SYM-AM-19-056

PROCEEDINGS
OF THE
SIXTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

WEDNESDAY SESSIONS
VOLUME I

Acquisition Research:
Creating Synergy for Informed Change

May 8–9, 2019

Published: April 30, 2019

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Risk Management and Information Assurance Decision Support

Hanan Hibshi—is a Research and Teaching Scientist at the Information Networking Institute at Carnegie Mellon University. Dr. Hibshi's research area includes: usable security, security requirements, and expert's decision-making. Dr. Hibshi's research involves using grounded theory and mixed-methods user experiments to extract rules for use in intelligent systems. Dr. Hibshi received a PhD in Societal Computing from Carnegie Mellon University, an MS in Information Security Technology and Management from the Information Networking Institute at Carnegie Mellon University, and a BS in Computer Science from King Abdul-Aziz University in Jeddah, Saudi Arabia. [hhibshi@cmu.edu]

Travis D. Breaux—is an Associate Professor of Computer Science, appointed in the Institute for Software Research of the School of Computer Science at Carnegie Mellon University. Dr. Breaux's research program searches for new methods and tools for developing correct software specifications and ensuring that software systems conform to those specifications in a transparent, reliable and trustworthy manner. This includes demonstrating compliance with U.S. and international accessibility, privacy and security laws, policies and standards. Dr. Breaux is the Director of the Requirements Engineering Laboratory at Carnegie Mellon University. Dr. Breaux has several publications in ACM- and IEEE-sponsored journals and conference proceedings. Dr. Breaux is a member of the ACM SIGSOFT, IEEE Computer Society, and USACM Public Policy Committee. [breaux@cs.cmu.edu]

Abstract

Like any organization, the DoD still relies on security analysts who can ensure that security requirements are satisfied. Relying on one expert's opinion can be risky, because the degree of uncertainty involved in a single person's decision could increase with time, memory failure, or inexperience. In previous work, we introduced the multifactor quality measurement method (MQM) where we reduce this risk by collecting security ratings from multiple experts with documented expertise in specific technical areas of cybersecurity. The next step is to automate the scenario generation where less experienced IT personnel can create scenarios that correspond to their own system architecture using our tool. The automation allows one to crowdsource security assessments from experts. The tool will collect and analyze the expert ratings and return the results to the original requestor. In this paper, we propose our designed prototype for the tool and we share the results of evaluating the prototype on 30 students who are completing a master's degree in cybersecurity at Carnegie Mellon University. Based on the qualitative and usability analysis of responses, our proposed method is shown effective in systematic scenario elicitation. Participants had a 100% task completion rate with 57% of participants achieving complete task-success, and the remaining 43% of participants achieving partial task-success. Finally, we discuss our findings and future directions for this research in systematic scenario elicitation.

Introduction and Background

Organizations, including the DoD, rely on security experts to evaluate system security and determine appropriate mitigations (Garfinkel, 2005, p. 5; Hibshi, 2016; Hibshi, Breaux, & Broomell, 2015). Despite the abundance of requirements that are available in security checklists and control sets, such as the NIST 800-53 control set ("NIST/ITL Special Publication (800)," 2015) security analysts continue to rely on their own experience and background knowledge when analyzing system security (Hibshi et al., 2015; Hibshi et al., 2016). Checklists are convenient because they generally apply to systems; however, they

lack the context needed to assess the threat against a specific configuration (Haley et al., 2008). Claims that negative events are unlikely is difficult without being explicit about one's trust assumptions (Haley et al., 2008). Moreover, mapping the checklist to threat scenarios or other requirements is laborious process repeated by an analyst for each system. Finally, security requirements are not independent; instead, they work together in composition with different priorities and inter-dependencies to improve overall security (Garfinkel, 2005, p. 5).

Recently, we examined the effect of context and requirements composition on security requirements expert ratings (Hibshi et al., 2015; Hibshi & Breaux, 2017). In that work, we used factorial vignettes in which requirements and system constraints are variables in a scenario description. We use scenarios from four technical areas: networking, operating systems, databases, and web applications (Hibshi et al., 2015; Hibshi & Breaux, 2017). The result is a new method that we call the multifactor quality measurement (MQM) method. The MQM process, which relies on using scenarios expressed in natural language text, would greatly benefit from introducing automation. The automation would involve using a tool where less experienced IT personnel can create scenarios that correspond to their own system architecture. The IT personnel could crowdsource security assessments from experts, and the tool would then analyze the collected data and send the results back to the IT personnel.

