SiVenture
A division of NDS

# Attack Potential: Using it Properly and Evolving it for the Future

Tony Boswell

CLEF Technical Manager

SiVenture

9th International Common Criteria Conference

## Overview

- What do we expect from vulnerability rating?
- What do we get in CC?
- Lessons from using attack potential in CC
- Future directions and CCv4

# What do we expect?

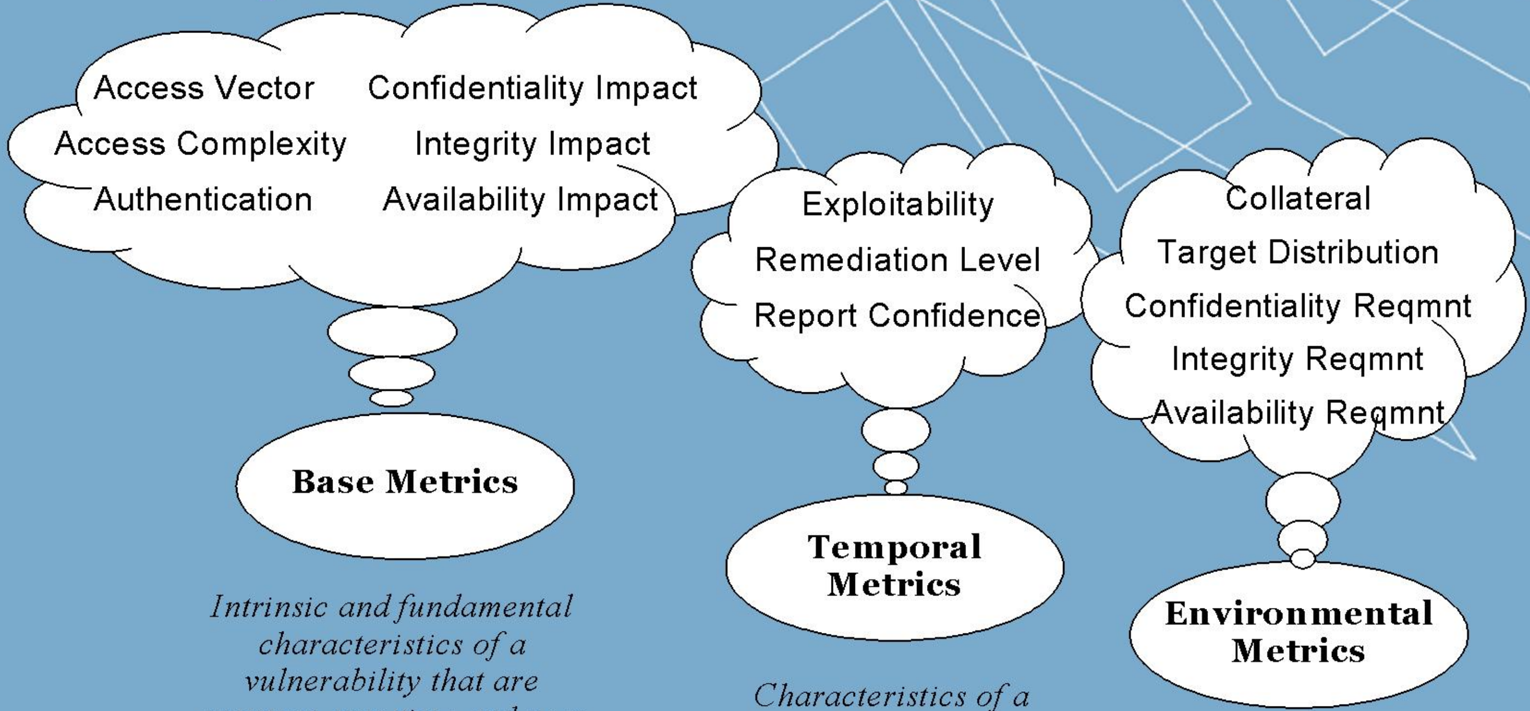
What do we expect from a vulnerability ‘score’ or ‘rating?

