple DAAs come repeated rounds of ST&E, typically conducted by the same IV&V contractors—who, being already familiar with the system, are the logical ones to test it at a reasonable cost—and using similar or identical test procedures.

But this repeated testing at the ST&E phase to the same or similar criteria by different government agencies would seem to be conflating the principle of defence-in-depth with the practice of IV&V. If testing is done competently, *i.e.*, following well-defined test procedures based on established principles of test coverage analysis and test design, then it seems obvious that repeating effectively the same tests—often performed by



A	Participants in country A
B	Participants in country B
C	Customer
D	Software developer
E	Systems integrator
G	A's government security agency
L	CC testing Laboratory
N	B's government security agency for Common Criteria (CC)
N'	Another government information-security certifying agency of B
R'	The product being CC evaluated
S	System being developed by E
T	TOE boundary
U, V	Other systems outside S
X, Y, Z	Sub-components of S

Figure 3: Software developer D had influence only over the Target of Evaluation (TOE) R', whereas the system as a whole was built by E. When the customer's government security representative G did their analysis, they considered the superset of S^* including the external interfaces to U and V.

the same people according to scripts derived from a common source—does not lead to increased assurance [8, Chapter 13], [9, Chapter 14]. Why does this situation arise? It seems to be a structural problem emergent from the compartmentalisation of esp. national security systems. To solve it, we can look at CT&E as a testable microcosm of ST&E.

III. Examples

As evidence that the problem is real, consider the following examples of the same CDS under different circumstances.

A. An Unsuccessful Common Criteria Evaluation

The Common Criteria for Information Technology Security Evaluation is an ISO standard for developers to show that products satisfy specified security requirements to a specified level of assurance.

The original motivation for this research derived from events related to a military communications software project in 2006. The example has been anonymised for national security reasons. Customer C in country A ordered a complex system S to be built (Figure 3) by systems integrator E. One component of the system proposed by E was to be R'—a slight variant of R, a mature and well-regarded product of the D corporation in country B. R had been evaluated for security by N' (and other government agencies) numerous times before. Acting as a subcontractor to E on the project, D made the necessary alterations to R. (In reality, D and E were two branches of the same company, not an uncommon occurrence in large organisations.) The customer imposed an additional requirement, however: R' needed to have a Common Criteria certificate, which R did not yet have. D felt that CC evaluation presented no great difficulty, as R had endured many similar security certifications over the past decade; satisfying the CC ought to be no great burden and as a bonus, R would benefit from the—quite expensive—evaluation of R'. An authorised CC testing laboratory, L was engaged to assist with the security evaluation of R'. D and L worked together to assemble the voluminous documentation required to make their case before N, the CC certifier in country B that would formally evaluate the security of R'.

Shortly before the evaluation documentation of R' was to be delivered to N for formal security evaluation, C made the overall system S^* available to G, their government's most trusted information security adviser. G pronounced S 'not fit for purpose' and recommended against its adoption. C cancelled the project, and soon after funding dried up, the CC evaluation of R' was abandoned.

What can be learnt from this example? The software should not have failed its CC evaluation. It had previously withstood numerous, rigorous, and repeated security evaluations over a period of more than ten years. In a sense, the root cause of the failure was a requirements change, but it was a change in the environment having nothing to do with functionality. Nevertheless, major shifts in CT&E criteria are a fact of life for CDS developers and the development methodology must acknowledge this.

B. DIACAP Certification

The second example has to do with a U.S. Department of Defense (DOD) Information Assurance Certification and Accreditation Programme (DIACAP) certification of R'', a later version of the original system R from the first example.

Is there a difference in the post-CT&E software defect rate, measured in terms of the number of Category I–IV findings between different versions of the same system in subsequent rounds of ST&E by different DAAs? (See Figure 4.) Data from approximately five rounds of CT&E and penetration testing conducted by N' and other agencies over a five-year period

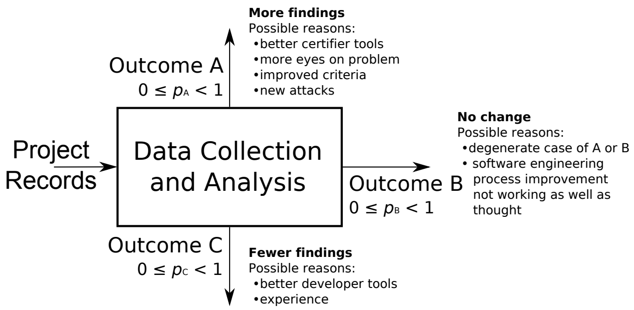


Figure 4: Is there a difference in the post-CT&E software defect rate between different versions of the same system in subsequent rounds of ST&E by different DAAs?

on successive versions of R and R'' are available. The answer to the question in Figure 4 will give us a quantitative way of measuring the efficacy of CT&E methodology *vis-à-vis* CDS the way it is done today.