In this paper, we prototype the tool for scenario elicitation from IT personnel. Since eliciting scenarios in natural language text format can be an ad hoc process with possible ambiguity, we build our tool prototype using a scenario language based on a simplified process model of iterative scenario refinement. The model consists of three steps: (1) eliciting an interaction statement that describes a critical action performed by a user or system process; (2) eliciting one or more descriptive statements about a technology that enables the interaction; and (3) refinement of the technology into technical variants that correspond to design alternatives. In the upcoming sections of this paper, we will provide more details about the prototyped model and the results of its evaluation.

Systematic Scenario Elicitation

We now describe our approach to study the activity of systematic scenario elicitation. The approach assumes a model of structured scenario elicitation that results in a user story (Cohn, 2004) in natural language text that we refer to as scenario throughout this paper. To describe the model, consider the example text scenario shown in Figure 1. The example starts with an *interaction statement*, which is a statement that describes a critical action performed by a user or a system process. The *interaction statement* used in the example is specific to a domain (healthcare) but can also be stated more generically with no domain. Next, appears the *descriptive statement*, which describes a technology that enables the interaction.

For any type of technology, based on the stakeholder's needs and environment, there could be a variety of design alternatives to identify. To accommodate this diversity, the model allows a stakeholder to define a *variable* for a technology and list the design alternatives as different *levels* of that variable. In the example shown in Figure 1, we define a $Network variable with three possible levels.

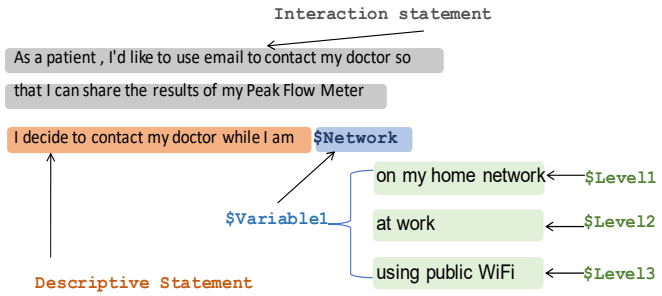


Figure 1. Example of a Text Scenario

The model is intentionally limited to these three elements: *interaction statement*, and one or more *descriptive statements* that each contains a variable with *levels*. This limitation is necessary to identify and isolate sources of error in scenario generation. In the future, one could imagine studying more advanced scenarios with nested levels of interaction and description.

Stakeholder Input

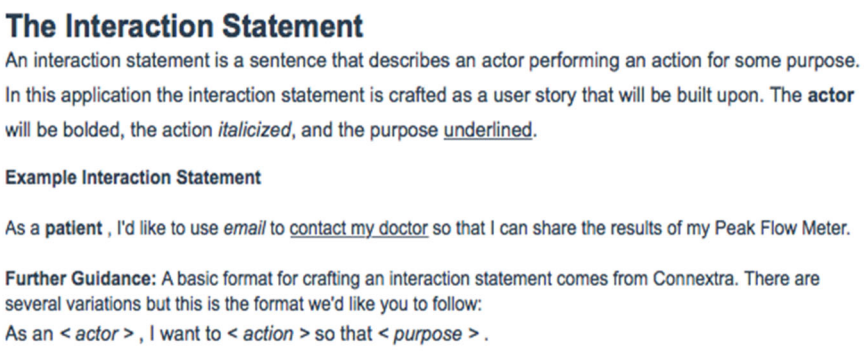
To elicit scenarios from stakeholders, our approach involves three steps corresponding to the model elements described above:

1. *Interaction statement elicitation*: where stakeholders are asked to provide a domain of interest and a related interaction statement in the following format:

   As an **< actor >**, I want to **< action >** so that **< purpose >**.

2. *Descriptive statement(s) elicitation*: where stakeholders are asked to provide one or more descriptive statements.
3. *Technology refinement:* where stakeholders define variables to represent the chosen technology and define a number of levels representing different design alternatives. After defining their own variables, stakeholders are asked to rank these variables based on a certain quality (e.g., security).

