A Visualization and Modeling Tool for Security Metrics and Measurements Management

Reijo M. Savola
VTT Technical Research Centre of Finland
Oulu, Finland
Reijo.Savola@vtt.fi

Petri Heinonen
VTT Technical Research Centre of Finland
Oulu, Finland
Petri.Heinonen@vtt.fi

***Abstract*—Sufficient and credible information security measurement in software-intensive systems requires use of a variety of security metrics offering security-related evidence from different viewpoints. Visualization is needed to facilitate management of security metrics and measurements and to increase the meaningfulness of them in decision-making such as security assurance and risk management. We introduce a novel visualization and modeling tool for hierarchical specification and deployment of security metrics and measurements. The tool connects high-level risk-driven security objectives with detailed measurements and evidence gathering. The tool facilitates the management of a large number of metrics and measurements without losing appropriate granularity that is crucial for informed security decision-making.**

Keywords-security metrics; visualization; security assurance; risk management

I. Introduction

Systematic and practical approaches to measuring information security are needed to be able to offer sufficient and credible evidence of the security level and performance of software products, services or an organization. The foundational quality criteria of security metrics are *correctness*, *measurability* and *meaningfulness*. All these criteria are critical for the practical applicability of security metrics and measurements. Correctness can be ensured by a well-established metrics development methodology that includes validity and reliability analysis. Measurability of security metrics can be enhanced by developing a suitable measurement framework to the system under investigation and promoting the implementation of *measurability-enhancing mechanisms* [1].

The meaningfulness of security metrics and measurements in security decision-making, such as security assurance and risk management, carried out by humans is often a remarkable challenge. Many security metrics activities suffer from poor meaningfulness. Although security metrics collections can be applicable by themselves, their meaningfulness is constrained by their poor manageability due to the vast amount of uncategorized information. In general, collections with a low number of metrics with aggregated values are more understandable by human decision-makers than wider collections. Consequently, in the history of security metrics activities, the primary aim has been to develop only a few metrics. However, a lot of essential security-relevant information is lost in the process of aggregation of lower-level metrics results. It is more desirable to maintain a more complete collection of security metrics and measurements for better granularity of security-related evidence. Well-designed visualization tools and methods enabling better management of collections of metrics and measurements are needed.

This study's primary contribution is in introducing a novel visualization and modeling research prototype tool called Metrics Visualization System (MVS) for security metrics and measurements management. The tool is able to present decompositions of security objectives which are the basis for security metrics development. Metrics decomposition starts at the top from security objectives and security controls developed based on them, and is carried out down to the level of practical measurements. The detailed metrics and measurements at the leaf level are mapped to infrastructure components of the System under Investigation (SuI).

Section II discusses the background of security metrics and measurements and visualization. Section III discusses how hierarchical presentation of security metrics increases the meaningfulness of measurements, and summarizes the main needs for visualization. Section IV presents the security metrics modeling concepts of the MVS with a simplified example metrics hierarchy. Section V presents related work, before Section VI offers some conclusions and poses future research questions.

II. Background

The term *security metrics* refers to the interpretation of measurements of the security performance (or in other words, security effectiveness, efficiency and correctness) and level, and security indicators or security strength of a SuI [2] – a technical system, product, service or organization. The complexity, lack of common definitions and the difficulty of predicting security risks, along with their dynamic nature, make it impossible to measure security as a universal property. Consequently, terms such as *indicators* or *strength* might be more appropriate in the case of security-related objectives [3]. In this study, however, we use the most widely-used term, *metrics*. The goal of security measurements and the development of security metrics, is to be able to obtain and manage evidence about the security level and performance from the SuI. It is desirable that this security-related evidence is utilized in a proactive way, e.g., during the research and development phases of the SuI. Examples of security metrics application areas include risk management, comparison of security solutions, (software) security assurance, security testing, and security monitoring [4].

Security controls are a means of managing risk, which can be administrative, technical, management, or legal in nature [5]. In relation to security controls, there are three fundamental objectives of security measurement: (*i*) correctness, (*ii*) effectiveness, and (*iii*) efficiency of deployed security controls [2]. Of these, sufficient effectiveness is obviously the main goal of security activities. In practice, a balanced tradeoff is needed between effectiveness and efficiency. Correctness of security controls is a necessary *but not sufficient requirement* for effectiveness. Correctness can be investigated, e.g., with respect to requirements, regulations, legislations, and standards. Efficiency measurement is not within the scope of this study. Information Technology Security Evaluation Criteria (ITSEC) [6] originally made the distinction between correctness and effectiveness assurance. In software-intensive systems, in addition to security control –related measurements, security metrics can concentrate on vulnerability management.

In general, the state-of-the-art security metrics are still quite underdeveloped. Savola [7] lists some reasons for the situation: (i) security is often considered as "add-on" property, (ii) the security research field itself is in its infancy, (iii) there is a lack of suitable data to be used in metrics development. An important consideration is the lack of meaningful tools for the management of security evidence. Metrics are being developed to be able to make justified statements about reality, which is not measurable in its entirety. Consequently, every metric is a simplification: it has a much lower information dimensionality than reality. However, there are means to improve the real-world solutions based on the information offered by the measured data. The used measurement approach is very effective if the appropriate reaction enhancing the security level is the result from the measurements [7]. Surveys of security metrics approaches can be found in [7-12].

