A Security Metric Based on Security Arguments

Benjamin D. Rodes, John C. Knight
Computer Science Dept.
University of Virginia
Charlottesville, VA, USA
{bdr7fv,jck}@virginia.com

Kimberly S. Wasson
Dependable Computing, LLC
Charlottesville, VA, USA
kim.wasson@dependablecomputing.com

ABSTRACT

Software security metrics that facilitate decision making at the enterprise design and operations levels are a topic of active research and debate. These metrics are desirable to support deployment decisions, upgrade decisions, and so on; however, no single metric or set of metrics is known to provide universally effective and appropriate measurements. Instead, engineers must choose, for each software system, what to measure, how and how much to measure, and must be able to justify the rationale for how these measurements are mapped to stakeholder security goals. An assurance argument for security (i.e., a security argument) provides comprehensive documentation of all evidence and rationales for justifying belief in a security claim about a software system. In this work, we motivate the need for security arguments to facilitate meaningful and comprehensive security metrics, and present a novel framework for assessing security arguments to generate and interpret security metrics.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics; K.6.4 [**Management of Computing and Information Systems**]: System Management—*Quality assurance*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Measurement, Security

Keywords

Assurance Case, Confidence, Security Metrics

1. INTRODUCTION

The goal of a security metric is to enable stakeholders of software systems to answer questions such as: How secure is my system? Is the software adequately secure for its use? How has a series of modifications affected my system's security? And so on.

WETSoM'14, June 3, 2014, Hyderabad, India
Copyright 2014 ACM 978-1-4503-2854-8/14/06...$15.00
http://dx.doi.org/10.1145/2593868.2593880

Security metrics can facilitate enterprise-level design and operational decision making about software, but are difficult to capture [1, 13, 9, 8]. Without effective security metrics, software deployment decisions, upgrade decisions and so on cannot be addressed in an informed way for security-critical systems. In this paper, we present a novel security-metric framework based on evaluation of a rigorous *assurance argument* for security (i.e., a security argument). We refer to the metric as the Measurement Based on Security Argument (MBSA) framework.

Measuring software security (i.e., the degree to which software copes with relevant security issues) presents a unique challenge, because malicious action is difficult to model. Motivated adversaries actively seek to cause a failure, and typically attackers only need to find one vulnerability to be successful. Thus, an effective security metric must synthesize systemic data; a difficult task since software security is a complex and multi-dimensional property that includes interdependent sub-properties such as correctness, efficacy, and efficiency [9, 6, 11]. Further, security goals tend to differ from one organization to another and even among software systems within the same organization depending on the software's functionality and use [9].

Many methods for measuring software security have been proposed (for example, Herrmann [5] has consolidated over 900); however, stakeholders face a formidable challenge in applying useful/appropriate metrics that capture their requirements. In dealing with this challenge, stakeholders must answer four questions: (1) which facets of the software system require measurement, (2) what kinds of measurements are appropriate, (3) how extensive should chosen metrics be applied, and (4) how should individual measurements be combined to provide a meaningful overall measure of security?

Of these four questions, the fourth is the most difficult to answer and most crucial, because there has to be a link between chosen metrics and stakeholder security requirements. In other words, for a metric to provide useful information to stakeholders, measurements must be linked explicitly to stakeholder security goals [8, 9]. Few existing security metrics are based on an explicit theory for this linkage.

We introduce a novel security metric framework in which security is measured in terms of degree of *belief* (i.e., confidence) in a security claim. The approach is to define a claim about the practical security of a system based on stakeholder requirements, such as freedom from one or more defined classes of vulnerabilities, and estimate the degree to which the claim can be believed. Typically, security-critical sys-

tems will be built with appropriate technologies, and various analyses and audits will have been undertaken on the system. Belief in the security claim depends upon the degree to which the development technologies, the analyses and the audits support that belief.

A *security case* is a combination of a security claim, a body of evidence, and a rigorous *security argument* that links the evidence to the security claim. The argument explicitly documents the rationale for belief in the security claim based on the body of evidence. The argument permits scrutiny of the rationale by developers, stakeholders, certifiers, etc.