- a decision metric
- applicable to
  - process evaluation (DVS, DEL, etc.)
  - specialised areas (e.g. ICs)
- simplicity?
- specificity?
- detail?
- support for risk management

Take as a given that there are separate aspects for generic vulnerability and its operational effect and impact

SiVenture
A division of NDS

# A comparison: CVSS



Access Vector
Access Complexity
Authentication
Confidentiality Impact
Integrity Impact
Availability Impact

**Base Metrics**

*Intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments*

Exploitability
Remediation Level
Report Confidence

**Temporal Metrics**

*Characteristics of a vulnerability that change over time but not among user environments*

Collateral
Target Distribution
Confidentiality Reqmnt
Integrity Reqmnt
Availability Reqmnt

**Environmental Metrics**

*Characteristics of a vulnerability that are relevant and unique to a particular user's environment*

- **Presented as a score (0..10) and the 3 source vectors**

# So…would CVSS be better for CC?

- No – attack potential calculations are more focussed on a specific TOE: for analysis, not notification
- But…we should note the common problem of many unknown deployment environments and configurations and the emphasis on the whole vector, not just a single score
- Scoring tips are provided as a support for consistency, but less detail and examples than CC
- Even the CVSS vector is not sufficiently detailed for CC purposes, studying an individual TOE in detail
  - e.g. supporting assurance maintenance/re-evaluation/ surveillance

# Typical issues remaining

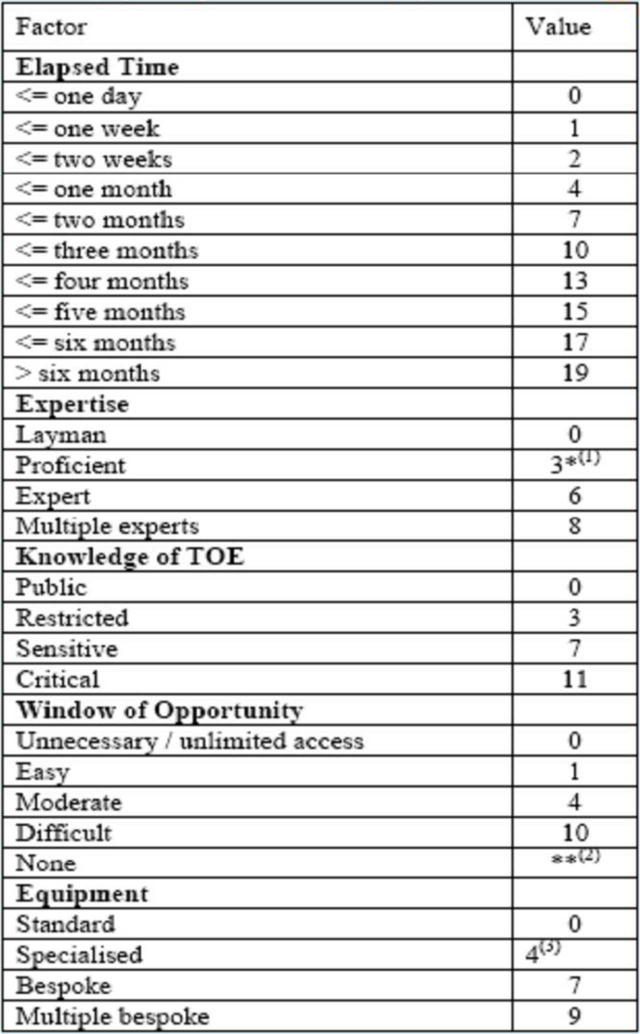
- hard to understand the meaning of a score/rating
- hard to 'perturb' and reapply
  - including use for composition
- hard to map to real operational risk
  - often have hidden 'average' or 'default' or 'worst case' assumptions in scoring
- unclear evidence basis
- potential inconsistencies between individuals/labs/ TOEs
  - quantitative does not necessarily mean objective