Security-measurability-enhancing mechanisms are crucial for practical security measurements, because they reduce the effort needed for gathering relevant security evidence and ensure the availability and attainability of it. Savola and Heinonen list, analyze and discuss solutions for the following security-measurability-enhancing mechanisms for software-intensive systems in [1]: flexible communication mechanism, security measurement mirroring data redundancy, auto-recovery on error, multi-point-monitoring, integrity and availability checks, timing framework, measurability support of building blocks of security, use of shared metrics and measurement repositories, reuse of available metrics and measurements relevant to security and secure coding.

Visualization is the graphical representation of data or concepts [13]. Vision is the most valuable sense for providing information from computers to humans because humans acquire more information through vision than through all the other senses combined [13]. Card et al. [14] propose six major methods in which visualization can amplify cognition by perception: (i) increasing memory and processing resources by allowing storage of massive amounts of information in a quickly accessible form, (ii) reducing searching by grouping information together, (iii) enhancing recognition of patterns by enhancing patterns, (iv) perceptual inference by making some problems obvious, (v) perceptual monitoring by allowing monitoring of a large number of potential events, and (vi) manipulable medium by allowing exploration of a space of parameters unlike static diagrams.

Adequate visualization of security metrics can help us interpret abstract and complex data, that is inherent to security issues, and form a mental image of them. Because research of security metrics is still in its infancy, visualization of security metrics and measurements has not been studied much before, and there is a lack of tools and frameworks supporting it. However, general visualization tools are used for the management of different kinds of metrics, sometimes also for security metrics. These tools do not well support security metrics meaningfulness goals, lacking support for hierarchical security metrics modeling.

III. Hierarchical Presentation of Security Metrics and Measurements and Needs for Visualization

Security risks and their impact on the SuI often remain too abstract from the perspective of software-intensive systems development. On the other hand, a lot of detailed security-related information is available yet rarely utilized because their relation to the actual security objectives is not understood. Evidently, a large number of security metrics are needed for sufficient understanding of the security level. The relations between high-level security objectives, specified based on the results of risk analysis, and the detailed level measurements are difficult to understand without hierarchical presentation. Moreover, compliance to standards and best practice specifications often need to be shown. In the following, we discuss decomposition of security objectives and aggregation of security metrics results, and the needs for visualization of security metrics. Security objectives form an important basis for security metrics development. The effectiveness of security controls is determined by continuous risk analysis, carried out in a more frequent cycle compared with business-level risk management activity [3]. At the architectural level, operational risk analysis can be implemented by Architectural Risk Analysis (ARA) [15], which is sometimes referred to as threat modeling [16] or security design analysis.

Hierarchical presentation of security metrics is based on the decomposition of security measurement objectives. Savola and Abie [17, 18] developed and analyzed collection of security metrics to measure the correctness and effectiveness of security controls. The aforementioned research [17] introduced an iterative methodology for security metrics development, which has been simplified here: (1) Carry out prioritized ARA of the SuI; (2) Utilize suitable security metrics taxonomies and/or ontologies to further plan the measurement objectives and metrics types; (3) Develop and prioritize security objectives; (4) Identify Basic Measurable Components (BMCs) from the security requirements using a decomposition approach. BMCs are leaf components of the decomposition that clearly *manifest a measurable property* of the system. Similarly, decompose the system architecture to components; (5) Define measurement architecture and evidence collection**.** Match the BMCs with the relevant system components with measurable data attainable; (6) Integrate metrics from other sources and select BMCs based on feasibility analysis; and (7) Develop appropriate balanced collection of metrics from the BMCs. The BMCs are identified by security objective decomposition [17, 18], based on the original idea proposed by Wang and Wulf [19].

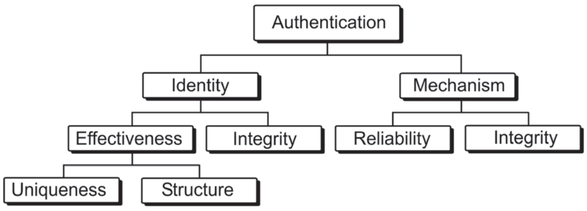


Figure 1. An example authentication decomposition [19]