The key contribution of this work is that, to our knowledge, MBSA is the first framework for measuring security through measuring confidence (i.e., belief) in a security claim, based on analysis of the security argument for that system.

In the remainder of this paper, we further motivate the insight of the MBSA approach to security metrics (Section 2), the need for and mechanics of security arguments (Section 3), and discuss problems with confidence in security arguments (Section 4). The mechanics of MSBA are described in Section 5, and an illustrative example of MBSA is presented in Section 6.

2. METRIC APPROACH

2.1 Principle

The underlying principle of the MBSA approach to measuring security is to state a security claim about a system, develop an argument about why that claim should be believed, and then construct a metric based on the argument designed to assess *confidence* in the argument. More specifically, the metric is based on the following general principles:

Claim The security *requirement* of the system's stakeholders is stated precisely as a security claim about the system. This claim is stated in terms of a particular context that defines the system itself, the system's operating environment, and the threat model to which the system is subjected. The claim is established carefully to document the stakeholders' fundamental security requirement. By beginning with a security requirement, MBSA does not need to measure security adequacy: adequacy is the basis of the security claim.

Evidence The system is engineered to meet the stated requirement. The technology employed in the engineering and assessment of the system generates a body of *evidence* about the systems security properties. This evidence includes information about system tests, system analyses, vulnerability avoidance techniques employed, and so on.

Argument In principle, the engineering of the system should ensure that the system meets the stated security claim. Doubts arise, however, in terms of the adequacy and completeness of the engineering and assessment, and so belief in the claim might not be warranted. Thus, in practice the critical issue facing the stakeholders is whether they can believe that the security requirement is met. Construction of a rigorous *argument* documents the rationale for belief in the claim based on the available evidence.

Metric In practice, residual doubts in an argument are inevitable. For example, the argument might not have taken into account all possible circumstances or some elements of the argument might be based on invalid or corrupted data. MBSA computes a metric based on the argument that reflects confidence in belief in the security claim, i.e., confidence in the argument. Therefore, The metric is an assessment of security: if the claim is true, the system is adequately secure. Any doubt about the claim is a reflection of possible insecurity, and stakeholders can make decisions about system deployment and operation based upon the degree of disbelief that they are prepared to accept. Stakeholders can also direct further analysis of the system based on exposed areas of doubt.

2.2 Metric Form

The metric computed to assess confidence in the security claim has to meet two goals:

- The metric must take account of all of the factors that could influence belief in the security claim, i.e., the metric must be *complete*. Since the metric is derived from an argument, the metric would be incomplete if either the argument itself was incomplete or the argument, though complete, was not fully considered.
- The metric must present all of the factors that could influence belief in the security claim in a way that enables a properly informed judgment of belief in the security claim, i.e., the metric must be *valid*. This goal requires that the metric present all aspects of possible doubt in the claim in a form that can be examined, judged, and acted upon by the stakeholders.

The MBSA framework addresses these goal using a two-step process:

- The system security argument is annotated with confidence information about the various elements of the argument.
- A function is computed with the annotations as input. The function produces a presentation of information that constitutes the metric.

3. SECURITY ARGUMENTS

When engineering a system to meet the demands of the system's stakeholders, developers are left with the burden of demonstrating that the software meets these demands adequately. While there is a desire to provide definitive, complete, and irrefutable proof that a system will always perform as intended, such proof is unattainable in all but the most trivial cases. In practice, developers rely on the available evidence to demonstrate that a desired property is "highly probable", although quantification of "highly probable" is virtually impossible and so the interpretation is left to intuition.

Assessment difficulty such as this arises in other contexts, in particular in *system safety*. In system safety, an approach to assessment that is gaining traction is to construct a rigorous *safety case* where a safety case has the following definition [12]:

> A Safety Case consists of a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment.

