

Realistic User Expectations of Assurance Levels

Tony Boswell
CLEF Technical Manager
SiVenture





Overview

- ⇒ Are EALs (and SFRs) related to user expectations?
- ⇒ What would move us closer to user-experiences?
- ⇒ Is this still CC?
- ⇒ Conclusions

Why care about user expectations?

- ⇒ Because CC certification is no good if it doesn't meet user needs

...rather than *our* needs

At the moment, for some types of TOE and some users, it's not clear we are meeting either set of needs *well*.

This makes it difficult to manage *realism* in user expectations

We have something that works (we manage to complete a lot of evaluations). But it seems to be hard work for all of us, and we aren't always sure what the end-result means.

SFRs & User Expectations

- ⇒ SFRs are not how I hear users express security requirements
 - they tend to be more uneven...with a lot of high-level ideas (authentication, encrypted channels, permissions, secure remote management,...), and a few low-level techniques (smart cards, TLS/IPSec, etc)
- ⇒ Does this matter?
 - When we show a link from the high-level concerns to an SFR (e.g. FIA): No
 - When we can't express a simple concept (like a TLS channel used for a particular type of channel): Yes
- ⇒ Shouldn't the user be able to 'see the product' (not just a list of abstractions) in the SFRs?

SFRs & User Expectations - Solutions

This is:

- ⇒ a criticism of CC part 2
 - we should get around to making it easier to use for real TOEs
- ⇒ something that can be helped by the development of good PPs for specific technologies and or domains
- ⇒ ...and something that I'm not going to talk more about here!

EALs & User Expectations

- ⇒ EALs are not how I hear users express assurance requirements
 - but they're not very clear about what they mean instead...
 - ...and usually they mean it to be specific to **their** operational environment
- ⇒ Does this matter?
 - That we don't have a good way to describe the assurance we (quite expensively) evaluate? Yes, I think so.

Understanding EALs

- ⇒ How do we understand an EAL?
 - by reading what is done to achieve it
 - by comparison with other EALs

CC assurance levels seem to make most sense if you think of EAL4 as the ‘real’ baseline, and everything else as defined by perturbations of this baseline.

Do these perturbations originate from assurance? Is it just **luck** that CC requirements look so similar between levels (‘just change the bits in bold font’)? Or are we only defining other levels in terms of things we find **easy** to express as changes to EAL4?

Expectations of assurance?

What might a user naively expect in a description of a security level?

- something that makes clear what analysis and testing has and hasn't been done
 - what was looked at? (E.g. what aspects of the environment were covered in testing?)
 - did the evaluator examine source code? Or was the testing black box?

- something that helps in understanding the risks of actually using the TOE
 - this relates to the evaluated configuration
 - deals with user *consequences* (not abstract criteria)



Let's think about EAL2...

What else could we do for EAL2? – (1)

- ⇒ Better use of ‘standard use cases’ (or ‘requirement clichés’)
 - allow more standardisation/sharing of both specification and analysis/test (e.g. vulnerability searches, platform/environment configuration, links to FIPS140)

What else could we do for EAL2? – (2)

- ⇒ sensitivity analysis for evaluated configuration?
 - we know that many users will want to move outside the evaluated configuration; can we make this easier by identifying ‘mostly harmless’ types of changes, and changes where we might worry?
- ⇒ spend proportionally more time on secure installation?
 - of course we already cover installation and initialisation (AGD_PRE, ADV_ARC)...but maybe this deserves more time and attention at EAL2 (e.g. to help with the sensitivity analysis)

What else could we do for EAL2? – (3)

- ⇒ more analysis of process
 - perhaps not so much the security aspects such as ALC_DVS & ALC_DEL, but more about identifying the use of techniques that would help to identify and correct security issues

- ⇒ more examination of developer ‘general testing’
 - checking the general correct operation of the TOE, including under appropriate stress conditions

What else could we do for EAL2? – (4)

- ⇒ selective ‘deeper’ development testing and analysis
 - e.g. in areas where SFRs are complex, or stress conditions are difficult to create or understand
 - how can we give credit for ‘good’ practices (e.g. tools), without making these mandatory?
- ⇒ support other analysis with selective code review
 - e.g.
 - generation of nonces
 - valid calling of FIPS140 certified modules
 - secure use of platform authentication facilities
 - appropriateness of test tools and strategies

Is this still CC?

- ⇒ If evaluation were completely ‘free-form’ then it would not be CC as we know it
- ⇒ Should we define a new approach in a new Part 3?
 - probably not: the point here is that at lower assurance levels (at least) then we should adapt to the ‘shape’ of the TOE
- ⇒ The intention here is not to do either of these, but to reflect priorities
 - set in an ST or PP
 - established by the evaluators as they investigate the SFRs, installation, and TOE environment
- ⇒ Adopting these approaches is best done in a community

Conclusion

- ⇒ We should try to make assurance activities reflect user experience and hence more closely match their expectations
- ⇒ EAL4 may be about right, because that was where we drew our baseline (it allows flexibility because at least we have all inputs and activities incorporated)
- ⇒ This does suggest some modification to CC part 3 (and CEM), but PPs could also help – maybe using new assurance levels and extensions to part 3
- ⇒ Defining standard approaches to common functionality clichés should just be done...and done well (with collaboration and open review)



Questions?

Tony Boswell
tony.boswell@siventure.com
tel: +44 1628 651 361