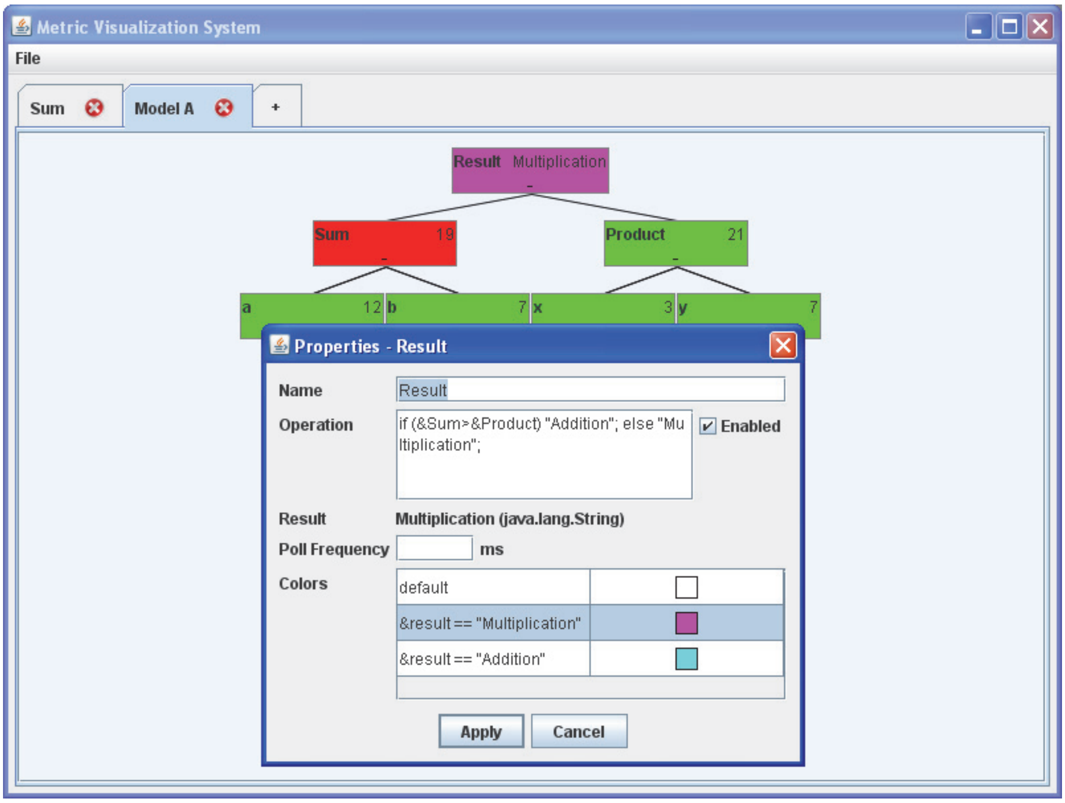
Figure 1 provides an example of high-level decomposition of authentication objectives. The BMCs are Authentication Identity Uniqueness (*AIU*), Authentication Identity Structure (*AIS*), Authentication Identity Integrity (*AII*), Authentication Mechanism Reliability (*AMR*) and Authentication Mechanism Integrity (*AMI*). The decomposition is based on the fact that identity concept and authentication mechanism contribute essentially to the security strength of authentication. A more detailed explanation of the listed BMCs can be found in [17]. The boxes in Figure 1 can be seen as *metrics nodes*, which are associated with logical expressions on security-relevant parameters. Note that the leaf nodes of Figure 1 are BMCs, not measurements. Metrics and measurement are further refined from the BMCs, and the leaf nodes resulting from that refinement contain *raw measurements* with security-relevant information. The raw measurements can contain different scale types, e.g., string values of chosen algorithms, counts of successful and unsuccessful authorization attempts, and penetration test results. Special *measurement probes* can be developed to obtain this kind of information from software systems. A measurement probe is a tool for performing checks on infrastructure objects of a software system in order to provide the required information for the purposes of security metrics. The scale type of security metrics can be nominal, ordinal, interval or ratio. Moreover, confidence values, e.g., in the range from 0 to 1 can be attached to each metric.

Security metrics can be aggregated in the form of a weighted sum. In the previous example of authentication, the following formula can be used to aggregate the BMCs in Authentication Strength *AS* [17]:

$$\begin{aligned} AS = {} & w_{AIU}\cdot\overline{AIU} + w_{AIS}\cdot\overline{AIS} + \\ & w_{AII}\cdot\overline{AII} + w_{AMR}\cdot\overline{AMR} + w_{AMI}\cdot\overline{AMI}, \end{aligned} \tag{1}$$

to quote, “where w_x is the weighting factor of component x and ‘¯’ denotes normalization and uniform scaling of the component metrics.” The scale types of the leaf metrics need to be normalized. Other ways of aggregation schemes and weightings are also possible, depending on the risk prioritization.

The meaningfulness of security metrics and measurements are affected by the aggregation: important details of security-



Name: Result

Operation: if (&Sum>&Product) "Addition"; else "Multiplication";

☑ Enabled

Result: Multiplication (java.lang.String)

Poll Frequency: ms

Colors:

default	□
&result=="Multiplication"	🟪
&result=="Addition"	🟦

Figure 2. Different colors are used to visualize the results and to offer better meaningfulness of aggregation of security measurement results

relevant information are often lost when values of sub-components are merged into one result. *Plain aggregation over-simplifies security-related issues.* Moreover, practical experience has shown that the aggregated results of security measurements tend to show too optimistic outcomes, or using traffic light terminology, they tend to be green.

When designing the visualization solution, it should be remembered that humans can only observe detailed changes in one object at a time, and yet we still look at our surroundings with the impression that we see all the objects and their details simultaneously [20]. The main needs for security metrics visualization can be summarized as follows, based on the above discussion:

- **Structured security metrics entities, "building blocks":** The metrics and measurements should be presented by nodes in which it should be possible to associate logical expressions. In measurements, results represented by plain values are sufficient.
- **Meaningful metrics relationship modeling:** the relationships between security objectives and low-level measurements should be visible. This can be achieved by hierarchical presentation of metrics nodes formed by decomposition of security objectives.
- **Alleviation of the metrics aggregation oversimplification challenges by visualization:** It should be possible to investigate measurement details without losing the manageability of a large collection of security metrics and measurements. To increase the usability of security metrics, visualization solution should enable to investigate both aggregated and non-aggregated metrics and measurements.
- **Measurement probe and security-measurability-enhancing mechanism support:** Connection to measurement probes offering raw measurements should be available to the visualization system to support automation of evidence gathering. Moreover, enough support for measurability-enhancing mechanisms are needed.