The concept underlying a safety case is to create a com-

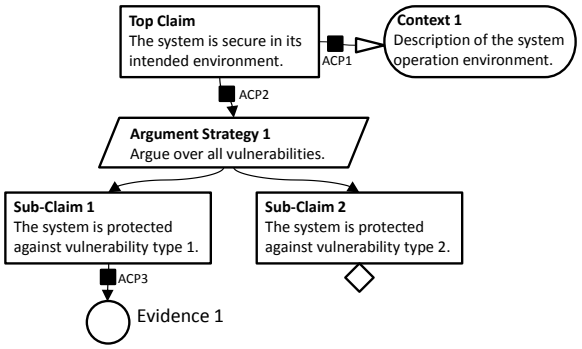

Figure 1: Example argument fragment in Goal Structuring Notation (GSN).

pelling argument justifying a claim that a system is adequately safe for its intended use. Safety cases are required by law for certain military systems, including those used by the U.K. Ministry of Defence [12]. Adapting this concept to security has led to the notion of a *security case*, a concept that is essentially identical to a safety case but with the concern being security rather than safety.

The argument included in a security case (and also in a safety case) is organized as a *goal structure* and documented explicitly, often using a custom notation, the Goal Structuring Notation (GSN) [7]. Figure 1 illustrates a simple example argument expressed in GSN. The major elements of the notation are: (a) goals (claims) represented as rectangles; (b) items of context represented as rounded rectangles; (c) argument strategies represented as parallelograms; and (d) items of evidence represented as circles. A diamond annotation indicates that the associated element remains to be elaborated, i.e., that part of the argument has not been developed. The black squares are *assurance claim points* (see Section 4).

A goal structure is hierarchical. Starting with the top-level goal, each goal is decomposed into sub-goals that are then further decomposed, and so on until leaf sub-goals are reached. A leaf goal can be justified based on supplied evidence. Decomposition of a goal is based on a strategy whereby the truth of the goal can be inferred from the sub-goals. The hierarchic decomposition of the top-level goal down to the leaf goals helps to make the argument convincing by introducing inferences from each goal into finer-grained sub-goals. Goal structured arguments are not formal, but the argument can be described as *rigorous*.

Regardless of how carefully constructed and well organized a rigorous argument is, there is rarely certainty in any goal. Such arguments are always *defeasible*, i.e., the argument might be invalidated and require revision as new information becomes available.

4. ARGUMENT CONFIDENCE

4.1 Defining Confidence

In general, belief in the top-level goal (claim) of an argument, whether for safety, security or another property is important. The premise upon which the MBSA framework is based is: (a) the top-level goal in the security argument defines the stakeholders' security requirement adequately, and (b) belief in the truth of the top-level goal is justified by the argument. Thus, security depends on the adequacy of the argument, and so meaningful security metrics are about confidence in the argument. The MBSA framework is designed to measure argument confidence in a practical way, and present the results of the measurement to stakeholders in a manner that allows them to make deployment decisions about security-critical software systems.

Although belief in the top-level goal rests on confidence in the associated argument, confidence in an argument is an elusive concept. Dictionaries do not give a definition that can be used in an engineering context, deferring instead to the use of synonyms such as "trust" or "faith".

The U.K. Ministry of Defence definition of a safety case given in Section 3 requires a *compelling* argument, yet no definition of "compelling" is given. Graydon et al. have proposed a suitable practical definition [3]. Their definition is termed *operational* since they equate a "compelling" argument to an argument having a set of established and measurable properties.

Hawkins et al. have introduced the notion of *confidence arguments* to supplement safety and security arguments in order to capture confidence assessment [4]. A confidence argument supplements a traditional argument and supplies the rationale for belief in the quality of each of the argument's items of evidence, context definitions, and inferences. When these elements are added to an argument, there is an assertion that the element is valid and correct, i.e., the element serves the intended purpose to support a claim. Assertions in the argument are linked via an *Assurance Claim Point* (ACP) to the relevant confidence argument where confidence in the assertion is argued.

The GSN extension to document ACPs is a black square, illustrated in Figure 1. For each ACP there is a corresponding confidence argument. The three points shown are associated with a context item (ACP1), an inference (ACP2), and an evidence item (ACP3).

In order to avoid the difficulties that arise with terms such as "compelling", we have adopted the concept of an operational definition from Graydon et al. [3], i.e., a definition that defines confidence in a practical way based upon measurable or estimable argument characteristics. The operational definition begins with the adoption of the confidence arguments as defined by Hawkins et al. [4] and then elaborates this concept to provide an operational framework suitable for use in security metrics.