SiVenture
A division of NDS

# What do we get in CC?

- Single ‘vector’ now – no separate Identification and Exploitation

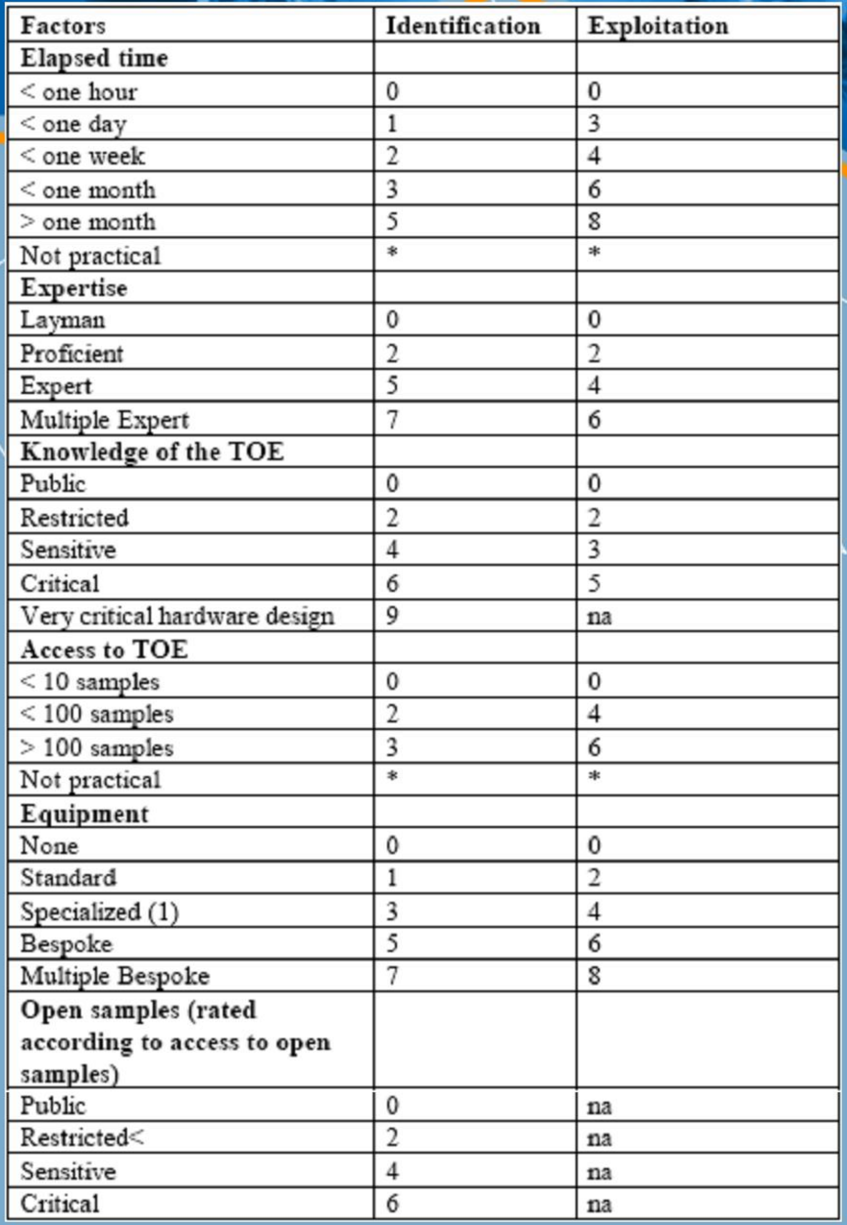
- Time
- Opportunity
- Expertise
- Equipment
- TOE Knowledge