IV. Metrics Visualization System

In the following, we briefly present the MVS visualization platform, discuss the security metrics and measurements specification and visualization functionalities of it, and propose how the tool can be used to support security-related decision-making.

The MVS is a graphical security metrics modeling environment enabling specification of security metrics and measurements based on *Security Metrics Nodes* (SMN). It offers an initial solution towards meeting the visualization needs listed above: structured security entity concept, metrics relationship modeling, alleviation of metrics aggregation oversimplification challenges, and measurement probe and security-measurability-enhancing mechanism support. It should be noted that the tool is a research prototype. It can be extended by more advanced graphical features, e.g., the display of different types of diagrams. Note that MVS is a tool for security *metrics* modeling, not for security modeling in general.

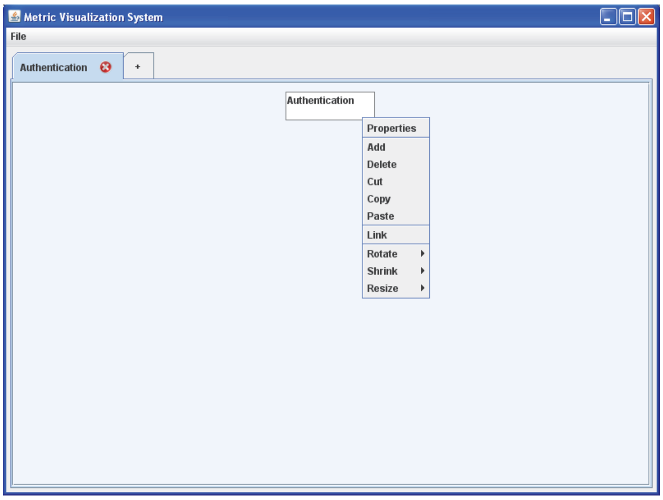


Figure 3. Pop-up menus can be opened from each SMN

A. *Visualization Platform*

The MVS Visualization Platform consists of the following entities: a hierarchy modeling engine, properties, a logical operations module and a Representational State Transfer (REST) [21] based communication module. These components implement the core functionality of the tool, including saving the current state of the metrics model and saving the measured data. The Visualization Platform supports extensibility well because of the REST communication module and logical operations module implementation. Communication between the MVS and SuI is implemented with the REST interface. Logical operations can be linked with the data gathered from the SuI through REST. A JavaScript engine is used in interpreting logical operations. This allows the user to freely create inference rules, the only restriction being JavaScript syntax. Values come from the results of the sub-component metrics' operations. Operation can also be a static value or direct measurement from the SuI.

B. *Security Metrics Presentation: Metrics Models and Nodes*

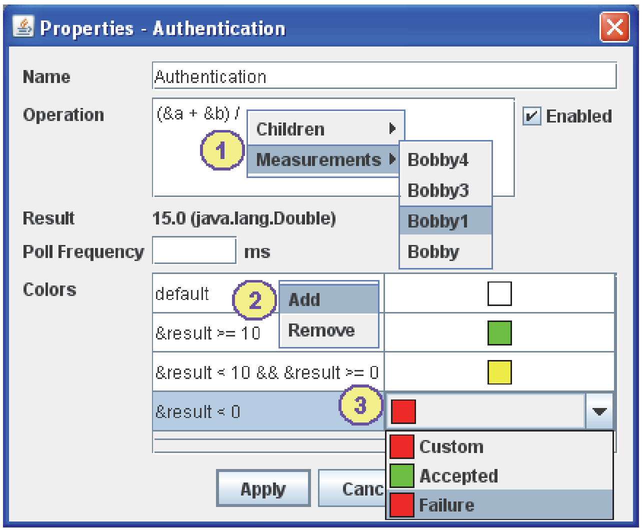
The MVS is used to specify, visualize and manage a *security metrics model* SMM. The basic building block of a metrics model in the MVS is a *security metrics node* SMN. In an SMM, SMNs form a metrics hierarchy, connected with relationship arrows. All SMNs in the model have the same default property fields:

- distinctive name,
- confidence value of the metric/measurement (range 0…1),
- operation specification (logical expression),
- threshold criteria and associated visualization,
- poll frequency field for automated measurements, and

- enable/disable flag for operation value evaluation.

Each SMN can be configured to represent (i) a security metric, (ii) an aggregated security metric, or (iii) a raw security measurement.

In the MVS, security metrics can be associated with logical operations using JavaScript scripting language. In addition, it is possible to set thresholds. The outcome is visualized by associating colors to different value ranges of the operation specified in the SMN. An aggregated metric represents more general security-relevant information compared to its sub-component metrics. A raw security measurement SMN is used to input different types of security measurement data to the model. Moreover, measurement probes can be associated with it.


