Author(s): Ouedraogo, Moussa; Mouratidis, Haralambos; Khadraoui, Djamel; Dubois, Eric
Title: A probe quality metric taxonomy for assurance evaluation
Year of publication: 2010
Citation: Ouedraogo, M., Mouratidis, H., Khadraoui, D. and Dubois, E. (2010) 'A probe quality metric taxonomy for assurance evaluation.', Proceedings of Advances in Computing and Technology, (AC&T) The School of Computing and Technology 5th Annual Conference, University of East London, pp.201-208.

A PROBE QUALITY METRIC TAXONOMY FOR ASSURANCE EVALUATION

Moussa Ouedraogo[1, 2], Haralambos Mouratidis[2], Djamel Khadraoui[1], Eric Dubois[1]
[1]Public Research Center Henri Tudor - 1855 Kirchberg/Luxembourg
{moussa.ouedraogo, djamel.khadraoui, eric.dubois}@tudor.lu
[2] School of Computing, Information Technology and Engineering, University of East London, England haris@uel.ac.uk

Abstract: Commonly, assurance is considered as "something said or done to inspire confidence". It is clear from this definition that the fundamental part of assurance is confidence. However, the level of confidence inspired from a statement or an action depends on the "quality" of its source. Inspired by the Systems Security Engineering Capability Maturity Model (SSE-CMM) and the Common Criteria, we tailored five ordinal levels of quality levels for probes performing the verification of system security measures; different levels of quality being possible depending on the coverage, rigor, depth and Independence of the verification. The metric taxonomy is intended to assist IT Products manufacturers in developing their products or systems and in identifying security requirements to be satisfied for their products or systems to be assured at some level of quality as far as assurance evaluation is concerned. It could also benefit consumers in supporting them in selecting IT security products depending on their organizational needs, while IT security evaluators may use it as reference when forming judgments about the quality of a security product.

1. Introduction

Nowadays, reliance on technology is increasing quicker than the ability to deal with the also increasing threats to information security. It is therefore important for stakeholders (Users, system administrators, database administrators, etc...) to know if their systems are susceptible to threats and if they can be trusted. Although, some of the stakeholders are not particularly interested in the details of the technology and how security solutions are deployed, they want assurance. In other words, they need some quantifiable evidence that the security measures put in place to countermeasure security risks have been correctly deployed and work as intended. In general, the verification of the correctness of in place security measures is performed by either security auditors or by dedicated software probes, as it is becoming more and more the case in this day and age for operational systems. In any case, it can be agreed that the level of expertise of the auditor plays a key role in the effectiveness of the audit and, so is the "quality" level of the software probe. This assertion calls for the elucidation of the quality levels that can be achieved and the requirements to be assured at a certain level of quality with respect to security assurance. This paper proposes to adapt the System Security Engineering Capability Maturity Model levels (SSE-CMM, 1999) to represent the possible levels achievable by a probe and some of the Common Criteria's families (ISO/IEC 15408, 2006) as quality requirements pertinent to assurance. As a matter of fact, the Common Criteria (CC) philosophy of assurance asserts that

greater assurance results from the application of greater evaluation effort, and that the goal is to apply the minimum effort required to provide the necessary level of assurance. The increasing level of effort is based upon:

1. **Coverage of the verification :** The effort is greater because a larger portion of the IT product is included in the verification
2. **Depth:** The effort is greater because it is deployed to a finer level of design and implementation detail.
3. **Rigor:** The effort is greater because the verification is applied in a more structured, formal manner.

Table 1 reviews the quality levels and their associated descriptions. The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 develops the quality metric taxonomy while in section 4 we highlight the relationship between confidence level and probe quality level. Section 5 concludes the paper.