| Factor | Value |
|---|---|
| **Elapsed Time** | |
| <= one day | 0 |
| <= one week | 1 |
| <= two weeks | 2 |
| <= one month | 4 |
| <= two months | 7 |
| <= three months | 10 |
| <= four months | 13 |
| <= five months | 15 |
| <= six months | 17 |
| > six months | 19 |
| **Expertise** | |
| Layman | 0 |
| Proficient | 3*(1) |
| Expert | 6 |
| Multiple experts | 8 |
| **Knowledge of TOE** | |
| Public | 0 |
| Restricted | 3 |
| Sensitive | 7 |
| Critical | 11 |
| **Window of Opportunity** | |
| Unnecessary / unlimited access | 0 |
| Easy | 1 |
| Moderate | 4 |
| Difficult | 10 |
| None | **(2) |
| **Equipment** | |
| Standard | 0 |
| Specialised | 4(3) |
| Bespoke | 7 |
| Multiple bespoke | 9 |

# JHAS table

- Specialises attack potential calculations for smart cards
- Retains two ‘vectors’ – with explanations
- Supported by lots of examples (*illustrations*)
- Strong demand for risk management and composition support



| Factors | Identification | Exploitation |
|---|---|---|
| **Elapsed time** | | |
| < one hour | 0 | 0 |
| < one day | 1 | 3 |
| < one week | 2 | 4 |
| < one month | 3 | 6 |
| > one month | 5 | 8 |
| Not practical | * | * |
| **Expertise** | | |
| Layman | 0 | 0 |
| Proficient | 2 | 2 |
| Expert | 5 | 4 |
| Multiple Expert | 7 | 6 |
| **Knowledge of the TOE** | | |
| Public | 0 | 0 |
| Restricted | 2 | 2 |
| Sensitive | 4 | 3 |
| Critical | 6 | 5 |
| Very critical hardware design | 9 | na |
| **Access to TOE** | | |
| < 10 samples | 0 | 0 |
| < 100 samples | 2 | 4 |
| > 100 samples | 3 | 6 |
| Not practical | * | * |
| **Equipment** | | |
| None | 0 | 0 |
| Standard | 1 | 2 |
| Specialized (1) | 3 | 4 |
| Bespoke | 5 | 6 |
| Multiple Bespoke | 7 | 8 |
| **Open samples (rated according to access to open samples)** | | |
| Public | 0 | na |
| Restricted< | 2 | na |
| Sensitive | 4 | na |
| Critical | 6 | na |

## Other hardware parameters

Other parameters may be interesting and relevant for other areas, especially when dealing with hardware devices:

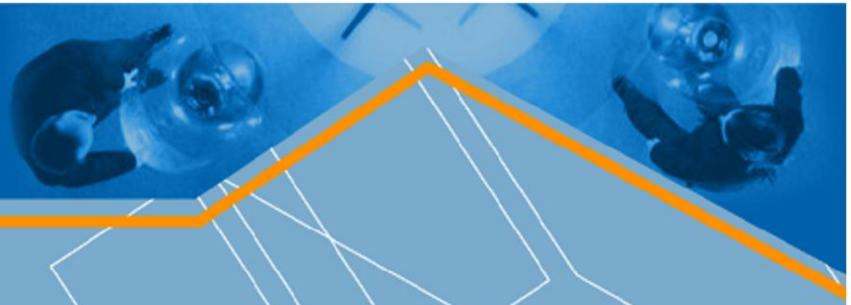
- Access to the device

  (Mechanical, Working, 'Operational')

  **note that points may be gained for each separate device needed

- Additional parts required

  (Standard, Specialised, Bespoke)



# Prominent themes

- user demand for risk management support
  - implies understanding vulnerability ‘picture’ (not just the foreground)
  - we probably can’t do this perfectly, but we can improve
- specialised calculation, but with transparency for compositions and system building
- how to deal with consistency across different specialised types of TOE, when we use different methods of attack potential calculation?
  - the margin here is too small! (Apologies to Fermat)