Scenario collection from users is completed online through online forms that prototypes the forms used in the design of the tool. The scenario elicitation process is accompanied with explanatory text and training material. For example, we use the text shown in Figure 2 to explain interaction statements to stakeholders. We follow a similar approach to explain the descriptive statements, the variables, and the levels.



The Interaction Statement

An interaction statement is a sentence that describes an actor performing an action for some purpose.

In this application the interaction statement is crafted as a user story that will be built upon. The **actor** will be bolded, the action *italicized*, and the purpose underlined.

Example Interaction Statement

As a **patient** , I'd like to use *email* to contact my doctor so that I can share the results of my Peak Flow Meter.

Further Guidance: A basic format for crafting an interaction statement comes from Connextra. There are several variations but this is the format we'd like you to follow:

As an < *actor* > , I want to < *action* > so that < *purpose* > .

Figure 2. Training and Example Text for the Interaction Statement of a Text Scenario

Evaluation of the Model

We designed a prototype and test the model on stakeholders in the form of an online survey. The survey consists of several forms that corresponds to the forms used in the prototype. Our target population is stakeholders interested in the cybersecurity domain. At the beginning of the survey, we explain to participants that the end goal of these tasks is to construct a *vignette,* which we define to participants of the survey as: *a story that people read before making an important decision. The vignette adds context to help the person make a more informed decision*.

Going through each step in the model, we provide stakeholders with definitions and running examples to help understand the concepts needed to perform the task related to that step (see Figures 1 and 2). The study participants are asked to provide their input following each explanation and training. For example, following the training shown in Figure 2, participants are asked to provide an interaction statement for their domain of interest (they have been presented with training materials and example domains prior to being introduced to the interaction statement).

Upon task completion, we ask participants to rate their own experience performing the tasks in the user study. We ask them to rate the difficulty of each individual task on a 7-point scale. In addition, we ask participants about the likelihood (using a 7-point scale) of using a tool for scenario creation that is similar in design to the exercise that they just completed. We repeat this likelihood-of-use question twice: for someone inside the participant's organization, and for someone outside the participant's organization. This repetition encourages participants to think more broadly about the possible broader benefits of the tool prototype that they just have tried even if they do not see a direct benefit to themselves in using such tool. We also allowed participants to provide additional open-ended comments.

Lastly, we ask participants to answer 14 security knowledge questions and standard demographic questions (e.g., gender, age, and years of experience).

We recruited participants from who are enrolled in a well-recognized information security master's degree program in a top university in the United States. Each participant was compensated with a $25 Amazon gift card.

Analysis of Participant Responses

We are interested in the effectiveness, efficiency, and user-satisfaction of the proposed three-step scenario elicitation model. We next describe how we analyze and measure these components:

- **Effectiveness** is concerned with a stakeholder success in completing a task while maintaining an acceptable level of accuracy (**Frøkjær**, Hertzum, & Hornbæk, 2000). In our results we measure effectiveness using task completion rates. To account for task accuracy, we differentiate between *full task success*, where participants complete the task with no missing information or errors; and *partial task success* where participants complete the task with some errors or missing information.
- **Efficiency** is concerned with the resources a stakeholder consumes to complete a task while maintaining an acceptable level of accuracy (**Frøkjær** et al., 2000). In our study, we use *task completion time* to measure efficiency.
- **Satisfaction** is concerned with stakeholders' attitudes when using a system (**Frøkjær** et al., 2000). To measure participants satisfaction with our model, we use rating scales to ask study participants to provide their perception of task difficulty and their projection of likelihood-of-use.

The constructs shown above rely on qualitative analysis of study participants responses. We use grounded analysis (Corbin & Strauss, 2007; Glaser, 1978) and coding theory (Saldaña, 2012) to code participants open-ended, text responses. The following is an explanation of how we analyzed the data to help measure the three constructs listed above and to provide qualitative insights.