The CC describes a framework in which developers can specify their security requirements and testing laboratories can evaluate the products to determine if they actually meet the claimed security. In other words, the CC provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard manner. Part 3 defines the assurance requirements both for the development environment and for the product itself as well as the tasks for the evaluator. These assurance requirements are organized in classes, then in families of components, which cover functional specification and design descriptions, testing, life cycle management, delivery procedures, security of the development environment, vulnerability analysis, etc. Developers can either build up their own consistent assurance package or use one of the seven predefined Evaluation Assurance Levels (EAL). EAL1 to EAL7 provide an increasing scale that balances the level of assurance obtained on the product security with the cost and feasibility of acquiring that degree of assurance. Another approach that was built upon the CC to probe the security of operational systems is BUGYO (Bulut et al., 2007). Unlike the

Capability Level 0 – Not Performed	The quality of the verification process is unknown
Capability Level 1 - Performed Informally	The verification process of the safeguards may not be rigorously undertaken nor planned and tracked. A human expert who relies on individual knowledge on the safeguard may perform it.
Capability Level 2 – structurally performed	A specific procedure for the evaluation is available and is carried out. The evaluation process conforms to specified standards and requirements with provision of appropriate tools to perform the process.
Capability Level 3 – Structured and Independent verification	Verifications are performed according to a well-defined process using approved standard or tools provided by third party.
Capability Level 4 – Semi complete verification	The verification follows a well-defined process with a usage of software tools that cover most of the relevant part of the security measure.
Capability Level 5 – Complete verification	The maturity of the verification is such that all known relevant part of the safeguard are investigated appropriately in depth as well as in breadth

Table 1. Probes quality levels and description

CC, BUGYO proposed five levels of assurance while casting doubt on the practical use of the Common Criteria's level 6 and 7. Nonetheless, one of the similarities between the two approaches lies on the fact that the metric taxonomy used is purely dedicated to working out the assurance level of an IT system or its components. Other taxonomy proposed in the literature includes Vaughn's (Vaughn et.al. 2003), Savola's (Savola, 2007) Seddigh's (Seddigh et al. 2004). Our work distinguishes itself from the above by proposing a metric taxonomy that aims at gauging the quality level of the assurance evaluating probe. This is relevant since a clear correlation exists between the evaluating probe quality and the result achieved. Highly qualitative probes will provide more accurate results which can be relied upon.

The next section is dedicated to the elucidation of the probe quality taxonomy.

3. Probe Quality Metric Taxonomy

3.1 Structure of the probe Quality metric Taxonomy:

The matrix shown in table 2 expresses the minimum requirements to achieve certain quality level.

Class	Family and meaning	Quality Level: QL				
		1	2	3	4	5
QAM: Probe Quality Metric	**QAM_COV: Coverage** (Larger coverage of the verified security measure provides more confidence on the results about its status)	1	2	2	2	3
	QAM_DPT: Depth (A detailed verification of the security measure will decrease the likelihood of undiscovered errors.)	1	2	2	3	4
	QAM_RIG: Rigor (The more structured the evaluation of the deployed security measure, the more reliable the outcome of the verification)	1	2	2	2	2
	QAM_IND : Independent Verification (verification performed by a third party evaluator or software tool provides more assurance)	1	1	2	2	3

Table2. Probe quality metric taxonomy

To that extent, considering *a probe, which quality evaluation provided results represented by I(coverage= x, Depth=y ,Rigor= z, Independence of verification= t), satisfies quality level k if all the parameters (Coverage, Depth, Rigor and independence of verification) capability for I are greater or equal to the corresponding parameters for* QL_k. The matrix indicates that in order for a probe to be at level 3 of quality for instance, at least the following requirements should be satisfied:

- *QAM_COV.2*
- *QAM_DPT.2*
- *QAM_RIG.2*
- *QAM_IND.2*

The rationale for the structure of the matrix in table is provided in section 3.2. The subsequent presentation of the probe quality classes follows the example set by the Common Criteria and more precisely the structure of the class *ATE: tests, which* emphasizes on confirmation that the Target Security Function or TSF (in Common Criteria terminology) operates according to its design descriptions. The ATE: Tests class

separates testing into developer testing and evaluator testing. The Coverage (ATE_COV) and Depth (ATE_DPT) families address the completeness of developer testing. Coverage (ATE_COV) addresses the rigor with which the functional specification is tested; Depth (ATE_DPT) addresses whether testing against other design descriptions (security architecture, TOE design, and implementation representation) is required.

