A Decision-Theoretic Approach to Measuring Security (Author names omitted for initial submittal)

Abstract

The question "is this system secure?" is notoriously difficult to answer. The question implies that there is a system-wide property called "security," which we can measure with some meaningful threshold of sufficiency. In this concept paper, we discuss the difficulty of measuring security sufficiency, either directly or through proxy such as the number of known vulnerabilities. We propose that the question can be better addressed by measuring confidence and risk in the ***decisions*** *that depend on security. A novelty of this approach is that it integrates use of both subjective information (e.g. expert judgment) and empirical data. We investigate how this approach uses well-known methods from the discipline of decision-making under uncertainty to provide a more rigorous and useable measure of security sufficiency.*

1. Introduction

Fundamental questions such as "Is the system secure [enough]?" and "How much security is needed?" have proven notoriously difficult [12]. Answering such questions depend heavily on having a reliable and meaningful measure of *security sufficiency*. However, these questions regularly assume that "security" is a tangible system-wide property that can be measured directly or by proxy. This assumption has proven problematic on a number of fronts. Security is recognized as an emergent property of a system [8] [18]. That is, security arises in the complex interaction of many factors across the system as a whole, and can't be determined by measuring the security of individual components.

The difficulty of measuring security directly has led both researchers and practitioners to try to measure it by proxy with properties that hypothetically have a strong relationship to security. There have been two widely used proxies: the first is degree of compliance to security standards, e.g., implementation of the NIST controls such as those in NIST SP 800-53 [18]. It is now recognized that this is an unreliable proxy. "Security does not equal compliance" and the correlation between the two is hard to make rigorous 0. Another often considered proxy is based on the number and severity of known vulnerabilities (or, alternatively, weaknesses). Various scoring methods, such as the Common Vulnerability Scoring System (CVSS) [5] and Common Weakness Scoring System (CWSS) [6] have tried to account for the number and severity of vulnerabilities/weaknesses in the system to arrive at an overall number. However, again, there is no clear relationship between this proxy and a useful measure of security.

As with classic software quality attributes (e.g. reliability), security seems to suffer the problem that tangible and practical-to-measure indicators have undeterminable or overly weak relationship to an abstract system attribute. But this is not the entire problem. Even if we had good indicators for security, how would we know how to use them? For example, what would the threshold values be for "good enough" security?

In this work we investigate a fundamentally different approach. We view security as a decision problem, rather than one of measuring a property of the system. Security decisions are made relative to what is tangibly "at risk" and the cost of mitigating that risk within the specific system context. That is, there is no absolute measure of system security and of its sufficiency. What may be insecure in one context may be secure enough in another. Consider, for example, a CubeSat, a small, relatively inexpensive satellite [7]), verses a large, complex and expensive earth-science orbiter such as the Soil Moisture Active Passive (SMAP) satellite. One may be reasonably confident that an open source real-time OS, say FreeRTOS, is secure enough for use in a CubeSat, yet not be confident enough to use it for SMAP. The consequences of an OS exploit on SMAP would be much greater.

From the decision perspective, security measures are used to decrease uncertainty in the factors used to determine the "least risky" decision choice for a given situation. Accuracy in measurement is not as important as how much uncertainty the measure can reduce, that is, how much justified confidence it gives us. Measuring confidence and risk from the decision perspective takes uncertainty into account, and this gives us better insight into the question of sufficiency. If the decision is "too uncertain to determine," the uncertainty itself can be considered an insufficiency, and the best course of action may be to invest in more measurement to "buy down" the uncertainty. One need

URI: http://hdl.handle.net/10125/41901
ISBN: 978-0-9981331-0-2

not have perfect confidence in a decision. Rather enough so that the risk of not making the best decision is low. This decision process will be discussed in detail in a subsequent section.

This paper is organized as follows. In Section 1 we provide some context, background, and prior work related to a decision-driven approach to measuring security sufficiency. In Section 2 we review some definitions of security and make adjustments to orient them to decision making. In Section 3 we investigate a decision-driven approach to addressing security sufficiency. In conclusion, we describe the research contributions from this study. Both theoretical and practical implications are discussed. We also discuss the limitations of this study and plan for future work, building on the results presented here.

1. Background and context

The impetus for this research stems from our work on improving the cyber security of earth and space mission systems, where we needed to: assess the current security level of the systems, select activities to improve security in an economic way (i.e., with an understanding of cost vs. risk), and evaluating whether the results were sufficient. As such, being able to measure security sufficiency is central to our work.

Security sufficiency decisions are invariably fraught with great uncertainties and complexity, with risk falling on multiple stakeholders. Because of this, important decisions must be made with *justified confidence*. While when this is not fundamentally different than many other kinds of systems sufficiency decisions (e.g. sufficiency of reliability, safety, or quality), the risk profile resulting from security-related uncertainty is different and must be taken into account. For example, with security, as with safety, it is impossible to be completely confident in the factors that determine sufficiency. That is, we may not know all potential causes of failure – hazards (in the case of safety) or vulnerabilities (in the case of security) – along with their impacts. For both safety and security, the likelihood of a defect becoming a failure and impact of that failure tends to follow a distribution with large variability and generally be positively skewed [19]. But unlike a failure from a safety hazard, when a security vulnerability is exploited, one must assume the impact will reach its maximum potential due to the presence of a persistent intelligent agent. One must also assume that the longer a vulnerability remains in place, the greater the risk it will be found, and when found, exploited. Hence vulnerabilities have a very different risk profile from safety hazards, and this affects sufficiency decisions.

