Composite Security Requirements in the Presence of Uncertainty

Hanan Hibshi

CMU-ISR-18-102

October 2018

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee
Travis D. Breaux, Chair
Lorrie Faith Cranor
Stephen B. Broomell
Dongrui Wu (Huazhong University of Science and Technology, China)

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

The research reported in this thesis has been supported in part by the graduate scholarship from King-Abdul-Aziz University, Jeddah, Saudi Arabia. It has also been supported by ISR at Carnegie Mellon using the following: Award No. H9823014C0140 by the National Security Agency, Awards No. HQ00341810014 and N00244170012 by the Office of Naval Research, Awards No. W911NF0910273, W911NF1310154 by the Army Research Office, Award No. CCF1453139 by the National Science Foundation, and the Software Engineering Institute.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of King-Abdul-Aziz University and the U.S. Government.

Keywords: security, requirements, decision-making, situation awareness, user study, vignettes, scenarios, security requirements, qualitative analysis, context, empirical study, fuzzy logic, type-2, uncertainty

Abstract

Providing secure solutions for information systems relies on decisions made by expert security professionals. These professionals must be capable of aligning threats to existing vulnerabilities to provide mitigations needed to minimize security risks. Despite the abundance of security controls, guidelines, and checklists, security experts rely mostly on their background knowledge and experience to make security-related decisions. In this thesis I explore how security experts make security-related decisions, collect their assessments of security measures nested in scenarios, and extract security mitigation rules. These rules could be used to build an intelligent fuzzy logic intelligent system, which captures the knowledge of many experts in combination. I present the Multi-factor Quality Measurement (MQM) method that I introduced to the field of requirements engineering to empirically elicit and analyze security knowledge from experts. This is done by using user-studies that instruments factorial vignettes to capture the experts' assessments of mitigations in scenarios composed of many components affecting the decision-making process. The results are analyzed quantitatively with multi-level modeling in order to capture the weights and priorities assigned to security requirements, and qualitatively to explore new or refined security requirements.

The outcome of the analysis will be used to generate membership functions for a type-2 fuzzy logic system. The corresponding fuzzy rule-sets encode the interpersonal and intra-personal uncertainties among experts in decision-making.

I explore security decision-making in presence of: composite security requirements, varying expertise, and uncertainty. This work makes methodological contributions on two aspects: empiricism, where I adapt different data collection and analysis techniques adapted from other interdisciplinary fields and apply it to requirements engineering; and modeling, where I explore a data-driven modeling approach that can fit data collected from experts in the security domain, where the experts are scarce and the amount of data collected is not sufficient to use machine learning.

Acknowledgments

I want to thank my thesis committee members for their support and valuable feedback that helped me become the researcher I am today. I would also like to thank the members of the Requirements Engineering Lab at Carnegie Mellon University, the CUPS lab, and my research collaborators: Dr. Christian Wagner, Dr. Stephen Broomell, Dr. Laurie Williams, Dr. Maria Riaz, Dr. Jennifer Cowley, Dr. Jianwei Niu, Dr. Rocky Slavin, Jean-Michel Lehker, and Dr. Florian Schaub.

Special gratitude goes to my advisor, Dr. Travis D. Breaux, for his advice and remarkable support throughout my PhD journey. Dr. Breaux's support went beyond research feedback to help advise me on professional and sometimes personal matters to help me grow as a researcher and continue to succeed after my PhD. I am fortunate to work with such a dedicated, hardworking advisor who leads by example and has a great sense of humor!

I am indebted to my thesis committee member and my role model Dr. Lorrie Cranor, who I came to know before starting my PhD when she advised my Master's thesis. Without Dr. Cranor's advice and support, I would not have pursued my PhD journey. Dr. Cranor has been always a great mentor providing valuable advice on professional and personal levels. I met Dr. Cranor while I was completing my Information Security Master's degree at the Information Networking Institute (INI) at CMU. The INI program opened the door for me to pursue my research career. I am particularly thankful for the INI program director, Dr. Dena Haritos Tsamitis, for her commitment to support me as an INI alumni, and as woman in engineering. In addition, I would like to thank Dr. Carol Frieze and extend my gratitude for women support organizations at CMU: WINI and W@SCS.

At the Institute for Software Research where the Societal Computing PhD program is housed, I would like to thank all the ISR staff for their prominent support throughout the years of my PhD. Special thanks go to Margaret Weigand, and Connie Harold. I also like to thank faculty at the ISR in the Societal Computing and the Software Engineering programs, the ISR director Dr. William Scherlis, and the SC program director Dr. Jim Herbsleb.

I am grateful for student colleagues in CMU. My PhD journey would have been less exciting without the support and encouragement of fellow students. Special thanks to my RE lab colleagues: Jaspreet Bhatia and Morgan Evans.

Outside CMU, I am grateful for my wonderful friends who continue to support me in many ways. I am also deeply indebted to my father, Dr. Hisham Hibshi, my mother, Tamadhur Al-Hariri, my brother, Loai Hibshi, and my Grandmother, Madiha Rajab for their continuous encouragement and strong support.