Functional tests (ATE_FUN) addresses the performing of the tests by the developer and how this testing should be documented. Finally, Independent testing (ATE_IND) addresses evaluator testing: whether the evaluator should repeat part or all of the developer testing and how much independent testing the evaluator should do. We tailored these families to represent the characteristics of the evaluating probe by making the following mapping:

- **ATE_COV** subdivided into three capabilities and **QAM_COV**
- **ATE_DPT** subdivided into four capabilities and **QAM_DPT**
- **ATE_FUN** subdivided into two capabilities and **QAM_RIG**
- **ATE_IND** subdivided into three capabilities and **QAM_IND**

The metric construction class and the families associated are next described. For each family, a description of its dependencies and components are provided. Figure 1 describes the quality families and their associated capabilities. The components are hierarchical and, if not otherwise specified, higher-level components include the lower levels.

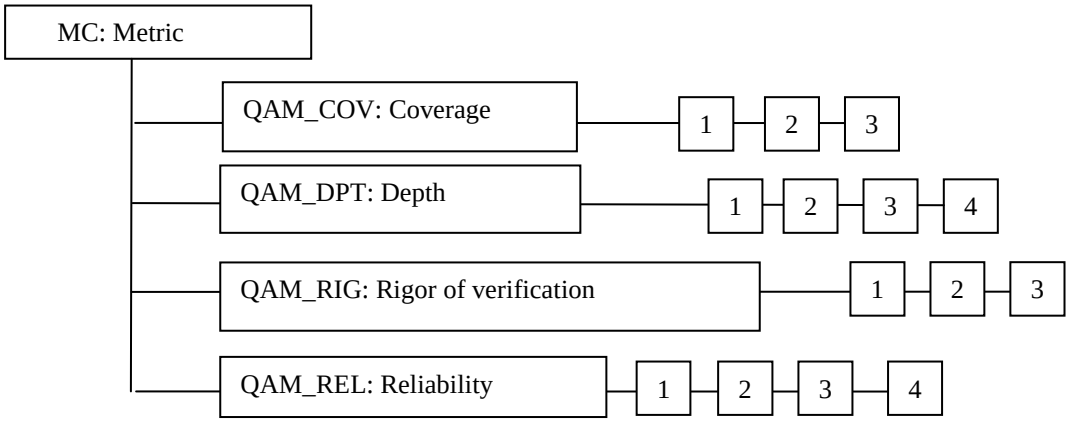

Figure1. Metric families and capabilities

- *QAM_COV: Coverage*

The coverage (scope) family indicates that the more elements of the security measure are verified by the probe, the more it can be assumed that the metric result represents the security measure. The objective of this component is to confirm that all relevant parameters of the security measure have been verified.

Component leveling

QAM_COV: Coverage	1	2	3

Dependency: QAM_DPT

The analysis of the coverage shall demonstrate that all key aspects of the security measure have been completely verified.

- ***QAM_COV.1 The verification process only targets specific areas of the deployed security measure not necessarily representative of its status.***
    - A part not formally estimated regarding its importance, which contribute to the security measure correct functionality, is verified.
- ***QAM_COV.2 Only some of the key areas of the security measure, known to be relevant for its well functioning are evaluated in the process***
    - A selection of the known important parts, as estimated by an expert, which contribute to the security measure correct functionality, are verified.
- ***QAM_COV.3 All relevant aspects of the deployed security measure are verified in the evaluation process***
    - All parts characterized as significant, by an expert, which contribute to the security measure correct functionality, are verified.

- *QAM_DPT: Depth*

The components in this family deal with the level of detail to which the security measure is verified by the probe and therefore minimizing the risk of missing an error in the security measure.

Component leveling

QAM_COV: Depth	1	2	3	4

Dependency: QAM_COV

- ***QAM_DPT.1: Evaluation of the security measure is done without a clear idea on how deep the verification is conducted.***
- ***QAM_DPT.2: High level verification of the security measure through its interface.***
- ***QAM_DPT.3: Most of the relevant modules of the* security measure are verified during the evaluation Process.**
- ***QAM_DPT.4: Detailed verification of the security measure is undertaken with the entire relevant modules assessed***

- *QAM_RIG: Rigor of verification*

The more structured the evaluation of the deployed security measure, the more reliable the outcome of the verification.