- **Domains:** Participants were asked to list their domains of interest and the interaction statement. Using open coding, we review participant answers and categorize the elicited domains into a broader domain category. For example, the forensics domain is categorized into the broader domain of cybersecurity, and the banking domain is categorized into the broader domain of finance (finance can include corporate investment for example).
- **Interaction Statement:** A full interaction statement should contain the actor, action, and purpose. We coded interaction statements as *complete* if the participant provides a full interaction statement, and *incomplete* if participant provided an interaction statement that is missing the purpose. We coded empty responses with N/A, and non-statement responses (e.g., words and phrases) as not provided.
- **Descriptive Statement:** A correct descriptive statement should follow the format shown in the example shown in Figure 1 and must contain a variable preceded by the ($) sign. We coded descriptive statements as *correct* if the participant provides a descriptive statement using a format similar to the training, *partial* if the participant provides partial text that still can be comprehensible as a descriptive statement but is missing the variable or the dollar sign ($) preceding the variable, and *incorrect* if otherwise. We also coded the relationship between descriptive statements and interaction statements with one of the following codes: *related* if a strong relationship can be derived from the text; *semi-related* if the relationship can be derived but is not obvious; and *not related* if otherwise.
- **Variables:** Initially, we coded a variable *correct* if it correctly represents a technology that can have multiple design alternatives (levels), and incorrect otherwise. Later, we added the code: *level* if the variable is not perceived as a broader category of its level, but rather is perceived as another level (e.g., the variable “home network” is coded as level, if the participant provides “employer network” and “public network” as levels). Variables that are missing the dollar sign ($) are coded as *partial*.
- **Variable/level structure:** We coded the structure as *correct* if the participant provided variables and levels in the expected format where variables are a broader technology category of the levels, and we coded the variable/level structure to be *incorrect* if otherwise.

Training material used in the experiment includes an example of a $Network variable with three possible levels (see Figure 1). The levels shown to participants are technical variants of different network configurations that vary in their security strength (some levels are more secure than others). For each variable/level combination, we assigned codes that best describe the relationship between the levels and the variable they are supposed to refine. In cases where the variable is missing or wrong, then we code the relationship between the levels themselves. The codes, or concept labels, follow the Glassier view of *open coding*, wherein the codes emerge from the data without any pre-defined initial code set (Glaser, 1978).

Inter-Rater Reliability

When coding qualitative data that is subject to different interpretations, it is recommended to use multiple raters and calculate inter-rater reliability where researchers use statistical measures like Cohen's Kappa to measure above chance agreement (Cohen, 1968) and be able to judge the quality of the code set being used (Cohen, 1968; Saldaña, 2012). We use two coders for our data set (the first and second authors), and we calculate Cohen's Kappa for each coded data type separately. Our calculated Kappa averaged at 0.9, which is considered good agreement (Cohen, 1968). Next, the disagreements were resolved to reach complete agreement to finalize the dataset for analysis.

Results

We now present our analysis results. We collected scenarios from 30 participants. The mean time that a participant used to complete the scenario elicitation tasks including training is 24 minutes.

Demographics

All participants have a bachelor's degree in computer science or a related field and are currently enrolled in a graduate information security program at a top U.S. university. Out of the 30 students, three participants already work for industry and one works for the U.S. government. The mean score for participants on the security knowledge test is 58%. Table 1 summarizes the demographics statistics of study participants.

Table 1. Demographics Information

Description		Participants	
		Number	*Percentage*
Gender	Male	21	70%
	Female	8	27%
	Prefer not to say	1	3%
Years of Computer Security Experience (Mean=2)	Less than 1	6	20%
	1–2 years	13	43 %
	3–4 years	7	23 %
	5–7 years	4	13%
Age range	18–24	18	60%
	25–34	12	40%
Took job training in security		27	40%
Self-taught security knowledge		12	57%
Security Knowledge Score	Scored above 60%	12	31%
	Scored between 40% and 60%	16	41%
	Scored below 40%	2	5%

Task Completion

All 30 participants completed the user study from start to end, and they provided a domain of interest. The task completion rate that maps to our research questions is related to constructing a scenario using the three steps of providing an interaction.

We define three completion categories: full completion when a participant completes the interaction statement and at least one descriptive statement with its associated variables and levels with full accuracy; partial completion if a participant completes the interaction statement and at least one descriptive statement with its associated variables and levels

with partial accuracy; and failure if a participant did not provide an interaction statement and did not provide any description statements with an associated variable. Since our evaluation of responses relies on qualitative analysis, we show in Table 2 how we classify full accuracy vs. partial accuracy based on the codes used in the grounded analysis.

Based on our definitions above, our study data shows that 57% of participants achieve full completion (17 responses), 43% achieve partial completion (13 responses), and 0% failures.