When making a security sufficiency decision, the confidence we have in the *decision* factors is what determines sufficiency. Note that high confidence here means we can determine, relative to a given level of risk tolerance, sufficiency versus insufficiency. Low confidence results in indetermination, which implies a need for further investment in reducing uncertainty in the decision factors. Low decision confidence means that if the “best” decision option is chosen, based on decision factors that are currently known, there would be a high risk that the decision would result in unacceptable losses.

Unfortunately, current security metrics are inadequate for addressing these kinds of sufficiency decisions. Most software security measures are either excessively pragmatic (e.g., they count detected vulnerabilities) or excessively rigorous (e.g., they exhaustively assess degree of compliance to the NIST security standards). But while these may provide insights into the effectiveness of whole processes, they are not precise enough to measure the effectiveness of individual practices and techniques, under specific circumstances and in particular environments [16].

Currently, most software security metrics focus on counting and comparing vulnerabilities, measuring attack surface, measuring complexity, or assessing compliance to a standard, and use some form of checklist or scoring system. But measures such as the average number of vulnerabilities per X lines of code or the Microsoft metric, comparing the numbers of vulnerabilities in earlier versus later versions of software programs, have not proven useful for indicating exactly how to reduce the risk of making a “bad” decision due to the uncertainty in security factors. For example, such metrics wouldn’t help predict whether software that appears to be secure in a development environment would be not be exploited when deployed in the field [16]. When these unreliable metrics are combined with limited budgets and schedules, and the fundamental inability to provide absolute security, they tend to render us unjustifiably over-confident in making decisions on security sufficiency. Furthermore, when security risk is too high, it is difficult to ascertain the most cost-effective security areas to address and the effectiveness in options for managing this risk. Cost-effectiveness is also a security sufficiency question [12].

The tangible consequence of uncertainty in security sufficiency is security decision risk. As we show in Section 3, we can measure this decision risk to address security sufficiency questions.

2. Related work

Measuring sufficiency of security has been investigated in both the information security and the software engineering communities. The metrics proposed are often based on or analogous to existing measurements of quality, reliability, or safety. The Department of Homeland Security's (DHS) Working Group on security metrics and measurement seems to be focused mainly on metrics adapted from the information security community [9]. Others, such as SAMATE [13], mainly focus on tools to help measure security. The most comprehensive and closely aligned work can be found within the Software Engineering Institute's (SEI's) Cybersecurity and Software Assurance Measurement and Analysis (CERT) initiative [4]. This initiative is based on several fundamental security measurement efforts [17][12][16], and it includes the Software Security Measurement and Analysis (SSMA) project, whose purpose is to aid decision makers with risk-based assessment and evaluation methods.

The works referenced above, and the security literature in general, proposes measures of security as a system or process property ("capturing a concrete attribute") [17][16]. At the same time, these works often view security as a form of assurance (e.g. [11][17][12]). The concept discussed in this paper differs in that we view security as a level of confidence relative to making a decision rather than as an inherent property of the system itself. This is consistent with our view of assurance, in general, which we also see inextricably tied to decision making [15]. Additionally, most security measures tend to "presume innocence" and assume perfect security until the elements being measured show otherwise. Our approach is necessarily conservative. We assume maximum uncertainty for any decision factor until we have measurements that justify how much uncertainty can be reduced for that factor. This is our source of "justified confidence."

2. Defining Security Sufficiency

As discussed above, security sufficiency measurement is tricky. Perhaps a contributing factor is that it is difficult to construct a meaningful and useful definition of security sufficiency. In our review of security terms contained in the literature, we noted that the concept of sufficiency is frequently used without being explicitly specified. For example, in the NIST Glossary of Key Information Security Terms [10], one definition of security assurance refers to "adequately met" and "sufficient" numerous times:

[Security] Assurance - Grounds for confidence that the other four security goals (integrity, availability, confidentiality, and accountability) have been adequately met by a specific implementation. "Adequately met" includes (1) functionality that performs correctly, (2) sufficient protection against unintentional errors (by users or software), and (3) sufficient resistance to intentional penetration or by-pass.

It is difficult to measure something that is poorly defined. We propose a definition of security sufficiency that represents our interest from the decision making perspective. While we accept that there is no inherently correct definition, we find validation through its utility in addressing security sufficiency problems.

Complicating our effort to define "security sufficiency" is that there are a variety of views and definitions of security and security risk to which our definition tries to maintain consistency. For our purposes, we will consider the NIST definition of security [10]:

Security - A condition that results from the establishment and maintenance of protective measures that enable an enterprise to perform its mission or critical functions despite risks posed by threats to its use of information systems. Protective measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correction that should form part of the enterprise's risk management approach.