Component leveling

QAM_RIG: Rigor	1	2

- ***QAM_RIG.1: The verification is undertaken by a human expert who is familiar with the deployed security measures.***
- ***QAM_RIG.2: The verification process is structured and follows the requirements within a verification documentation or a standard.***

- The verification is performed by a software tool

- *QAM_IND: Independent Verification*

The more independent the evaluation of the deployed security measure, the more reliable the outcome of the verification. In another words, performing the verification of the security measure one has deployed with a self developed tools is a self assessment exercise which cannot be too reliable.

Component leveling

QAM_IND: Independent verification	1	2	3

Dependency: QAM_COV, QAM_DPT, QAM_RIG

- ***QAM_IND.1: Verification is performed by probe developed internally***
  - The verification is undertaken by a dedicated self-developed tool
- ***QAM_IND.2: partial verification by independent probes available on the market***
  - The verification is performed by a commercial or open source automated software tool but not all relevant parts of the security measure are verified.
- ***QAM_IND.3: Complete verification by independent probes available on the market***
  - The verification is performed by a commercial or open source automated software tool with all relevant parts of the security measure verified.

3.2 Justifying the Probe Quality Matrix Structure and the Minimum Requirement for Achieving a Quality Level:

The determination of the minimum requirement to satisfy a given quality level is made through consideration of the **definition of the quality levels themselves**, **the positive correlation between quality levels and the families capability levels** and finally **the maximum capability of each family**.

According to the definition of the quality level 1 (QL1), the evaluation process is not structured and may be performed by a human expert. This suggest that for that level the Rigor family should be at least at capability level 1(refer to QAM_RIG.1 description). When considering the definition of QL2 (structurally performed) one could see that a key element of improvement between QL1 and QL2 is that the verification becomes structured and at least a software tool is used, meaning that the Rigor family should be at least at level 2. Since the maximum capability level for that family is 2 and the fact that family's capabilities should be at least static when going up in quality level, the Rigor capability for QL3, QL4 and QL5 should be at capability level 2. We can therefore assume the following evolution trend for the Rigor family QAM_RIG from QL1 to QL5:

QAM_RIG: Rigor	1	2	2	2	2

One of the differences between QL2 and QL3 is that at the latter level, the verification is performed by a third party probe, making the verification more independent. Therefore the capability level for the Independent verification family (QAM_IND) is at least at 2 for QL3 and at level 1 for QL2. QL4 stipulates that the verification process is semi-complete i.e. although independent, the verification does not cover all the relevant parts of the security measure, which correspond to capability level 2 for the QAM_IND family. All these considerations, added to the correlation between the families capability

level and the probe quality level leads to the following evolution trend for the QAM_IND family from QL1 to QL5:

QAM_IND: Independent verification	1	1	2	2	3

QL5 is referred to as "complete" verification, meaning that all known relevant parts and module of the security measure have been verified. This implies that the QAM_COV and QAM_DPT should be at their maximum capabilities, 3 and 4 respectively. QL4 "semi-complete" verification implies that at least the most relevant parts and modules of the security measure have been verified, which correspond to at least QAM_COV.2 and QAM_DPT.3. The use of a software tool which is necessary for being at QL 2 would imply that the evaluator should know how deep the verification is being conducted and the components of the security measures concerned by the verification. This means QAM_DPT should be at least at capability level 2 for QL2. Similarly for QL3, QAM_DPT should be at least at level 2. The previous arguments hold for the QAM_COV family. The capability for that family for QL2 and QL3 should be at least 2. The capability evolution for QAM_COV and QAM_DPT from QL1 to QL5 are therefore as shown below:

QAM_COV: Coverage	1	2	2	2	3

QAM_DPT: Depth	1	2	2	3	4

4. Probe Quality Level and Confidence Level

The quality of an assurance evaluation probe as defined in this paper influences one's confidence in the accuracy of the verification result achieved by the probe and subsequently the security assurance value of a system or its component. The quality level (QL) of a probe serves as a cap to the confidence level one can expect from using a certain type of probe. Thus, the verification of the correctness of a deployed security measure will be assigned the value QL if the security measure posture is found to be compliant with the security requirements specification while zero "0" will be used to signify either that the compliance is a the lowest possible level or that the mismatch detected is critical for the system. Intermediate states will be assigned a discrete value within]0, QL[and classified depending on their gravity for the system.

While conducting a security assurance evaluation of a Domain Name Server (DNS) a Samhain probe (Samhain, 2008), an open source host-based intrusion detection system using cryptographic checksums of files to detect modifications, has been used for the verification. An effective functioning of that probe helps detect the address resolution files integrity being corrupted as a result of any malicious attack. Based on information obtained from the Samhain documentation and by comparing the quality metric taxonomy and the specification in table 2, we derived the following conclusions:

Coverage of the measures: The coverage of the Samhain measurements satisfies QAM_COV.2. In fact the measures only represent a static behavior of the service and not a dynamic network view (in and outgoing flows from and to the DNS server).

Depth of the measures: The measures undertaken by the Samhain target the address resolution file. This is good because a missing address resolution file or a bad content is relevant to the correct DNS behavior. This satisfies at least QAM_DPT.3.

Rigor of the measures: Samhain is a dedicated open source integrity check software tool. (QAM_RIG.2)

Independence of verification: A recent version (v2.4.5) was used with a continuous evolution of the dictionary. The main weaknesses of the DNS controlled. (QAM_IND.3)

The values of the Samhain capabilities do not explicitly correspond to any of quality level of table 1. Nonetheless, all the parameters of the Samhain (Coverage :2, Depth:3, Rigor:2 and Independent verification:3) are greater or equal to those of quality level 4, while some are lower than those of quality level 5. We can here conclude that the Samhain probe corresponds to quality level 4**.**

Taking into account the quality level of the Samhain (level 4) and the possible results obtain from the Samhain (detection of possible malicious change) and the self-developed script (configuration errors); the confidence level on the conformity of the DNS (depending on the gravity of the security breach in the expert view) can be summarized as follows: If the address resolution files integrity is compromised:

- Corrupted files: An evil-minded modification then the confidence on the conformity level is 0.
- In case of corrupted files and errors (configuration errors), the confidence on the conformity level is 1.
- Otherwise, if everything is fine the confidence on conformity level is 4.

5. Conclusion

In this paper, we have presented a probe quality metric taxonomy with respect to assurance evaluation. The probe quality taxonomy is part of a wider framework for the evaluation of operational systems security assurance that we developed.

Regarding future work we envisage the implementation of the taxonomy so to enable an automatic decision on probes quality levels. A wider application of the taxonomy on more and diverse type of probes is plan to judge on its effectiveness and also for possible enhancement.

6. References:

Bulut E., Khadraoui D., and Marquet B., " Multi-Agent based security assurance monitoring system for telecommunication infrastructures", In proceedings to the Communication, Network, and Information Security conference, Berkely/California 2007.

ISO/IEC 15048, Common Criteria for information Technology, part 1-3, version 3.1, September 2006.

Samhain, http://www.la-samhain.de/samhain [Accessed: 10 March 2008]

Savola, R.M., "Towards a Taxonomy for Information Security Metrics", International Conference on Software Engineering Advances (ICSEA 2007), Cap Esterel, France.

Seddigh N., Pieda P., Matrawy A., Nandy B., Lambadaris L. and Hatfield A., "Current Trends and Advances in Information Assurance Metrics", In proceedings of PST 2004: 197-205

SSE-CMM: System Security Engineering Capability Maturity Model, April, 1999.

Vaughn R.B., Henning R., Siraj A., "Information Assurance Measures and Metrics – State of Practice and Proposed Taxonomy", in Proceedings of the IEEE/HICSS'03, Hawaii.