Properties - Authentication

Name: Authentication

Operation: (&a + &b) /

☑ Enabled

Result: 15.0 (java.lang.Double)

Poll Frequency: ms

Colors:

default	□
&result >= 10	🟩
&result < 10 && &result >= 0	🟨
&result < 0	🟥

Figure 4. Properties of each metrics node can be edited in a window opened from the pop-up menu

C. Coloring of Nodes

All SMNs can be colored (or left blank). Coloring is important from the metrics management perspective. In security metrics hierarchies, coloring makes it possible to track the status of a large number of metrics in the same view. Using the Card et al.'s [14] terminology, coloring helps especially in reducing the searching by grouping information together and enhancing recognition of patterns by humans. The default coloring scheme of the MVS imitates traffic lights: red stands for insufficient level, yellow for intermediate level, and green for sufficient level. It is possible to add any number of coloring rules for each SMN.

Figure 2 shows an example of how coloring can be used to visualize value ranges. The operation in the figure selects the larger of the values of the sub-metrics representing addition and multiplication and outputs label "Addition" or "Multiplication" according to the comparison. Finally, a color is assigned to the SMN according to the operation result using JavaScript based coloring rules.

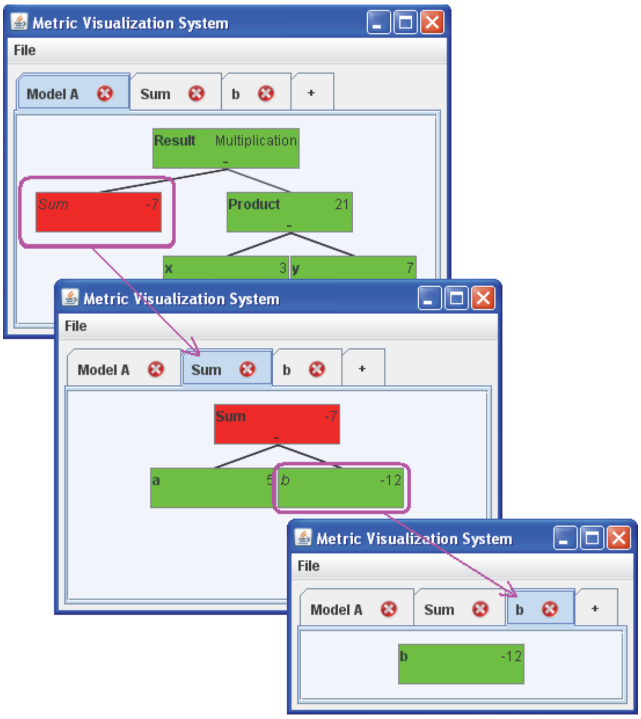


Figure 5. Sub-component SMNs and linking

D. Security Metrics Nodes Properties

The SMNs can be specified and managed by using a pop-up menu (see Figure 3). Using menu actions, it is possible to add, delete, copy, cut and paste nodes, and create links between different model diagrams. Moreover, it is possible to scale and rotate their graphical box form.

A property editing window is opened from the pop-up menu, See Figure 4. (1) The actual security metrics are expressed here in the "Operation" box using JavaScript. The actions associated with the numbers shown in the figure are as follows. (i) Results of the child nodes and measurement value can be used as variables in the operation. (ii) Thresholds for different values of the operation can be set, along with colors for their visualization. Only the default color is compulsory, otherwise any number of color rules can be added. The colors can be selected from a wide color palette. (iii) The thresholds are also expressed using JavaScript. Moreover, if the SMN is used to poll information from a measurement probe, the poll frequency is set from here.

E. Saving and Linking SMMs

The SMM can be saved to a file. MVS uses XML format; it is also possible to read and edit SMM files with external XML editors, or with web browsers. The SMM can be split into several files using linking, as Figure 5 shows. Linking is very useful when the SMM is built from logically separated sub-models. When using the same sub-model in several SMMs, linking can be used to decrease duplication. A linked SMN obtains its values from the root SMNs of linked SMMs (sub-models). When opening an SMM which has links to other