It is important to note that since we can never know all the threats or all the potential vulnerabilities, we can never obtain a perfectly objective measure of security. We address this by using an operational definition of security that measures confidence relative to a threshold:

Security sufficiency – the degree of confidence that the security controls in place for this system will keep losses from system vulnerabilities under an acceptable level.

This new definition enables us to focus on a measurable outcome of security decisions, namely "losses from system vulnerabilities." Moreover, we can compare the expected losses from various decision choices and derive the *expected opportunity loss* – the additional cost of not making the "best" choice, the one that minimizes the expected loss.

In this light, the "best" security choice is the decision option that has the maximum expected outcome or minimum expected opportunity loss. A "bad" security decision is one in which a different decision would have been made had there been no uncertainty in the assessment of security.

If we have perfect information, we could make the best decision every time. Trouble arises when there is a large amount of uncertainty in the factors that determine security. Like all humans, decision makers have inherent biases, which can lead them to make biased and over-confident decisions in the face of uncertainty. This is the decision risk we wish to avoid, defined as follows:

Security decision risk is the potential loss of making a bad decision due to the uncertainty in security factors.

In cyber security, uncertainty often comes in the form of unknown-unknowns (e.g., zero-day exploits), because in a well-tested and managed system, the known-knowns may have already been eliminated, and the known-unknowns have been addressed through analysis. However, we don't take this as a given., We approach system security decisions by assuming the maximum uncertainty and perform investigations and interventions to justifiably reduce this uncertainty.

We also realize that uncertainty cannot be completely eliminated. Good decisions can still have bad outcomes (through highly improbable "bad luck"). Bad decisions can have good outcomes (through "dumb luck"). But even though we cannot completely control the outcome, our aim is to enable making optimal security decisions, taking into account tradeoffs between cost, effectiveness, and operability. What we are maximizing is confidence that our decisions have mitigated unacceptable losses.

The definitions given above enable us to make use of a variety of established risk metrics, such as value-at-risk (VaR), to measure security decision risk to determine security sufficiency. We provide a detailed example of this in Section **Error! Reference source not found.**.

2.2. The Role of Security Activities

One of the things we want to know is how security activities (our actions) can help us achieve security sufficiency, as defined above. There are two general classes of activity to consider: Security interventions and security investigations. Security interventions are activities that mitigate security risks by directly reducing the likelihood or impact of an unacceptable loss from an attack. For example, introducing 2-factor authentication for user login will reduce the likelihood of an intrusion from account hijacking or brute force password attacks. Security investigations are activities that increase our knowledge of security: these reduce our uncertainty about the likelihood or impact of an unacceptable loss from an attack. For example, auditing compliance to the ISO 27001 standard or performing a credentialed vulnerability scan [20] will decrease the uncertainty that a vulnerability exists or has been overlooked.

3. Using Decision Risk to measure Security Sufficiency

In this section we will use the concepts developed in Section 2 to build an example measurement of security sufficiency for making a security sufficiency decision. Along the way we will explore consistency with the concepts discussed above and validation though sensitivity analysis. While this example is representative, it is not based on an actual situation. Some considerations are necessarily simplified and the data and estimates presented are only illustrative.

First, let's consider some examples of the types of security sufficiency encountered during a system's development and operational life cycle:

1. Approval to Operate (ATO) – Deciding whether the system is secure enough to put into operations
2. Prioritizing Security Mitigations/Controls (which mitigations or security controls to implement) given a limited budget
3. Fix/no-fix decisions for individual vulnerabilities or weaknesses described by scans
4. Incident response decisions, such as when to take a system off-line in response to an attack

For our example, we will examine a system being considered for release to operations and the approval to operate (ATO) security decision. We formulate the options in terms of security sufficiency as follows:

- "Send" - The system is secure enough for operation. Implement security as designed, release system and monitor security.
- "Hold" - The system is not secure enough for operation or there is too much uncertainty that it is secure enough. Perform security investigations to reduce uncertainty or interventions to increase security.

Our aim is to determine an optimal decision with acceptable decision risk. To evaluate this decision, we will need to use some indicator of security sufficiency relative to the security controls implemented. While there are a variety of indicators we might use, for our purposes we will consider a simple Pass/Fail indicator where "Pass" means there will be no unacceptable losses from a security breach, and "Fail" is otherwise.

The Pass/Fail state of the indicator is uncertain at the time the decision is made. Indeed, it cannot be determined unless an unacceptable loss actually occurs. Therefore, we view it as a random variable whose expected value is given by p, the probability of Pass. This will be a significant factor in determining the optimal decision. However, at best we can only estimate p, perhaps with some sample data, and so we must account for this uncertainty when determining the optimal decision. We start by assuming the estimate of p is completely uncertain (i.e. no prior information) which we represent as a uniform distribution on [0,1] i.e. maximum entropy.

Our technique is to use Bayesian analysis to update our estimate of the distribution, based on information from estimates of security factors for the decision. The factors may be quantitative or qualitative values (e.g., it could be based on expert judgment), and these factors are also represented as random variables. The information needed is how the factors relate to the decision, which then can determine how the prior distribution is affected.