4.2 Sources of Doubt

Any goal within an argument might or might not be true, and belief in a goal is usually a matter of judgment. A goal thought to be true might not be true because:

- The inference upon which the goal depends is invalid.
- The evidence upon which the goal depends is invalid or inaccurate.
- The goal might be invalid for the defined context or the context definition might be inaccurate for the system of interest

When an argument inference is used to connect one claim to another claim or an item of evidence or a context is added to an argument to support or contextualize a claim, there is an assertion about the validity and accuracy of that element of argument. These assertions are in essence the sources of doubt within an argument.

As an example of how doubt can arise in an argument,

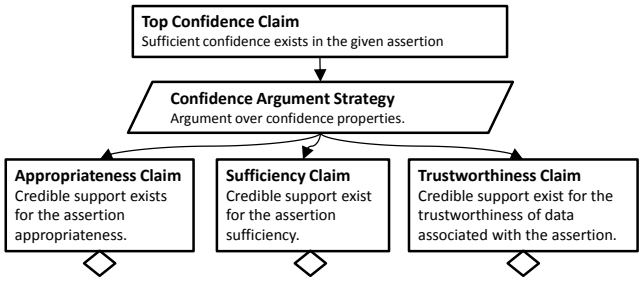

Figure 2: Confidence Argument Structure

consider a goal within a security argument which states that a given application is adequately protects against buffer overflow attacks. One form of evidence upon which the truth of this goal might rest could be the results of testing the application whereby the program is probed for buffer overflows. To support the truth of the goal with the results of testing is an assertion that: (1) the test results were correctly documented, (2) the test suite had sufficient coverage, (3) the tests were performed correctly, (4) the tests actually demonstrate an ability to thwart buffer overflow attacks, (5) the test conditions are within the intended operating conditions, (6) the test conditions represent realistic operating conditions, and so on.

4.3 Definition of Confidence

In order to develop the operational definition of confidence that we require, we define confidence with respect to three properties:

Appropriateness Does the argument element perform the intended purpose (i.e., goal support or contextualization) and is it realistic?

Sufficiency Does the argument element achieve its purpose sufficiently, i.e., have all considerations that might affect the efficacy of the argument element negatively been considered and have all relevant concerns been handled?

Trustworthiness If the argument element relies on data of any kind, can the integrity or accuracy of the data be trusted, i.e., believed?

In our approach, we adopt the use of confidence arguments, based on the concept introduced by Hawkins et al. [4]. A confidence argument is constructed for each ACP in the security argument, i.e., for each assertion associated with contexts, items of evidence, and inferences. Each confidence argument demonstrates sufficient confidence in a given assertion by arguing the existence of our defined properties of confidence, illustrated in Figure 2. The strategy for how to argue each property is a topic of current research, and not defined here.

Confidence arguments are used by MBSA to generate confidence metrics for any claim or set of claims in the argument, discussed in the following section.

5. MBSA

The Measurement Based on Security Argument (MBSA) framework is in two parts that operate sequentially:

Confidence Assessment Given a security argument where all assertions have corresponding confidence arguments, each confidence argument is annotated with a vector of measurements. Each measurement vector, referred to as a *Confidence of Assertion Vector* (CAV), contains a set of measures, the collection of which is meant to capture the quality/strength of the corresponding confidence argument. Each measurement in a CAV corresponds to a particular property of confidence and is derived from expert judgment in almost all circumstances.

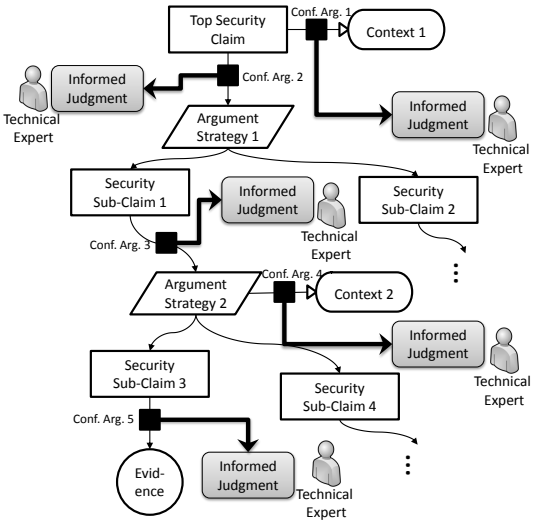

Figure 3: Confidence Assessment

Interpretation Confidence metrics are presented to decision makers by interpreting the results of confidence assessment (i.e., CAVs). Given a security argument annotated with CAV data, interpretation contextualizes, simplifies, and presents CAV data at the request of decision makers to best facilitate decision making. Interpretation of CAVs is performed by a sequence of *interpretation functions*.