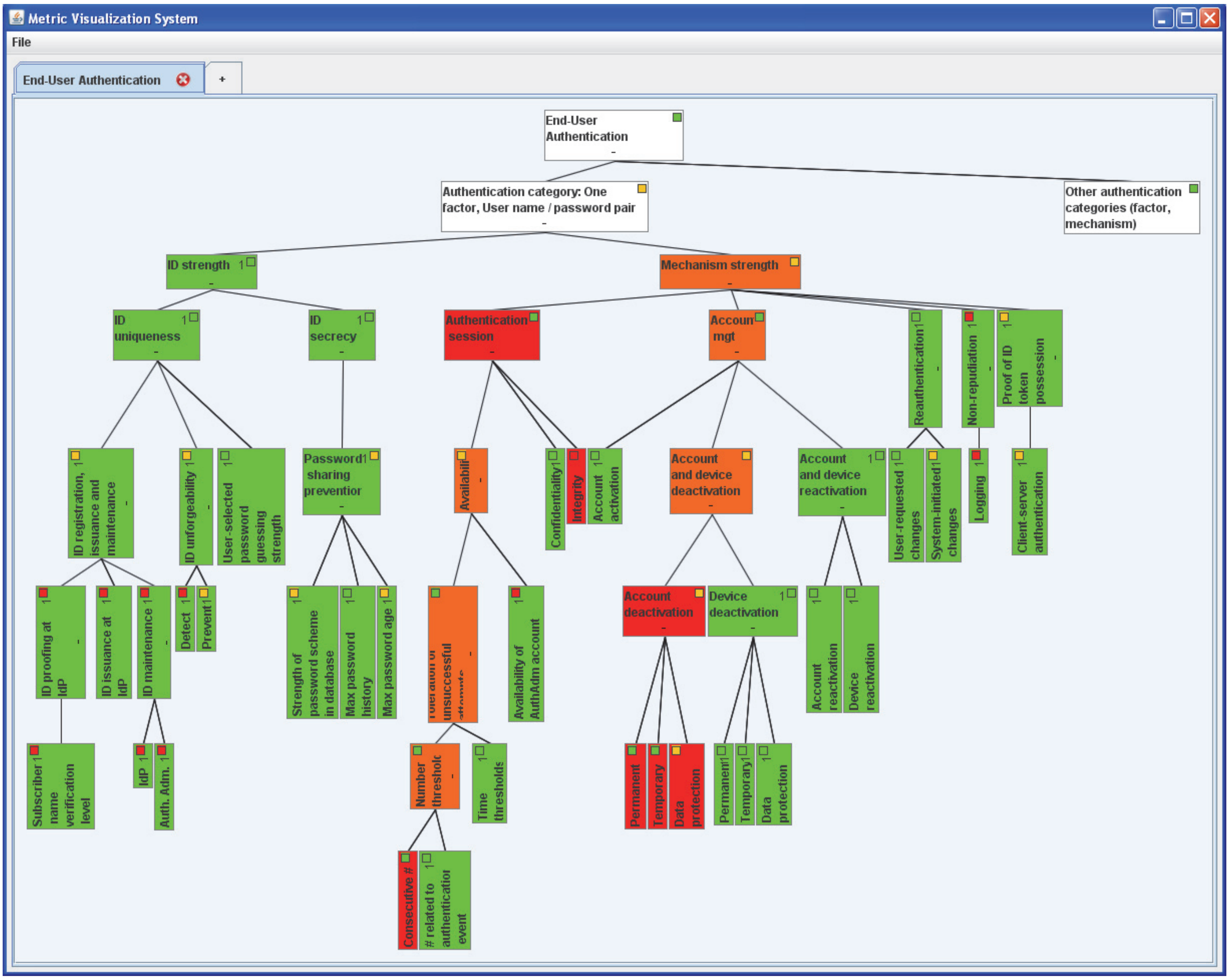


Figure 6. Example of a security metrics hierachical presentation (End User Authentication) with a specific coloring scheme

models, the linked models are opened at the same time. Only one instance of the model that is saved to a file can be opened at a time. As Figure 5 displays, values and colors from sub-component SMNs are propagated to the parent SMN. An SMN with a link to another SMM is labeled with italic font.

F. *Interface to Measurement Probes and Security-Measurability-Enhancing Mechanisms*

A specific interface for specific measurement probes was implemented to the MVS. The interface implements the connection to specific measurement probes that are part of an adaptive distributed measurement framework described in [22] and [23]. However, the tool can be extended with different kinds of measurement probes. It should be noted that availability and attainability of measurements will evolve over time, e.g., when new probes (or new versions of old probes) become available or existing ones are removed. The changes are managed in the MVS using the leaf SMNs property editing and proper communication solutions between the MVS and the SuI.

MVS supports the following security-measurability-enhancing mechanisms:

- a flexible communication mechanism by the REST communication module,
- multi-point-monitoring: easy interconnection of several SMMs,
- use of shared metrics and measurement repositories: XML file structure enables use of other XML-based metrics files,
- reuse of available metrics and measurements relevant to security and secure coding: easy interconnection of several SMMs.

G. *Security Metrics Model Example*

Figure 5 depicts a simplified yet realistic SMM for end-user authentication metrics. In the example metrics hierarchy, the focus is on a user name/password pair based authentication mechanism. However, a blank SMN is included for other authentication categories which can be applicable later during the process of analysis. In fact, the SMM supports comparison of the security performance of different authentication categories, provided that relevant sub-component metrics are included in the SMM.

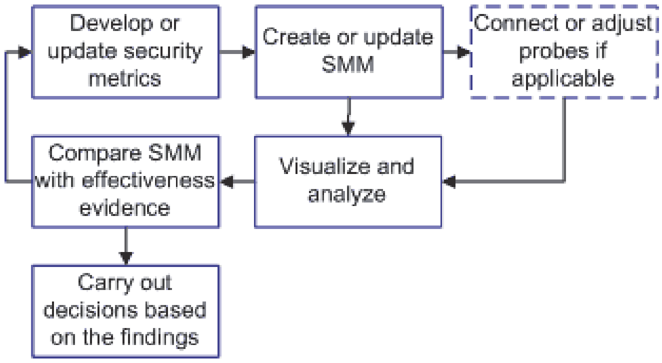

Develop or update security metrics
Create or update SMM
Connect or adjust probes if applicable
Compare SMM with effectiveness evidence
Visualize and analyze
Carry out decisions based on the findings

Figure 7. Proposed process for security effectiveness assessment

Following the decomposition examples of Figure 1, the main branches of authentication decomposition are identity strength and authentication mechanism strength. The metrics in the identity strength branch are based mostly on the reference architecture discussed in the U.S. National Institute of Standards and Technology (NIST) document, the Electronic Authentication Guideline [24]. The ID strength branch contains two sub-branches: ID uniqueness and ID secrecy. The ID uniqueness sub-branch contains sub-metrics for ID registration, ID unforgeability and user-selected password guessing strength, focusing both on detection and prevention. The ID secrecy branch contains sub-metrics for the strength of password scheme in a database, maximum password history and maximum password age. Note that '+' in the SMN denotes that the hierarchy can be further opened, and '-' means that the hierarchy in question is opened. The mechanism strength sub-branches are opened in the figure.

