Tokeneer ID Station
Approaching the Common Criteria

S.P1229.40.2
Issue: 1.1
Status: Definitive
19th August 2008

Originator

David Cooper

Approver

Janet Barnes (Project Manager)

Copies to:

National Security Agency

Praxis High Integrity Systems
File

SPRE Inc.

Contents

1 Introduction 3

2 Common Criteria 4
2.1 Issues with the CC 4
2.2 An Alternative 5

3 The TIS Security Target 7

4 TIS Security Model 8

Document Control and References 9
Changes history 9
Changes forecast 9
Document references 9

1 Introduction

In order to demonstrate that developing highly secure systems to the level of rigour required by the higher assurance levels of the Common Criteria is possible, the NSA has asked Praxis High Integrity Systems to undertake a research project to re-develop part of an existing secure system (the Tokeneer System) in accordance with their own high-integrity development process. This re-development work will then be used to show the security community that it is possible to develop secure systems rigorously in a cost-effective manner.

This document discusses some of the issues around the Common Criteria, and how they affect the current Security Target.

2 Common Criteria

2.1 Issues with the CC

Evaluations to the ITSEC standards and to the CC standards are similar in principle, but in practice there is a difference of emphasis.

ITSEC has a number of standard *assurance* measures that must be followed to reach each of the seven levels (E0 to E6). But it has no standard set of security functionality that is expected — each product or system is expected to define its own security policy, and it is against this security policy that security is measured.

CC is in some respects more flexible. It has no standard set of measures, assurance or functional, that all products and systems have to meet. In theory, almost any set of security functions and assurance measures could be defined in a security target, and security evaluated against that. But in practice it has seven standard assurance measures (so-called packages), labelled EAL1 to EAL7, and very comparable to ITSEC's levels. It also has a large collection of standard security functions to choose from, reminiscent of the TCSEC (Orange Book).

So the CC's apparent flexibility is often not used. Security Target writers and Protection Profile writers find their job is easier if they pick up the standard offerings and adjust them in a small way. It is easier to write documents this way, easier to argue for their completeness, and easier to get buy-in from a range of stakeholders.

However, the savings made in writing the security target must be offset against the cost during development when development processes and functions are required to meet all the security target requirements.

Equally importantly, concentrating on a *standard* set of security functions and assurance measures often misses the key security properties of the actual product in hand.

There are some specific examples of this in our experience.

1 Tokeneer TIS PP nowhere states straightforward security requirements such as

> "no unauthorised user shall be allowed into the secure enclave"

and also fails to mention some key dependencies on the environments, such as

> "the secure enclave is protected against physical attack so that the only reasonable method of entry is through the doorway controlled by the TIS"

2 The Mondex electronic cash card had two key security properties (and a number of support properties): "no monetary value shall be created" (i.e. it will not be possible to use the system to forge money) and "no monetary value shall be un-accounted" (i.e. it will not be possible for the

system to lose money during transactions). These are the two key reasons for the existence of the electronic cash card, yet would not be found in the CC's set of standard security functions.

3 The Mondex Certificate Authority was housed in a *very* secure location. The major threat was from insiders obtaining monetary value illegally. Hence most of the technical security features were designed to protect against fraud, not malicious attack, and required a significant amount of manual procedure in support. The security target had to justify these security mechanisms, which were in no way "standard".

4 The Tokeneer PP has requirements on backup and restore. However, the amount of configuration data is very small, the amount of modification that is expected to be made to it is small, and the task of re-entering the values should a failure occur is not onerous. Although backup and restore can be argued to be a useful function, it is not a necessary *security* one. However, the CC encourages such unnecessary security properties to be defined.

The problems with the "standard" approach to security are:

1 Key security properties are not captured, and hence not proved, not traced to, and potentially not implemented.

2 Unnecessary security properties are required, costing time and effort to develop and evaluate, and possibly obscuring the true needs of the system.

2.2 An Alternative

We have experience of a different approach, which we used on the Mondex Certificate Authority, and I personally used on the Mondex electronic cash card with a previous company. These were carried out under the ITSEC rules, not the CC rules, but the approach should still be applicable:

1 Engage with the accreditors early (in our case, this was CESG), and ensure that they agree with all procedures and documents as you go.

2 Ruthlessly trim the security target down to the fundamental security requirements.

3 Develop specific security mechanisms to address specific threats, rather than incorporating a range of common mechanisms that have been proved to be useful in the past, but for which no clear justification can be identified for this specific product or system.

4 Make trade-offs *at the security target stage* between technical security mechanisms, environmental security assumptions, and the effort of implementation. That is, analyse the threats created by the chosen environment, decide whether the implementation cost of protecting against these threats using technical means is acceptable, and if not, modify the environment to reduce the threats.

5 Tailor the assurance mechanisms to the properties being enforced. So, for example, if dead-lock freeness is critical, apply analysis techniques that prove the absence of dead-lock, rather than going for a one-size-fits-all approach of, say, refinement proofs.