This is a prime example of the most general case of certification to unstable criteria: when a new standard replaces an older one and developers of existing products are forced to switch over. The software has outlived its evaluation criteria. In the case of R'', Director of Central Intelligence Directive (DCID) 6/3 is being replaced by the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53, which defines an entirely new set of security controls along with the risk management framework in NIST SP 800-37 under DIACAP [5], [10]–[13]. What is unique about this example is that the certifiers are learning the new standard at the same time as the developer—R'' is the first system to undergo NIST SP 800-53 certification as it replaces DCID 6/3. So far, certification is proceeding normally, without any of the problems that occurred during the CC evaluation of R'.

Secondly, by observing the activity of developers, certifiers, IV&V, and DAAs throughout the process of CT&E and ST&E on a new version of an existing CDS, it should be possible to elucidate the ideal amount of inter-DAA communication consistent with a minimum of duplicated testing to a defined level of security assurance.

IV. Proposed Solution

Security accreditation of CDSs ultimately comes down to a relatively small number of people. A new tool being developed at the University of Oxford, called *nihil obstat*, is designed to facilitate concord amongst DAAs by presenting test procedures and test results in a form acceptable to all of the DAAs responsible for ST&E accreditation of a particular CDS.

The theory that informs the design of the tool characterises residual risk as a function of multiple DAAs with a range of security clearances who know about different sets of risk and risk mitigations that are at least partly disjoint. Each data owner perceives a set of risks A_i that would be desirable to mitigate, a set of risks B_i it is possible to mitigate, and their relative complement $R = A_i - B_i$, being the residual risk known to that accreditor. If we can prove to subsequent DAAs that the aggregate residual risk is less than R_i, then repeated rounds of ST&E may be avoided.

V. Conclusion and Future Work

Cross Domain Systems are unique because they exist in the intersection of stringent security requirements and ever-changing standards. They are different from safety-critical systems in two important ways: firstly, safety-critical standards do not change as rapidly as security standards do; and secondly, CDSs are *always* located on the disputed borderlands between mutually distrusting security domains. With multiple data owners come multiple responsible authorities (DAAs) and with multiple DAAs come repeated rounds of the same or similar tests, without any concomitant improvement in security assurance. The problem seems amenable to solution by a process of gathering, collation, and presentation of test procedures and test results to DAAs in a neutral format. A pair of related examples yield real-world experience with unstable criteria—both successful and unsuccessful—derived from contiguous versions of a single CDS spanning more than a decade. CT&E is more visible, but ST&E happens all the time and can be streamlined.

Acknowledgement

I would like to thank my supervisor, Andrew Martin for his advice and encouragement; Joanna Ashbourn for teaching me scientific writing; Cornelius Namiluko and John Lyle for their insightful comments; Olav Kjono for access to data; Steve Steinberger for conversations over coffee and help with interpretation; and Chip Auten and Annie Cruz for mentoring.

References

[1] N. G. Leveson, "High pressure steam engines and computer software," *IEEE Computer*, vol. 27, no. 10, pp. 65–73, October 1994.

[2] N. Leveson, *Safeware: System Safety and Computers*. Boston, Massachusetts: Addison–Wesley Publishing Company, 1995.

[3] *Software Considerations in Airborne Systems and Equipment Certification*, RTCA, Inc., December 1, 1992, DO-178B.

[4] U.S. Department of Transportation, Federal Aviation Administration, *Software Approval Guidelines*, June 3, 2003, order 8110.49.

[5] Director, Central Intelligence, "Protecting sensitive compartmented information within information systems," 1 August 2000, DCID 6/3.

[6] R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd ed. John Wiley & Sons, Inc., 2008.

[7] M. Fagan, "Design and code inspections to reduce errors in program development," *IBM Systems Journal*, vol. 15, no. 3, pp. 182–211, 1976.

[8] B. Beizer, *Software Testing Techniques*, 2nd ed. New York: Van Nostrand Reinhold, 1990.

[9] T. A. Kletz, *What Went Wrong?: Case Histories of Process Plant Disasters*, 4th ed. Burlington, Massachusetts: Elsevier, 1999.

[10] U.S. Department of Commerce, National Institute of Standards and Technology, *NIST Special Publication 800-53, Revision 3: Recommended Security Controls for Federal Information Systems and Organizations*, June 2009, final Public Draft.

[11] R. Ross, A. Johnson, S. Katzke, P. Toth, G. Stoneburner, and G. Rogers, *Guide for Assessing the Security Controls in Federal Information Systems*, July 2008, NIST Special Publication 800-53A.

[12] National Institute of Standards and Technology, *Guide for Applying the Risk Management Framework to Federal Information Systems*, February 2010, NIST Special Publication 800-37 Revision 1.

[13] United States Department of Defense, "DoD information assurance certification and accreditation process (DIACAP)," ASD(NII)/DoD CIO, November 28, 2007, DoD Instruction 8510.01.