The remainder of this section describes the underlying form and mechanics of confidence assessment and interpretation.

5.1 Confidence Assessment

During confidence assessment, each confidence argument is evaluated by one or more technical experts, illustrated in Figure 3. The figure depicts a security argument where each confidence argument (represented by black squares) is sent to technical experts to evaluate based on their informed judgment. Confidence arguments are used to support belief in an assertion. Consequently, experts are chosen for their expertise in the type of assertion associated with the confidence argument as well as their expertise in the component of the software affected by the assertion.

Experts derive a vector of measurements to quantify the quality/strength of the confidence argument, referred to as a Confidence of Assertion Vector (CAV). CAVs are the basic unit within the MBSA framework from which all metrics presented to decision makers are derived. At the completion of confidence assessment, each confidence argument is annotated with a CAV.

The general form of a CAV is represented as:

$$\vec{C}^{a\_type} =< \mu_1^{m\_type}, \mu_2^{m\_type}, ..., \mu_n^{m\_type} >$$

Each CAV, $\vec{C}$, has an associated assertion type, *a_type*, the primary assertion types being: inference, context or evidence (see Section 4). Similarly, measurements within the vector, μ, also have a corresponding type, *m_type*, indicating what

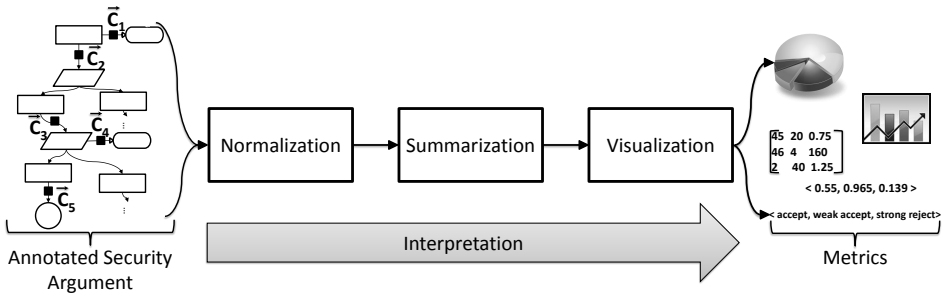

Figure 4: Interpretation of Confidence Assessment

property of confidence the measurement is meant to capture, e.g., sufficiency. All CAVs of the same *a_type* contain the same number of type of measurements. Each individual μ can be a single value or vector themselves, potentially containing further sub-vectors.

The general CAV form is intentionally left abstract and flexible to facilitate any custom implementation; however, we propose an implementation based on a direct mapping to the confidence argument. Our proposed confidence argument structure is to argue over three primary confidence properties, i.e., appropriateness, sufficiency, and trustworthiness (see Figure 2). A corresponding CAV structure for $\vec{C}^{context}$, $\vec{C}^{inference}$ and $\vec{C}^{evidence}$ would map measurements to confidence properties as follows:

$$< \mu^{appropriate}, \mu^{sufficient}, \mu^{trustworthy} >$$

While this implementation of the CAV concept provides a structure for what measurements should represent, the precise form of values each measurement should take is not specified. Fundamentally the value produced for each measurement should be based on further examination of the strategy of argument used to demonstrate the confidence property. For example, an approach to argue the sufficiency of an assertion is to argue that all assurance deficits have been considered, and that all relevant deficits have been adequately handled. This kind of argument lends itself to measures of confidence in terms of Baconian probability [14].

Generally, measurements can take any form, which can be, in addition to Baconian probability, Bayesian probability, expert judgment, checklists, or even some combination of these measurements. Since the precise strategy of argumentation in confidence arguments is a topic of current research and debate (see Section 4) the exact form is not well established. Consequently, the precise form of each measurement type should take is also not currently established.