Such an approach is gaining support on this side of the Atlantic. As mentioned, under ITSEC we have followed such a method, with active support from CESG. We are also engaged in a funded project to improve the accreditation of military avionics (and other systems) from both safety and security perspectives, which has been arguing for a similar approach. The project, SafSec has been trying to convince CESG and others that the design and evaluation of secure systems should be driven by risk assessment and management, i.e. you identify the real security objectives and real assurance requirements, and you only need to build and evaluate these. SafSec has defined two processes: "unified risk management" and "risk directed design", which should result in no unnecessary functions and no unnecessary assurance: everything is traceable back through root objectives and assurance requirements to the risks and measures of risk that they are derived from. Clients in the MoD are enthusiastic about this approach, as they see it as a way of avoiding wasted effort, and therefore reducing time and costs of development and certification.

The consequences of adopting such an approach for the Tokeneer TIS Kernel would be

1 No protection profile (just develop a security target directly).

2 A smaller security target (at least, the security target would be smaller than the current combination of the security target and the protection profile).

3 Clearly recognised, high-level security aims, such as "protection of the enclave".

4 Clear reliance on the environment, making it easier to assess the impact of deploying in a different environment.

5 In the case of the Tokeneer TIS Kernel, there is probably no need to deviate significantly from the EAL5 package for assurance. Some refinement is necessary, especially with respect to testing (as the SPARK approach reduces the need for testing, replacing it with analysis), but such refinement should be arguable as achieving EAL5 goals by other means.

3 The TIS Security Target

As we agreed at our kick-off meeting, we have written the Security Target by basing it on the Protection Profile from SPRE, and excluding those portions that would force the development beyond the scope of the re-development project. As Bill made clear originally, he favours re-designing Tokeneer, rather than re-developing the kernel, and this has been reflected in his PP. We believe that the Security Target we have presented is a reasonable reduction based on the aims of this project.

But note that the exclusions the ST makes from the PP would not necessarily prevent the achievement of EAL5. They are exclusions in scope of this particular system, against a PP that is broader in scope. So, for example, the PP discusses an exit control, which we have excluded. This does not make the system any more or less secure; it just makes it different. Some of the exclusions will limit the environments in which the system can be installed, but will not prevent it being certified to EAL5 in the environments where it is allowed. For example, the PP defines some mechanisms for protection of the communications to the peripherals. This is necessary in hostile environments, but not in environments where attackers have no unsupervised physical access to the connections. Both are acceptable security targets, which make different demands upon their environments.

Of course, the system we are actually building now will not be certifiable to EAL5 because it is not a working system with real peripherals. But I believe that a real system developed in its entirety along the lines we have been developing, and evaluated against this ST, would be able to achieve EAL5.

The discussions in the first part of this document, however, point to a wider issue of evaluation. To achieve your aim of cost-effective evaluations to the higher levels the whole process needs to be addressed, from analysing the security needs, through writing the security target and developing the software, to the actual evaluation itself. I believe that this project is focussed primarily on the middle section of this: developing the system from requirements to code. As such it is not the vehicle for addressing the improvements needed at the front (security vulnerability analysis, security target and protection profile writing) or at the back (working with the evaluators, penetration testing, maintenance, etc.).

4 TIS Security Model

We are working on the security *model* for the TIS, which will include what we believe will be the key security properties expected of the system. We have identified six properties, listed below. (For comparison, the security policy model for the Mondex electronic purse had 12 abstract and 3 concrete properties, and the MULTOS smartcard had 7.)

1 If the latch is unlocked by the TIS, then the TIS must be in possession of either a User Token or an Admin Token. The User Token must have valid ID, Privilege, and I&A Certificates, and either have a valid Authorisation Certificate or have a template that allowed the TIS to successfully validate the user's fingerprint. Or, if the User Token does not meet this, the Admin Token must have a valid Authorisation Certificate, with role of "guard".

2 If the latch is unlocked automatically by the TIS, then the current time must be close to being within the allowed entry period defined for the User requesting access.

"close" needs to be defined, but is intended to allow a period of grace between checking that access is allowed and actually unlocking the latch. "Automatically" refers to the latch being unlocked by the system in response to a user token insertion, rather than being manually unlocked by the guard.

3 An alarm will be raised whenever the door/latch is insecure.

"insecure" is defined to mean the latch is locked, the door is open, and too much time has passed since the last explicit request to lock the latch.

4 No audit data is lost without an audit alarm being raised.

5 The presence of an audit record of one type (e.g. recording the unlocking of the latch) will always be preceded by certain other audit records (e.g. recording the successful checking of certificates, fingerprints, etc.)

Such a property would need to be defined in detail, explaining the data relationship rules exactly for each case.

6 The configuration data will be changed, or information written to the floppy, only if there is an Admin person logged on to the TIS.

These properties appear to capture the key behaviours: allowing entry only under certain circumstances; alarming when a security problem occurs; auditing; and controlling access to the working parameters of the system. The model will *formalise* these using Z in terms of the formal functional specification, and then demonstrate that they are indeed upheld by the functional specification.

Document Control and References

Changes history

Issue 0.1 (23/4/2003): internal review within Praxis

Issue 0.2 (24/4/2003): internal review within TIS team

Issue 1.0 (25/4/2003): issued to client

Issue 1.1 (19/8/2008): Updated for public releae

Changes forecast

None.

Document references

1 www.safsec.com