# Lessons from using attack potential (1)

- Can we really rate things (reliably) with tables in this way?
  - yes, they give us structure and boundaries. But we need to use guidelines and examples, and put effort into achieving common experience, problem solving and practice (e.g. working groups and evaluation community forums)
- What use is the single number (‘score’)?
  - it allows a basic sorting and sifting, and almost forces a sanity check. *But it should be obviously absurd to try to ‘use’ a single number as a comparative and informative assurance measure*

# Lessons from using attack potential (2)

- Formal methods (and sometimes source code analysis tools) take us down a path of translating from an ambiguous domain (in which we may be all too comfortable) into a much harsher domain
  - the final result may be a record of the activity, but the *process* of translation is really responsible for resolving the ambiguities and exposing assumptions

# Lessons from using attack potential (3)

- We can use attack potential even where no known vulnerability exists
  - e.g. ‘bounding calculations’ can be used to say that ‘If a vulnerability exists then it has an attack potential of *at least x*’
- Transparency (in documenting the analysis) allows inconsistencies to be accommodated and overcome
  - standard formats from tables and examples *invite* comparisons between different calculations (real and in reference documents such as from JHAS)

# Lessons from using attack potential (4)

- Although very useful for risk analysis, the attack potential discussion (and indeed the whole vulnerability analysis) is ‘hidden’ by placing it in the ETR

# Future directions and CCv4 (1)

- Vulnerability analysis *starts* a discussion…
  - we shouldn’t assume we ‘know it all’: the discussion enables evaluation results to be challenged, challenging a developer to respond in the areas the evaluators are most concerned about
  - this contrasts with CC part 2, which in general *stops* discussions: we stop at a hard-won, but often unsatisfactory translation from what we wanted (and intended) to say
- It is important to think about how we *record* such a discussion

# Future directions and CCv4 (2)

- So, noting the usefulness for risk management, can we extend this idea of a visible discussion to more of the evaluation?

  It looks as though when we examine other assurance properties then we could use a similar idea of ‘parameters’ that structure a discussion and conclusion.
- Although there may be a single ‘score’ or vector at the end, this is given lower emphasis by reporting to end-users (and hence risk managers) in a more visible way, exposing the analysis
- Of course some details have to remain non-public! But perhaps not as much as at present.

# Future directions and CCv4 (3)

- ➲ Early vulnerability analysis and theoretical attack potentials (attack potentials in principle, or based on limited experiments) can drive priorities for evaluation, could even give a rationale for differences in depth and type of evaluation activity for a specific TOE
- ➲ We can learn to deal with this degree of subjectivity in a positive way, based on transparency and comprehensiveness in the analysis

# Future directions and CCv4 (4)

- It may even be a developmental (i.e. 'growing up‘ or maturing) step to move from a CC that requires us to state evaluation as a series of TOE-independent work units (in CEM) to one in which we feel comfortable with judging the evaluation by its results in the form of a vulnerability *discussion*
  - of course the discussion remains constrained by the structure imposed by the parameters, and by comparison with good normative examples (perhaps supplemented by expressing common security features in this form)
- It remains a question as to how we do ‘quality assurance’…but one that (I think) has an answer

# Questions?

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

## References

***CVSS**:* *A Complete Guide to the Common Vulnerability Scoring System Version 2.0,June 2007, http://www.first.org/cvss/cvss-guide.pdf*

***CEM**:* *Common Methodology for Information Technology Security Evaluation: Evaluation Methodology, CCMB-2007-09-004, v3.1 Release 2, September 2007*

***JHAS**:* *Application of Attack Potential to Smartcards, v2.5 Revision 1, April 2008, CCDB-2008-04-001*

***VLA-Centric**:* *www.cesg.gov.uk/products_services/iacs/cc_and_itsec/media/formal-docs/vla-centric_evaluation.pdf*