Finally, any measure of confidence will have residual doubts about the measure itself; however, measuring confidence in this manner can be used to reduce residual doubts to be As Low As Reasonably Practicable (ALARP) [10].

5.2 Interpretation

When confidence assessment is completed, the security argument is annotated with CAVs for each confidence argument. Interpretation functions are then applied to CAVs to allow decision makers to derive various "views" (i.e., metrics) of the state of confidence in the security argument. Interpretation functions come in three forms that are applied in the following sequence (illustrated in Figure 4):

Normalization Normalization functions standardize or contextualize each measurement stored within a given CAV. For example, security arguments frequently require amendments as concepts of security or the operating environment evolve over time. In response to these inevitable context changes, a normalization function can be developed to apply weights to (i.e., scale) affected CAVs to keep the results relevant within the new context. Other examples of normalization include standardizing the scale of measurements (e.g., between 0 and 1), filtering pertinent data, and contextualizing measurements with respect pass/fail thresholds.

Summarization Summarization functions are used to distill CAVs into simpler and more easily comprehended forms. For example, to determine how much confidence there is in a given claim would require examination of all CAVs stemming from that claim and all its sub-claims. Instead of evaluating all CAVs individually, summarization can be applied to provide a simpler measurement of confidence for the given claim, such as an average, dot product, magnitude, min, max, standard deviation, etc. Summarization functions can operate on individual CAVs, a collection of CAVs, or a collection of previously summarized data. The data resulting from summarization can take almost any form, including individual scalar values, a single CAV, or a collection/matrix of CAVs.

Visualization Once CAVs have been normalized and summarized, one or more visualization functions can be applied to display the results to decision makers. Visualization can be used to produce graphs, charts, tables, annotated assurance cases, raw numbers, etc.

Generally, all CAVs are processed by these functions in the given order; however, decision makers might choose not to implement one or more of these interpretation functions. For example, decision makers could choose to view all raw CAV results without any interpretation, view raw summarizing numbers without visualization, view normalized CAVs without summarization, etc.

No single method for implementing and combining interpretation functions is defined. Decision makers are given the freedom to choose dynamically how to generate metrics. By altering interpretation functions, decision makers can alternate between very abstract and highly detailed measurements. Decision makers can also alter the parameters of interpretation functions to observe how hypothetical alteration to the software might affect overall security. As such, a practical implementation of MBSA would require users to define a database of interpretation functions.

6. ILLUSTRATIVE EXAMPLE

6.1 The Stoplight Metric

We illustrate the general mechanics and uses of the MBSA framework with an example security metric that we call the *stoplight metric*.

In this scenario, a decision maker must decide if a given software system is adequately secure to ship, and if not, needs to make recommendations as to what components of the software system need to be improved. The decision maker has at his or her disposal a security argument for the software system that is complete with confidence arguments. For this example, the exact nature of the given software system is not important and is therefore not specified.

The decision maker requires a means to evaluate the security argument systematically using a series of measurements of confidence. By selecting alternative views of the state of the system, the decision maker can affirm their decision and also isolate areas of highest concern. As such, the decision maker first uses a very terse and cursory metric to get an idea of the general confidence of the entire argument, referred to as the stoplight metric.

The premise of the stoplight metric is that each goal (claim) in the security argument can be described in terms of three levels of confidence, corresponding to three colors:

Green "Pass"; acceptable confidence
Yellow "Pass Pending"; provisionally acceptable confidence
Red "No Pass"; unacceptable confidence

While the stoplight metric can be used to provide a single color for any given claim, the decision maker desires an overview of the entire argument to understand the systemic security strengths and weaknesses.

6.2 Application of MBSA

To implement the stoplight metric using MBSA, first every confidence argument is assessed and assigned a CAV by teams of experts. For purposes of this illustrations, CAVs will have the form proposed in Section 4, i.e., provide measurements of trustworthiness, appropriateness, and sufficiency. As mentioned in Section 5, the form of these measurements is a topic of further study, so for this example, each measurement is in the form of a number between 1 and 10, inclusive.