The information is incorporated through Bayesian updating, which reduces uncertainty in the posterior distribution. As discussed in the previous section, the information that reduces uncertainty can come from either security interventions (directly improving security) or security investigations (just learning more about the state of security). In practice, we use confidence intervals or Bayesian creditability intervals [2] to represent ranges for our estimates of the security factors.

What is the optimal decision, Send or Hold, given the a-priori security sufficiency distribution? We will define the optimal decision as the option (Send or Hold) that results in the minimum median expected opportunity loss (min EOL) for the decision. This minimizes our decision risk.

To compute this, we consider the potential losses given the various decisions and outcomes. For convenience we will consider losses in terms of dollars. We also assume the release of the system for operation is expected to generate significant value. That is, the desirable decision or "default decision" is to Send unless the security sufficiency is too low.

TABLE 1 is the decision payoff table. It summarizes the potential costs for each decision given an outcome. If, for example, the system is put into operation and no unacceptable loss occurs from vulnerabilities ("Pass") the table shows loss is $0 because all has gone as planned, nothing is lost and there are no additional unplanned costs or losses. It is the "default" or expected decision-outcome, because the costs for this situation have already been allocated and there are no unexpected costs or losses.

TABLE 1. POTENTIAL LOSSES FROM RELEASE DECISION

Payoff	Send	Hold
Pass	0	B
Fail	A	C

However, the table shows a cost of $A if the system is put into operation and it unexpectedly incurs an unacceptable loss from a security breach. In this case, in addition to the security breach loss, there will be the additional cost of securing the system against future attacks, including lost time, while the system is non-operational. Furthermore, the table shows a cost of $B if the release is held up but subsequently no significant security breaches occur. The losses are due to schedule stretching and additional security effort (unplanned work). Lastly, a cost of $C is incurred if the release is held up for rework and it still fails through a security breach, though a less severe one than the one that caused the $A loss. The $C cost is due to both improving the system and the later security breach. To summarize the relative magnitude of the costs: $A < B < 0$ and $A < C < 0$.

We need to deal with the uncertainty in the values of A, B, and C. We do not have their exact values, but we are able to reasonably estimate 95% credibility or confidence intervals. Accurate and useful credibility interval estimates can be obtained empirically from a combination of sample historical data and expert judgment. One approach is to obtain a triangular distribution from worst case, most likely, and best case estimates and from prior analogous projects then compute the 2.5th and 97.5th percentiles to generate a 95% credibility interval.

Finally, we focus on the uncertainty in the value of *p*, the probability of Pass – experiencing no security breach in the system. Again, we will express this uncertainty as a 95% confidence interval estimate, obtained from sample historical data and expert judgment. We can now consider the Expected Opportunity Loss (EOL) for each decision as the expected cost of each outcome for each decision, given the parameters A, B, C, and p defined above:

$$EOL_{Send} = (1-p)*(C - A)$$
$$EOL_{Hold} = -p*B$$

The EOL is summarized for the release decision in TABLE 2.

TABLE 2. OPPORTUNITY LOSSES FROM DECISION

OL	Send	Hold
Pass	0	-B
Fail	C - A	0

In order to minimize decision risk, choose the decision option with the smallest EOL. This gives the rational decision criterion: The decision is to Hold when $EOL_{Send} > EOL_{Hold}$, which occurs when $p < (A-C)/(A-C+B)$, otherwise Send. This criterion is easily seen to be consistent with our release judgment because $0 < (A-C)/(A-C+B) < 1$. If we were certain the release would pass ($p=1$), the criterion indicates Send, which is what we would expect. If we were certain the release would fail ($p=0$), then the criterion indicates Hold, again as we would have decided without the criterion.

The important principle is that *decision risk* is the potential loss from making a wrong decision due to uncertainty in the decision factors *A*, *B*, *C*, *p*. If there is no uncertainty in the security factors, there will be no decision risk, since there is no uncertainty in knowing whether $EOL_{Send} > EOL_{Hold}$ for the decision criterion. However, if there is uncertainty in the security factors, then there can be uncertainty in whether $EOL_{Send} > EOL_{Hold}$, potentially leading to a wrong decision. The decision risk here is the potential loss resulting when a decision is based on one EOL being smaller than the other when actually it is not.

This "higher order effect" is not particularly straightforward to visualize, but a practical approximation is to represent the EOL uncertainties by computing their 95% confidence intervals from the OL model and the confidence intervals for the security factors as described earlier. The decision risk is indicated approximately by the amount that these confidence intervals "overlap" each other. The more they overlap, the more likely we may make a wrong decision; the overlap represents an "area of confusion" regarding which EOL is greater than the other. To be more precise about this, it is worth defining the Expected Decision Loss (EDL) as the expected loss when we are wrong about the decision criterion relative to a given decision, in our example whether EOL_{Send} is indeed greater than EOL_{Hold}.