Confidence in the metrics or measurements in the SMN is shown in a small box in the upper right-hand corner of each SMN using traffic-light notation.

By coloring the different branches and implementing a color aggregation mechanism, it is possible to see the status of larger security-relevant entities and detailed metrics values at the same time. In the aggregation, either positive or pessimistic thinking can be applied: the aggregated color can be chosen to be an "average" color of the sub-values, or the lowest value can be copied upwards. In some cases, being more pessimistic is a better choice, while in others it is not. For example, in the mechanism branch, if there are fatal implementation metrics values, such as, the account and device deactivation not being configured adequately, it is better to copy red values upwards in the metrics hierarchy.

H. *MVS in Interpretation of Security Effectiveness Evidence*

Figure 7 proposes a process how to use the MVS in security effectiveness assessment. There are various factors which enable security effectiveness of the target under investigation: security-relevant configuration correctness, correct design, implementation and deployment of security controls and proper security testing activities. Metrics to investigate the different factors should be part of the security metrics hierarchy. Different SMMs can also be generated for different types of factors. Metrics representing the enabling factors can be compared with the security effectiveness evidence, such as results from penetration testing (during research and development), and from incidents (during the operation and maintenance). Moreover, vulnerability information found from open databases can be considered to be security effectiveness evidence. By using the MVS, it is possible to detect shortcomings in security correctness in a systematic way, compare the metrics results with security effectiveness evidence, and base decision-making on this evidence. For instance, if penetration tests show failures in the authentication mechanism, the types of failures can be associated with the relevant SMNs in the hierarchy, and the configuration, architecture, implementation or deployment of the relevant part of the mechanism can be fixed.

V. Related Work

The related work lacks security *metrics* visualization tools being able to increase the meaningfulness of metrics connecting high-level security objectives and low-level measurements. However, there are tools and frameworks available for (i) the visualization of security-related data and (ii) the visualization of information in general. In the following, we discuss these categories of contributions.

Security visualization is often used in analyzing huge security-related logs or traffic. However, the field is very young. Davix [25] is a collection of 25 different security visualization tools, allowing, e.g., the building of maps from pcap files and map protocol use in real-time across a network. Davix is based on the SLAX Linux operating system [26]. Interactive Quality Visualization (IQVis) [27] tool is designed to visualize the variability of software quality attributes, including security, at run-time. In IQVis, nodes of the model represent infrastructure components, such as devices, sensors and smart agents. PortVis [28] is a tool for visualizing security-related network data. It is useful especially as a port scanning application. SnortView [29] uses Network-based Intrusion Detection Systems (NIDS) logs to present security alert information. Being designed to monitor run-time behavior, these tools can potentially be connected to the MVS, offering input to the raw measurements. Marty [30] widely discusses security visualization approaches and available tools. A lot of research has been conducted in the field of information visualization in general, and a variety of tools and frameworks have been developed to visualize different properties. Software visualization is used to analyze artifacts related to software and its development. Evolve [31] was originally developed to visualize the run-time behavior of Java programs, but can also be used as a standalone tool. Prefuse [32] is a toolkit for interactive information visualization. Streamsight [33] is a visualization tool for large-scale streaming applications.

VI. Conclusions and Future Work

We have introduced a modeling and visualization tool called MVS for the management of security metrics and measurements in software-intensive systems. The tool increases the meaningfulness of metrics in the contexts of security assurance and risk management by hierarchical metrics modeling, connecting high-level security objectives with detailed measurements. In addition, the tool reduces human visual searching by grouping security level information together and enhances recognition of patterns by metrics value aggregation coloring schemes. The tool can also be utilized for visualization activities other than security measurement, but has been especially developed to alleviate meaningfulness challenges involved in the use of security metrics and measurements. The tool clearly increases usability of security metrics approaches.

Development of the MVS tool is still at initial research stage and we intend to enhance it with different types of visualization *views* facilitating better understanding of different types of security-related evidence, e.g. security effectiveness, correctness, efficiency, or different categories of compliance. Moreover, more alternatives to present the measurement results are planned.

References

[1] R. Savola and P. Heinonen, “Security-measurability-enhancing mechanisms for a distributed adaptive security monitoring system”, SECURWARE ’10, Venice/Mestre, Italy, Jul. 18–25, 2010, pp. 25–34.

[2] R. Savola, “A security metrics taxonomization model for software-intensive systems,” Journal of Information Processing Systems, Vol. 5, No. 4, Dec. 2009, pp. 197–206.

[3] R. Savola, H. Pentikäinen and M. Ouedraogo, ”Towards security effectiveness measurement utilizing risk-based security assurance,” Proceedings of the 2010 Information Security for South Africa (ISSA 2010) Conference, August 2-4, 2010, Sandton, South Africa, 8 p.