With CAVs for every confidence argument, we define a stoplight algebra for assigning confidence status (Green, Yellow, Red) to a claim through the use of interpretation functions. The sequence of interpretation functions is defined as follows:

Normalization. For each CAV, a threshold is applied to each individual measurement within a CAV to translate numeric measurements into one of the three stoplight colors. The choice of thresholds is entirely up to the decision maker. For this example, measurements greater than or equal to 7 are colored Green, measurements above 4 but below 7 are colored Yellow, and measurements that are less than or equal to 4 are colored Red. For example a CAV with values $< 8, 2, 5 >$ would be normalized into a new CAV with values $< Green, Red, Yellow >$.

Summarization. Summarization is accomplished through first summarizing each CAV into a single color (referred to as a CAV summary color) and then percolating CAV summary colors up through the argument to each claim (i.e., from leaf goals up to the top-level claim). Percolation is another type of summarization function in which a claim is assigned color based on summarizing all CAV summary colors from that claim and all its sub-claims. Both summarizing functions operate using the same color algebra:

- When all values being summarized are Green, the result is also Green.
- If the values being summarized contain a mixture of Yellow and Green, but no Red, the summarized value is colored Yellow.
- If any values being summarized are Red, regardless of the color of the other values, the summarized value is colored Red.

For example, a CAV with normalized values $< Green, Red, Yellow >$ would be summarized as *Red*. The CAV summary color *Red* would then be used in the percolation summarization function with other summarized CAV colors.

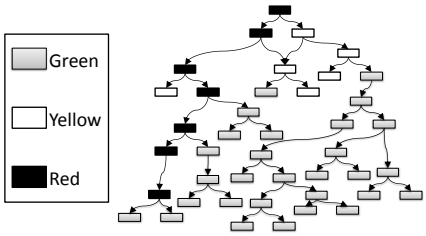

Figure 5: Stoplight Metric Example

Visualization. The visualization chosen for the stoplight metric is to present a miniaturized view of the security argument, where the argument has been stripped of all elements except for claims, and each claim has been colored. A claim is colored based on the summarized color value from that claim and all its sub-arguments. An example visualization is shown in Figure 5.

6.3 Uses of the Stoplight Metric

The stoplight metric provides a quick method to assess the entire state of security of a software system. For example, in Figure 5, the top-level claim is colored Red, hence, sufficient confidence in the primary security claim does not exist. The claims on the left side of the argument can be seen to be unacceptable and therefore requiring immediate attention.

The stoplight metric provides a initial view of the security of a software system, not a definitive measurement. For example, if the top-level claim were colored Green, a skeptical decision maker would be unwilling to accept that result without first posing some additional questions such as: What if a claim's color was changed from Green to Red, or vice versa? What if the color thresholds were altered? What if portions of the argument were considered higher priority than others? Etc. MBSA facilitates these answering questions by allowing decision makers to switch between different metrics, or change the parameters of a single metric by using different interpretation functions.

7. RELATED WORK

Many security metrics have been proposed. For example, Hermann [5] discusses over 900 different security metrics. The MBSA framework is based on argument confidence, a completely different concept from those used previously. Metrics have inherent assumptions and abstractions that

have led some to conclude the need for a *meta*-metric to measure these risks [13]. MBSA addressing this issue.

The key to providing meaningful metrics is the linkage of evidence to a security goal [8, 9]. MBSA utilizes security arguments to structure security goals for the purpose of a security metric. The key benefit of this approach is the flexibility that it provides stakeholders to choose what constitutes adequate security for their purposes.

Other approaches that link metrics to goals have been proposed in which metrics are defined that focus on a set of desired security-critical properties [8, 1]. These approaches have to consider multiple forms of measurement to justify that the software has a desired security property. Such metrics rely on implicit arguments about why a security property should be measured, and what measurements will show that the desired property is present. MBSA does not necessarily obviate the need for these metrics, but provides a framework in which justification for a metric is made explicit.