When analyzing the 13 partial completions, we found four participants providing incomplete interaction statements that did not include a purpose, five participants did not precede the variables with a dollar sign ($), three participants used another level instead of a broader category for levels, and one participant who provided a variable with levels that do not relate or show a clear variable/level structure.

Table 2. Tasks Accuracy Definitions Based on Codes

Coded Task	Codes		
	Full accuracy	Partial accuracy	Failure
Interaction statement	complete	incomplete	Not provided, N/A
Descriptive statement	correct	partial	incorrect
Variable	correct	partial, level	incorrect

Participant Satisfaction

We measure participants interaction using participants ratings of task difficulty and likelihood of use. All 30 participants provided ratings for task difficulty and likelihood of use, and only eight participants provided additional open-ended comments.

Task Difficulty

Table 3 summarizes the participant feedback about the task difficulty involved in scenario creation. For the first four tasks: understanding vignettes (i.e., scenarios), understanding interaction statements, crafting interaction statements, and understanding descriptive text; almost half (between 48–63%) of participants were skewed toward easy ratings (somewhat easy, easy, and very easy combined). For the later four tasks shown in Table 3, participants feedback is less skewed in any direction. By assigning numeric values to the 7-point scale (with 1=Very Easy and 7= Very Hard), we found that the mean value for all tasks ranges between 3.1 and 3.9, which is slightly below Neutral (Neutral=4), leaning towards the easy category.

Table 3. Participants Feedback About Task Difficulty

Task	ery Easy	asy	omewhat Easy	eutral	omewhat Hard	ard	ery Hard
Understanding vignettes	13%	33%	7%	27%	17%	3%	0%
Understanding interaction statements	7%	27%	30%	20%	10%	3%	3%
Crafting interaction statements	3%	14%	31%	24%	21%	0%	7%
Understanding descriptive text	3%	20%	33%	20%	17%	3%	3%
Crafting descriptive text	0%	13%	30%	13%	40%	3%	0%
Understanding variables	7%	17%	17%	30%	23%	3%	3%
Crafting variables	7%	7%	23%	23%	30%	7%	3%
Understanding levels	10%	13%	13%	27%	17%	10%	10%
Crafting levels	7%	13%	20%	27%	17%	7%	10%

Likelihood-of-Use

Table 4 summarizes participant feedback about the likelihood of using a tool similar to what was presented in the study by the participants themselves or someone else inside or outside their organization. In general, participants were slightly more skewed towards unlikely. Three participants explained in their open-ended comments that they did not fully understand the end goal of the tool presented in the survey. By looking at their performance, these three participants still managed to complete the required tasks. These observations suggest that participants might not been able to project the benefit of using the language proposed in the tool, which affected their projection of likelihood-of-use.

Table 4. Participants' Feedback About Likelihood of Using a Vignette Generation Tool

If this tutorial was integrated into an online tool for crafting vignettes that can be used later for running user study, how likely	Very Unlikely	Unlikely	Somewhat Unlikely	Neutral	Somewhat Likely	Likely	Very Likely
would YOU use such a tool	10%	13%	23%	17%	13%	23%	1%
would someone IN your organization use such a tool	10%	3%	7%	33%	27%	17%	3%
would someone OUTSIDE your organization use such a tool	17%	10%	13%	17%	30%	7%	7%

Discussion, Future Work, and Conclusions

In this paper, we introduced a language for scenario elicitation that is based on a three-step model that elicit structured parts of natural language text from stakeholders. When the natural language text parts are combined, the end result is short scenario template with a variable that can take different values of varying levels of technologies. The varying technologies allow us to compare different technology alternatives that can be further evaluated by other analysts, stakeholders, or domain experts. We present results from our evaluation of a user study where we examine the usability of our introduced method. Our analysis results for this preliminary study suggest a promising future in this area, because we had no empty responses or failures. The task completion is 100% divided between 57% full accuracy, and 43% partial accuracy.

Unlike previous research in requirements engineering where scenarios were produced from formal representations that more closely correspond to models, our method relies on guiding stakeholders to create scenarios presented in natural language text. Using a structured approach in collecting statements has shown a benefit in collecting scenarios that share similar syntax and differ in semantics. This uniformity has a number of benefits, as follows:

- **Scalability and more systemized collection process,** where a requirement engineer can tailor our method based on the domain of interest and use it to collect natural language scenarios from a larger participant pool. Systemizing natural language scenario elicitation offers more scalability and coverage compared to collecting unstructured stakeholder narratives.