Our goal then is to minimize the EDL, and to do this in our example, we must have a rule for determining whether $EOL_{Send} > EOL_{Hold}$ when there is uncertainty. One possible rule is to say $EOL_{Send} > EOL_{Hold}$ when $\text{median}(EOL_{Send}) > \text{median}(EOL_{Hold})$ or "the middle of the road rule." We can then express the EDL for this rule:

$\text{median}(EOL_{Send}) < \text{median}(EOL_{Hold})$:

$$EDL = P(EOL_{Send} > EOL_{Hold})* EOL_{Send}$$

$\text{median}(EOL_{Send}) > \text{median}(EOL_{Hold})$:

$$EDL = P(EOL_{Send} < EOL_{Hold})* EOL_{Hold}$$

For the decision risk we need the probability distribution of the EDL. This is a bit difficult to compute directly, but tractable using Monte Carlo methods. In order to make practical use of the decision risk, we need to establish a threshold for how much decision risk we are willing to assume. Our threshold is expressed as the Value at Risk (VaR), the maximum loss we are willing to accept within a certain tolerance. It is common to select a 5% tolerance, which means "we want to be 95% confident that we will not lose more than $X (the VaR) from decision risk."

What remains, then, is to estimate the decision risk and VaR from the decision parameters, A, B, C, and *p*, discussed earlier. We created a Monte Carlo simulation for this calculation, whose outputs are shown in Appendix A. Let's walk through the results:

First, we estimate 95% intervals for the decision parameters *A*=[-$2.5M, -$1M], *B*=[-$300K, -$100K], C=[-$700K, -$200K], and *p* = [0.75, 1]. After generating the distributions for the EMV's, it isn't obvious which decision will have the best-expected outcome. From the EMV's, the EOL distributions can be generated. But again, it's not clear which decision has the lowest EOL. The box plots of the EOL's, shown on the right side of Appendix A, reveal that the median for EOL_Send is smaller than the median EOL_Hold, hence our reference decision is to Send.

However, the large inter-quartile ranges in the box plot indicate that there is considerable uncertainty in this decision. The 95% EOL intervals show a significant overlap so we expect a large decision risk. The 95% EDL shows a VaR of about $188K, which is indeed a large decision risk, and potentially we should be willing to pay up to $188K to reduce it.

Finally, if we apply our decision rule "use the lowest median EOL," simulation shows what the potential losses are from this decision *for a given release* (not on average as with EDL). The 95% VaR here is about $260K, which as expected is larger than the 95% EDL VaR, since the variability for any given release is greater than the variability of the average over many releases. Because this is the risk for a given release, it represents the value of removing all the uncertainty, commonly called the expected value of perfect information (EVPI) for the decision at hand.

Is the VaR for EDL consistent with our model for decision risk? Let's say that some investment in the release improves our confidence in passing to *p* = [0.95, 1], then the VaR reduces to about $37, or basically zero. This is consistent with decision risk we would expect given that we are confident that it is better to Send when we are pretty sure of passing. Similarly, if p = [0, .5] the VaR is about $150, again basically zero. This also makes sense for decision risk, since we are less certain the release will pass than fail, and the consequence of failure is very large – a situation in which we are more confident to Hold. Let's say we are totally uncertain about the likelihood of passing, which means the probability of passing could be anywhere from zero to one, that is: p = [0, 1]. Of course the safest decision is to Hold, but relative to the previous example we see that this decision has significantly more VaR at about $55K. Here again, this matches our expected decision risk, since if we have no information, then we certainly should expect that we could be wrong in deciding to Hold and will pay some potential loss as a consequence. It is interesting to also note that the EVPI here is substantially larger at $256K. This too makes sense because there is no "averaging" over multiple releases. A bad decision here cannot be balanced with good (or more accurately, "lucky") decisions later. Hence the VaR for EDL appears to be a metric consistent with decision risk.

The relationship between decision risk and the confidence we have in the security parameters is not as obvious. Figure 1 shows a sensitivity graph of VaR as our uncertainty about p = .5 decreases (i.e. the credible interval of p gets smaller) from total uncertainty to total confidence. As expected, the VaR decreases as we become more confident in the estimated range for p. What is notable here is that the decision risk decreases non-linearly. This indicates that if there is a lot of uncertainty about p then even a small amount of increased confidence can result in a significant decrease in decision risk. It is also notable that when p=[.2 , .8] (i.e. x=.4) is the "confident enough" point where there is very little to be gained from increased confidence. Indeed, looking the VaR for the EVPI at this point is essentially zero indicating that there is little risk from making the wrong decision (on this given release) due to uncertainty. What is surprising here is how large this range is indicating that we don't have to estimate p with great accuracy to determine the best decision to make.

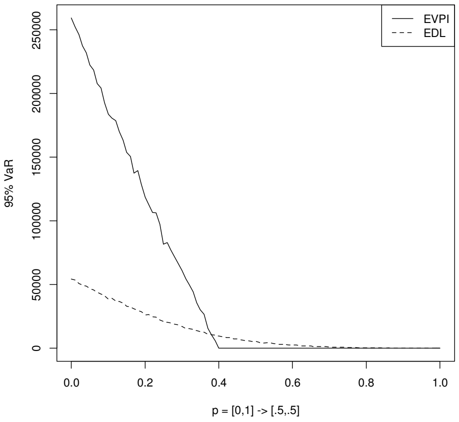


Figure 1: VaR as x →1, p=[.5x, .5(2-x)]