Denney et al. have proposed an approach to quantifying confidence in assurance arguments using Bayesian Belief Networks (BBN) [2]. In this approach, confidence is measured by identifying components of the argument and measuring the confidence in those components probabilistically. The probabilities are percolated through the argument using the mechanics of BBNs. A BBN confidence metric prescribes a single approach to modeling and measuring confidence. The approach relies on belief in the various probabilities used and might abstract too much information for decision makers. MBSA is a more general framework that does not prescribe methods for assessing and percolating confidence data. Hence, probablistic confidence measurements and the use of BBNs to interpret and percolate those metrics is one potential implementation of MBSA.

8. CONCLUSION

We have presented the Measurement Based on Security Argument (MBSA) framework, a new approach to measuring software security based on confidence in a security argument. We have also provided an illustration of the application of MBSA to construct a "stoplight" security (confidence) metric. MBSA provides a new direction for security metrics, which to our knowledge has not been explored previously. MBSA allows organizations to measure the adequacy of software to meet their particular security requirements. Security requirements are stated as claims within a security argument, and MBSA provides a framework for measuring confidence (i.e., belief) in those claims.

In general, security arguments document the rationale that engineers and auditors have for believing that security requirements are met, and such arguments have substantial value. Engineers always have a rationale for believing that their development techniques will provide adequate security, but their reasons are usually ad hoc and rarely documented. Explicit documentation of their rationale in a security argument allows scrutiny of that rationale by all of the system's stakeholders, including regulators. Rigorous arguments are in common use in some safety domains in the form of safety cases, and their use in security-critical system is increasing. Thus, while reliance on a security argument might seem to detract from the use of MBSA, the benefits of developing security arguments for security-critical systems suggest that their use will continue to increase and that the incremental cost of using MBSA will be small.

Finally, we note that MBSA is a general technique and not security-specific, and so it could be adapted to other properties of interest such as safety. Since safety arguments are used extensively in some industries, the MBSA could be applied there with virtually no added resources.

Acknowledgments

This research is supported by National Science Foundation (NSF) grant CNS-0811689, the Army Research Office (ARO) grant W911-10-0131, the Air Force Research Laboratory (AFRL) contract FA8650-10-C-7025, and DoD AFOSR MURI grant FA9550-07-1-0532. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, AFRL, ARO, DoD, or the U.S. Government.

9. REFERENCES

[1] C. Alberts, J. Allen, and R. Stoddard. Integrated measurement and analysis framework for software security. Technical report, DTIC Document, 2010.

[2] E. Denney, G. Pai, and I. Habli. Towards measurement of confidence in safety cases. In *The International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 380–383. IEEE, 2011.

[3] P. Graydon, J. Knight, and M. Green. Certification and safety cases. In *The 28th International System Safety Conference*, Sept. 2010.

[4] R. Hawkins, T. Kelly, J. Knight, and P. Graydon. A new approach to creating clear safety arguments. In *Advances in Systems Safety*, pages 3–23. Springer, 2011.

[5] D. S. Herrmann. *Complete guide to security and privacy metrics: measuring regulatory compliance, operational resilience, and ROI*. CRC Press, 2007.

[6] W. Jansen. *Directions in Security Metrics Research*. DIANE Publishing Company, 2010.

[7] T. Kelly and R. Weaver. The goal structuring notation–a safety argument notation. In *Proc. DSN 2004 Workshop on Assurance Cases*, 2004.

[8] S. Pfleeger. Useful cybersecurity metrics. *IT Professional*, 11(3):38–45, May 2009.

[9] S. Pfleeger and R. Cunningham. Why measuring security is hard. *Security Privacy, IEEE*, 8(4):46–54, 2010.

[10] F. Redmill. *ALARP Explored*. Technical report series. University of Newcastle Upon Tyne, Computing Science, 2010.

[11] B. Rodes and J. Knight. Reasoning about software security enhancements using security cases. In *The First International Workshop on Assurance for Argument and Agreement (AAA)*, Oct. 2013.

[12] M. O. D. Standard. Standard 00-56 issue 4-safety management requirements for defence systems. *UK Ministry of Defence*, 2007.

[13] S. Stolfo, S. Bellovin, and D. Evans. Measuring security. *Security Privacy, IEEE*, 9(3):60–65, 2011.

[14] C. Weinstock, J. Goodenough, and A. Klein. Measuring assurance case confidence using baconian probabilities. In *1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE)*, 2013.