- **Homogenous stakeholder scenarios** that result from using a structured approach in our method. Scenarios written in natural language are known to be more user-friendly to the stakeholder, but without proper structure, the process becomes ad-hoc and scenarios will be highly heterogenous with no unifying pattern that can help an analyst parse different scenarios. In our results, all elicited scenarios shared a common structure, even in cases where participants had partial accuracy.
- **Systemized scenario analysis,** which is a result of the homogeneity feature of scenarios collected using our proposed method. Following a uniformed syntax is a feature that facilitates the parsing of natural language text, which allows requirements engineers to analyze and validate scenarios using systemized means and automated tools. In our experiment, we were able to systematically analyze the data and we found the process to be less time consuming than analysis done on unstructured natural language text collected, for example, in user interviews and focus groups.
- **Real capture of stakeholder experiences and domain knowledge** because our method allows stakeholders to write scenarios using natural language text, where they only learn a certain structure to arrange their words. In our experiment results, the security domain knowledge was evident in the elicited scenarios.

Going forward, our future research involves introducing more automation to the tool. We envision that using our tool, an analyst would be able to build their own scenario and then send out invitations for experts to rate the overall security and the individual security requirements, and to provide further requirements that can enhance the ratings. Such a tool would have a great impact on the DoD and other organizations in the public and private sectors, because it would help systemize the evaluation of security components using real experts' input.

References

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213.

Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Corbin, J., & Strauss, A. (2007). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: Sage.

Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000). Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 345–352. The Hague: The Netherlands: ACM.

Garfinkel, S. (2005). *Design principles and patterns for computer systems that are simultaneously secure and usable.* Cambridge, MA: Massachusetts Institute of Technology. Retrieved from http://dspace.mit.edu/handle/1721.1/33204

Glaser, B. G. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory*. Mill Valley, CA: Sociology Press.

Haley, C. B., Laney, R., Moffett, J. D., & Nuseibeh, B. (2008). Security requirements engineering: A framework for representation and analysis. *IEEE Transactions on Software Engineering*, *34*(1), 133–153.

Hibshi, H. (2016). Systematic analysis of qualitative data in security. In *Proceedings of the Symposium and Bootcamp on the Science of Security* (p. 52). https://doi.org/10.1145/2898375.2898387

Hibshi, H., Breaux, T., & Broomell, S. B. (2015). Assessment of risk perception in security requirements composition. In *Proceedings of the 2015 IEEE 23rd International Requirements Engineering Conference (RE)* (pp. 146–155).

Hibshi, H., & Breaux, T. D. (2017). Reinforcing security requirements with multifactor quality measurement. In *Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 144–153). Lisbon, Portugal: IEEE.

Hibshi, H., Breaux, T. D., Riaz, M., & Williams, L. (2016). A grounded analysis of experts' decision-making during security assessments. *Journal of Cybersecurity*. Retrieved from http://cybersecurity.oxfordjournals.org/content/early/2016/10/04/cybsec.tyw010.abstract

Hibshi, H., Breaux, T. D., & Wagner, C. (2016). Improving security requirements adequacy: An interval type 2 fuzzy logic security assessment system. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8). Retrieved from http://ieeexplore.ieee.org/abstract/document/7849906/

NIST/ITL Special Publication (800). (2015, January 2). Retrieved from http://www.itl.nist.gov/lab/specpubs/sp800.htm

Potts, C., Takahashi, K., & Anton, A. I. (1994). Inquiry-based requirements analysis. *IEEE Software*, *11*(2), 21–32.

Saldaña, J. (2012). *The coding manual for qualitative researchers*. Thousand Oaks, CA: Sage.

Sutcliffe, A. (1998). Scenario-based requirements analysis. *Requirements Engineering*, *3*(1), 48–65.

U.S. Bureau of Labor Statistics. (2016, March 8). *Information security analysts: Occupational outlook handbook*. Washington, DC: U.S. Bureau of Labor Statistics. Retrieved from http://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm

Van Lamsweerde, A. (2000). Requirements engineering in the year 00: A research perspective. In *Proceedings of the 22nd International Conference on Software Engineering* (pp. 5–19). Retrieved from http://dl.acm.org/citation.cfm?id=337184

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CA 93943

www.acquisitionresearch.net