Figure 2 shows a sensitivity graph of VaR as p goes from zero confidence p=[0,1] to certainty p=[1,1]. It may seem surprising that smaller credibility intervals up to about [.7,1] have increasing VaR. While the uncertainty in the interval for p decreases, the EOL's get closer and closer together where a small amount of certainty making it more likely to make a wrong decision. Averaged over many releases the wrong decision will be made and losses will occur. At some point we switch the default decision and the intervals get very small while simultaneously the EOL's get farther apart making it less likely to make a wrong decision. This is another strategically useful thing to know. It indicates that increasing security, which would decrease the interval for p, is not valuable unless $p > 0.7$. Here again, when it is valuable, a small amount of credibility can result in large reduction of decision risk. Of course for a given release, the VaR EVPI will not suffer this kind of sensitivity and the more confident we are the less VaR as indicated in the EVPI graph in Figure 2.

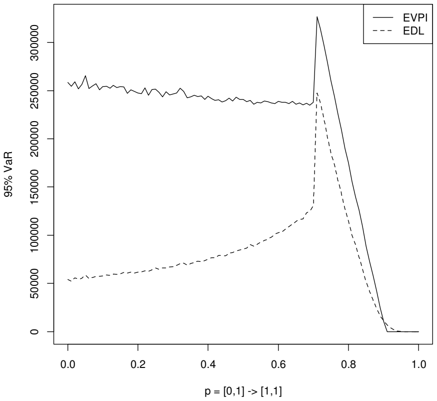


Figure 2: VaR as x→1, p=[x,1]

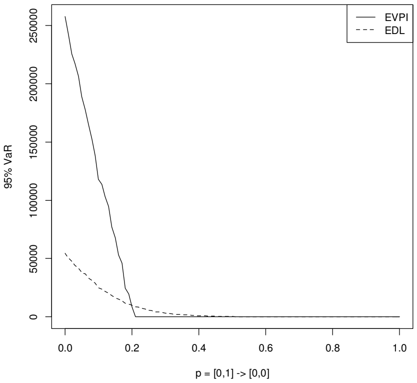


Figure 3: VaR as *x* →0, *p*=[0,*x*]

If we become more confident that the release will fail i.e. *p* → [0,0] then Figure 3 shows that at about p=[0, .2] we can be quite confident in our decisions to Hold.

As a final sensitivity check, we consider a sliding interval of fixed size (in this case, .02). This is to see which range of *p* is the decision most sensitive. Looking at Figure 4 we see that sensitivities for p between .75 and .95 indicating that the decision switch off point is somewhere around .85. That is, any range that includes this point will be risky, so if we want high confidence in our decision we should aim to obtain a range for p of about [0,.75] or about [.95, 1]. Of course, it may not always be possible to obtain one of these ranges.

We are now ready to address a quantitative indicator of security sufficiency. Recall that security increases the confidence in the estimates of the decision factors by providing evidence about the actual level of security. This has the effect of shrinking the credibility intervals of the decision factors *A*, *B*, *C*, *p*. That is, a decision factor is more credible with security than without. It is important to note that credibility increases (i.e. uncertainty decreases) regardless of the outcome of security activity. For example, vulnerability testing will decrease security uncertainty whether vulnerabilities are found or not found.

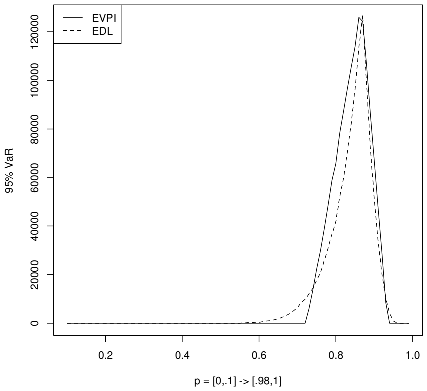


Figure 4: VaR as *x* →.99, *p*=[x-.01, *x*+.01]

Note that while all the decision factors are important, we generally focus on *p,* since the decision risk is more sensitive to this factor. Increased confidence in decision factors shrinks the sizes of the EOL intervals, narrowing the overlap, and thus reducing decision risk.

This is really just a form of information buying, where "perfect information" means no uncertainty (or zero length intervals), which is generally impossible or impractical to achieve in practice. However, in principle this represents the potential value of security, which we measure as the VaR discussed previously. Given some security, we can obtain VaR_{cert} or the revised VaR with the new credibility intervals from security. Hence we can define the Value of Partial Information, VPI = VaR - VaR_{cert}. as the value of partial information resulting from the security performed. It is the potential maximum monetary loss that is avoided by security and thus is a meaningful quantitative representation of its benefit. In practice we are usually more interested in $\text{VaR}/\text{VaR}_{\text{cert}}$ for use in cost-benefit and effectiveness of security analysis (it avoids some issues of accuracy in cost estimation).