[4] R. Savola, “A taxonomical approach for information security metrics development,” NORDSEC ’07, 2007.

[5] ISO/IEC 27000:2009: Information technology – Security techniques – Information security management systems – Overview and vocabulary. ISO/IEC, 2009.

[6] Information Technology Security Evaluation Criteria (ITSEC), Version 1.2, Commission for the European Communities, 1991.

[7] R. Savola, “On the feasibility of utilizing security metrics in software-intensive systems,” International Journal of Computer Science and Network Security, Vol. 10, No. 1, Jan. 2010, pp. 230–239.

[8] D. S. Herrmann, “Complete guide to security and privacy metrics – measuring regulatory compliance, operational resilience and ROI,” Auerbach Publications, 2007, 824 p.

[9] A. Jaquith, “Security metrics: replacing fear, uncertainty and doubt,” Addison-Wesley, 2007.

[10] N. Bartol, B. Bates, K.M. Goertzel, and T. Winograd, “Measuring cyber security and information assurance: a state-of-the-art report,” Information Assurance Technology Analysis Center IATAC, May 2009.

[11] V. Verendel, “Quantified security is a weak hypothesis: a critical survey of results and assumptions,” New Security Paradigms Workshop, Oxford, U.K., 2009, pp. 37–50.

[12] R. Savola, “A novel security metrics taxonomy for R&D organisations,” ISSA ‘08, 2008, pp. 379–390.

[13] C. Ware, “Information visualization: perception for design”, 2nd Ed. Morgan Kaufmann Publishers, San Francsico, CA, 580 p.

[14] S.K. Card, J.D. Mackinlay and B. Shneiderman, “Readings in information visualization: using vision to think”, Morgan Kaufmann Publishers, San Francisco, CA, 686 p.

[15] G. McGraw, “Software security – building security in,” Addison-Wesley, 2006.

[16] M. Howard and D. LeBlanc, “Writing secure code,” Microsoft, 2003.

[17] R. Savola and H. Abie, “Development of measurable security for a distributed messaging system,” International Journal on Advances in Security, Vol. 2, No. 4, 2009, pp. 358–380 (March 2010).

[18] R. Savola and H. Abie, “Identification of basic measurable security components for a distributed messaging system,” SECURWARE ’09, Athens/Glyfada, Greece, Jun. 18–23, 2009, pp. 121–128.

[19] C. Wang and W. A. Wulf, “Towards a framework for security measurement”, 20th National Information Systems Security Conference, Baltimore, MD, Oct. 1997, pp. 522–533.

[20] E.H. Chi, “A taxonomy of visualization techniques using the data state reference model”, IEEE Symposium on Information Visualization 2000, Oct. 9–10, 2000, Salt Lake City, UT.

[21] R.T. Fielding and R.N. Taylor, “Principled design of the modern web architecture”, ACM Transactions on Internet Technology, 2 (2): 115–150.

[22] T. Kanstrén and R. Savola, “Definition of core requirements and a reference architecture for a dependable, secure and adaptive distributed monitoring framework”, DEPEND ’10, Venice/Mestre, Italy, Jul. 18–25, 2010, pp. 154–163.

[23] T. Kanstrén, R. Savola, A. Evesti, H. Pentikäinen, A. Hecker, M. Ouedraogo, K. Hätönen, P. Halonen, C. Blad, O. López and S. Ros, “Towards an abstraction layer for security assurance measurements (Invited Paper)”, Proc. of ECSA ‘10: Companion Volume, pp. 189–196.

[24] W.E. Burr et al., “Electronic authentication guideline,” National Institute of Standards and Technology, U.S. Department of Commerce, NIST Special Publication 800-63-1, Draft, Dec. 8, 2008, 97 p.

[25] SecViz – Security Visualization. http://www.secviz.org/node/89 [Accessed: April 29, 2011].

[26] Slax – Your Pocket Operating System. http://www.slax.org [Accessed: April 29, 2011].

[27] J. Kuusijärvi, “Interactive visualization of quality variability at run-time”, VTT Technical Research Centre of Finland, VTT Publications 746, 111 p.

[28] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti and M. Christensen, ”PortVis: A tool for port-based detection of security events”, VizSEC/DMSEC ’04, Oct. 29, 2004, Washington, D.C., pp. 73–81.

[29] H. Koike and K. Ohno, “SnortView: Visualization system of snort logs”, VizSEC/DMSEC ’04, Oct. 29, 2004, Washington D.C., pp. 143–147.

[30] R. Marty, “Applied security visualization”, Addison-Wesley, 2008, 552 p.

[31] Q. Wang, W. Wang, R. Brown, K. Driesen, B. Dufour, L. Hendren and C. Verbrugge, “Evolve: an open extensible software visualization framework”, SoftVis ’03, June 11–13, 2003, San Diego, CA.

[32] J. Heer, S.K. Card and J.A. Landay, “Prefuse: A toolkit for interactive information visualization”, CHI ’05, April 2–7, 2005, Portland, OR, pp. 421–430.

[33] W. de Pauw, H. Andrade and L. Amini, “Streamsight: A visualization tool for large-scale streaming applications”, SoftVis ’08, Sept. 16–17, 2008, Ammersee, Germany, pp. 125–134.