But how can we measure how much a particular security activity reduces VaR? For this we look at Bayesian updating [3] represent the effect of buying security. As a simplified illustration of this, consider the question of performing a credentialed vulnerability scan (let's call this CVS). Suppose that historically and based on expert judgment for this particular release we estimate $p = [0.75, 1.0]$. On the one hand this states that $P[Pass] \geq 0.75$ and we are interested how this changes given a CVS was performed. That is, what does P[Pass ∣ CVS="Yes"] do to increases this bound. Bayesian updating suggests that

P[Pass ∣ CVS="Yes"] = P[CVS="Yes" ∣ Pass]*P[Pass]/(P[CVS="Yes" ∣ Pass]*P[Pass] + P[CVS= "Yes" ∣ Fail]*(1-P[Pass])).

Say we have reviewed 50 recent releases where CVS= "Yes" and found of these, 30 passed and 20 failed. So P[CVS= "Yes" ∣ Pass] = 30/50 and P[CVS= "Yes" ∣ Fail] = 20/50. Based on this sample data we estimate that P[CVS= "Yes" ∣ Pass] ≥ 0.82. The revised interval $p = [0.82, 1.0]$ gives a VaR_{cert} of about $87K and the VPI is about $100K. Similarly we can consider how P[Pass ∣CVS= "No"] would revise the upper interval limit $P[Pass] \leq 1$ which would also decrease the VaR.

In practice this example is a bit overly simplified. Generally, there is large variability in the sample data, which needs to be taken into account. It is also common that the answers to the security questions are not 100% confident. Typically, the answers are "Yes, but…" or "Mostly No" with details on why and how.

Another issue is that risk and confidence are frequently communicated qualitatively through "fever charts" of red ("high risk"), yellow ("moderate risk"), and green ("low risk"). Also a security activity (i.e. an intervention or investigation) tends to affect multiple decision factors simultaneously and there are not independent of each other. Another issue is that the likelihood of passing increases after Hold because further testing and repair will be performed to decrease this uncertainty. Finally, the default decision in practice is generally "Partial" and not "Send" as we assumed previously. To some degree these issues can be addressed in the Monte Carlo simulations. The main point of the current investigation is to see that a small amount of information obtained through security activities, such as answering the questions, can have a dramatic effect on the decision risk which results in a number of benefits.

3. Conclusion

3.1. Contributions

This study makes both theoretical and practical contributions. On the theoretical side, using decision risk provides an accessible means to analyze a Cybersecurity sufficiency decision. The insights gleaned also open potential avenues for further research.

The practical contribution this work is its potential to improve decision-making at NASA and other organizations that have to make Cybersecurity sufficiency decisions about complex systems.

3.2. Limitations and Future Work

The study presented above is conceptual in nature. It identified the problem of determining security sufficiency, of answering the question: "how much security is enough?" Reviewing the literature and current practice revealed flaws in the approaches taken to-date. The study then laid out the definitions and strategy required to measure security sufficiency. It's contribution was in changing the focus from abstract system properties to risk and uncertainty inherent in decision making.

The primary limitation of this study is that it has yet to be put into practice on real-world systems – and this is the goal for future work in this area. The strategy described above can be applied to any internet-connected system, which provides value through the network, yet faces security threats. The application would be especially useful when (as is commonplace), there is a limited budget for security controls and mitigations, and it is critical to answer know which ones and to what degree they should be applied.

6. References

[1] Alberts, Christopher., Allen, Julia., & Stoddard, Robert. 2012. *Risk-Based Measurement and Analysis: Application to Software Security* (Technical Report CMU/SEI-2012-TN-004). Pittsburgh: Software Engineering Institute, Carnegie Mellon University. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=10067

[2] Aynes, E. T. (1976). "Confidence Intervals vs Bayesian Intervals", in Foundations of Probability Theory, Statistical Inference, and Statistical Theories of Science, (W. L. Harper and C. A. Hooker, eds.), Dordrecht: D. Reidel, pp. 175 et seq

[3] Berger, J. (1985a), Statistical Decision Theory and Bayesian Analysis, 2nd ed. Springer, New York.

[4] CERT - Software Engineering Institutes Cybersecurity and Software Assurance Measurement and Analysis (http://www.cert.org/)

[5] Common Vulnerability Scoring System Support v2, https://nvd.nist.gov/cvss.cfm

[6] Common Weakness Scoring System (CWSS), https://cwe.mitre.org/cwss/cwss_v1.0.1.html

[7] CubeSat project, http://www.cubesat.org/

[8] Ellison, *The Influence of System Properties on Software Assurance and Project Management*, CERT whitepaper, https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/project/228-BSI.html

[9] Information Security And Privacy Board, NSA Center for Assured Software, March 21, 2006

[10] Kissel, Glossary of Key Information Security Terms, NISTIR 7298, nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

[11] Mitchell Komaroff (ASD/NII) and Kristin Baldwin (OSD/AT&L), DoD Software Assurance Initiative (September 13, 2005)

[12] Nancy R. Mead, Dan Shoemaker (University of Detroit Mercy), Carol Woody, Principles and Measurement Models for Software Assurance, 2013 IJSSE Special Issue on Cybersecurity Scientific Validation

[13] National Institute of Standards and Technology, "SAMATE—Software Assurance Metrics and Tool Evaluation" [http://samate.nist.gov] (Gaithersburg, MD: NIST)

[14] NIST SP 800-53, https://web.nvd.nist.gov/view/800-53/home

[15] Port, D.; Wilf, J., "The Value Proposition for Assurance of JPL Systems," Procedia Computer Science, Volume 28, 2014, Pages 398-403, ISSN 1877-0509,

[16] Risk-Based Measurement and Analysis: Application to Software Security, Christopher J. Alberts, Julia H. Allen, Robert W. Stoddard

[17] Shoemaker, Dan., & Mead, Nancy. 2013. *Software Assurance Measurement – State of the Practice* (Technical Report CMU/SEI-2013-TN-019). Pittsburgh: Software Engineering Institute, Carnegie Mellon University. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=72885

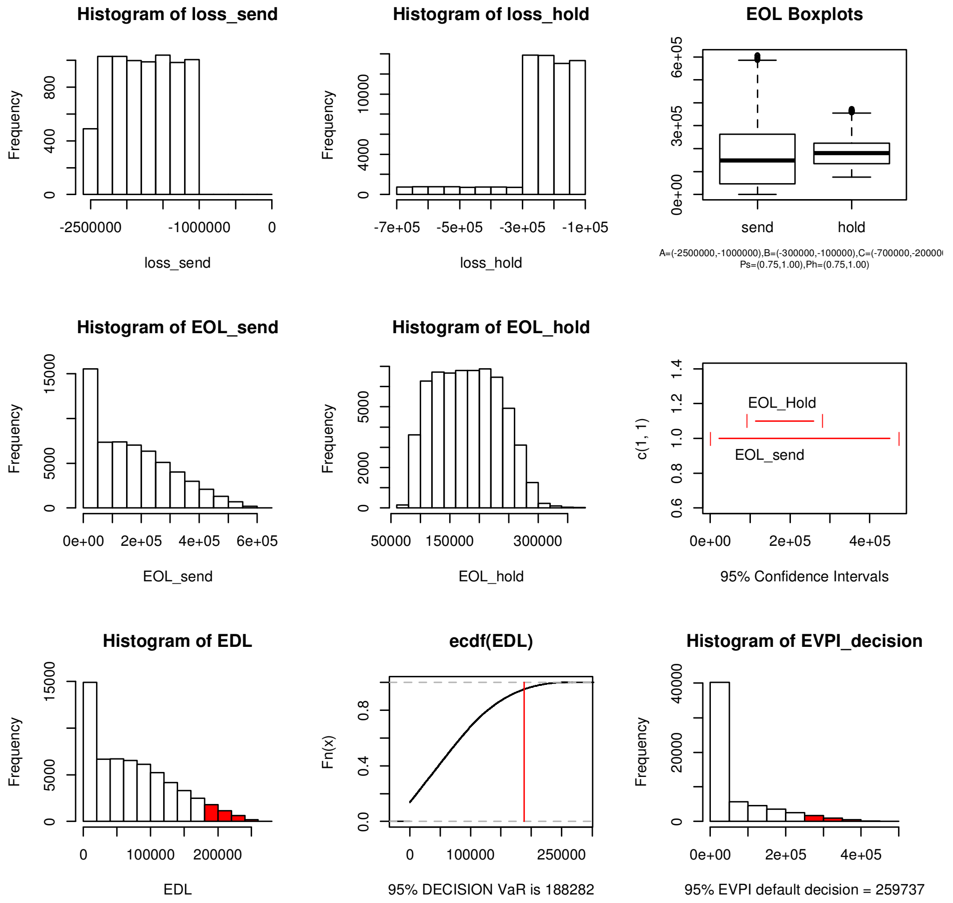
[18] State-of-the-Art Report (SOAR) July 31, 2007, Information Assurance Technology Analysis Center (IATAC), Data and Analysis Center for Software (DACS) (https://buildsecurityin.us-cert.gov/resources/dhs-software-assurance-resources/soar-on-software-security-assurance)

[19] Taber, W., Port, D. (2014). Empirical and face validity of software maintenance defect models used at the jet propulsion laboratory. IEEE International Symposium on Empirical Software Engineering and Measurement.

[20] *The Value Of Credentialed Vulnerability Scanning*, http://www.tenable.com/blog/the-value-of-credentialed-vulnerability-scanning

[21] Thurman, *Compliance does not equal security,* Computerworld, Jan 12, 2016

Appendix A: Decision Risk Monte Carlo Simulation Results


Histogram of loss_send
Frequency
loss_send
Histogram of loss_hold
Frequency
-7e+05
-5e+05
-3e+05
-1e+05
loss_hold
EOL Boxplots
0e+00
3e+05
6e+05
send
hold
A=(-2500000,-1000000),B=(-300000,-100000),C=(-700000,-20000
Ps=(0.75,1.00),Ph=(0.75,1.00)
Histogram of EOL_send
Frequency
0e+00
2e+05
4e+05
6e+05
EOL_send
Histogram of EOL_hold
Frequency
EOL_hold
c(1, 1)
EOL_Hold
EOL_send
0e+00
2e+05
4e+05
95% Confidence Intervals
Histogram of EDL
Frequency
EDL
ecdf(EDL)
Fn(x)
95% DECISION VaR is 188282
Histogram of EVPI_decision
Frequency
0e+00
2e+05
4e+05
95% EVPI default decision = 259737