This PhD would have never been a success without the support of my beautiful family. My husband, Muhannad Al-Hammad contributed a lot to this work by supporting my career choices, continuously encouraging me, and making me smile and laugh during the most stressful of times. I am also grateful for my wonderful kids who filled my PhD journey with moments of joy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Despite the abundance of well-documented security best practices, we continue to see security breaches that affect different organizations and industries. The 2014 OWASP Top 10 Application Security Risks report shows that attacks are occurring due to the exploitation of common, well-documented vulnerabilities, such as injection and cross-site scripting attacks [95].

Organizations rely on the judgment of security experts to evaluate the security of their systems. Despite the abundance of well-documented security best practices, such as the NIST Special publication 800-53 that lists 256 security controls [93], security experts rely on their own expertise and tacit knowledge to assess the security risk and provide recommendations [67, 68, 69]. The analyst must often reason over potentially millions of scenarios that account for various permutations of network type, services offered, threat type, etc. When requirements change by adding new components and features, these risk calculations must be updated [68]. What is not known is how changes in threats and requirements affect the analyst's ability to perceive changes in risk and their ability to identify and prioritize security requirements. In addition, security experts in the world are scarce. There are about 100,000 information security analysts in the U.S. in 2016 according to the U.S. Bureau of Labor statistics [120], and there is an expected 58% growth in demand by 2018 [112]. The scarcity of experts and the need for cybersecurity as the number of information security incidents keeps increasing, makes the provision of intelligent decision support and semi-automated solutions a necessity.

This thesis investigates how do design secure software when: 1) security expert-knowledge is stove-piped; 2) security decisions involve multiple factors (e.g. risk analysis, attacks, vulnerabilities), 3) uncertainty is present degrees in human decisions, and 4) the number of experts in security is limited and difficult to grow in sufficient time, which limits the volume of data collected. To address these challenges, I used mixed quantitative and qualitative methods from multiple disciplines. I collected data using interviews, surveys, and user studies that employ factorial vignettes and mixed methods designs. For data analysis, I have used the grounded analysis from social science, the theory of situation awareness established in psychology, and the statistical multi-level modeling. I model the analysis results using fuzzy logic, which is a formal method used in the computational intelligence community.

Thesis Statement

The increasing complexity of security attacks takes advantage of three challenges to making reliable security assessments: 1) security experts' knowledge is typically stove piped, 2) security against specific threats is achieved through composition of multiple requirements, 3) security-decisions carry a measurable degree of uncertainty, and 4) the limited number of security experts. This thesis examines security requirements composition in presence of uncertainty and attempts to extract and model experts' knowledge in the form of rules. The theoretical outcome is a repeatable methodology to create risk assessment models that conform to the real world, while the practical outcome is a step towards understanding how to automate and improve security recommendations.

Below, I will provide a summary of the technical contributions of thesis.

Exploring Challenges in Security Decision-Making

To reflect on the slow, deliberative decision-making process of security analysts, I have obtained qualitative data by interviewing security experts (some with over 10 years of experience) and used grounded analysis on interview transcripts to discover patterns of situation awareness (SA), a decision-making theory from cognitive psychology that decomposes decision making into four states: perception, comprehension, projections, and decision. In my work, I validated the decision-making model of situation awareness and the discovered patterns show that analysts try to handle uncertainties using assumptions and prior knowledge. In addition, the analysis show that even when presented with a checklist of requirements, experts' in practice tend to put security requirements in a context by creating their own scenarios and analyzing potential vulnerabilities and attacks. This work revealed the opportunity for further studies that measure how changes in certain factors, such as a network configuration, and password requirements can increase or decrease the experts security rating. Our work also revealed that experts who combine hands-on industry experience with academic knowledge exhibit different patterns of situation awareness compared to novices who rely on academic background alone. The experts made assumptions when faced with uncertainty, while novices asked the interviewer for additional details. The experts also demonstrated patterns in which they adopt an attacker's perspective, but novices failed at demonstrating this perspective. These patterns may be useful to design tests to measure whether novices can be effectively trained to reach expert SA. Researchers and practitioners may find results from this work useful in facilitating and improving training for novices. This work is explained in detail in Chapter 3

Establishing the scientific validity of Ad-hoc Security Measures

To ask experts about security decision-making, we must first decide on the measure to be used for security. Our initial expert interviews (described in Chapter 3) show that experts were hesitant to describe a feature or a component as secure or insecure, and they preferred to say "it depends". We examined a list of possible scales to describe security, and concluded by choosing adequacy of security requirement(s) as a metric. In our studies, we used the adequacy metric on a semantic scale including anchors for more or less adequacy. To align our semantic scale with intervals, we invited 38 security experts where we asked each expert to provide an interval for the word on a scale from 1-10 while imagining that the word describes a security scenario. The collected intervals show that the labels: adequate, inadequate and excessive cover the entire scale from 1-10 when modeled using type-2 fuzzy sets. I will explain these studies in more detail in Chapters 4, 5, and 6.

Capturing the Effect and Priorities of Composed Security Requirements

Asking security experts about decision-making has five challenges: 1) composition, risk assessment of a system must consider the system context in which the requirements apply, and the composition of requirements with components of a system; 2) priorities, some requirements have higher priorities than others, depending on their strength in mitigating threats; 3) ambiguity in abstract terms that could lead two experts to interpret a requirement differently; 4) stove-piped knowledge, security expertise crosses different domains, such as hardware, software, cryptography, and operating systems; and 5) the scarcity of security experts. To address these challenges, I developed the Multifactor Quality Measurement method (MQM), which models dependencies among requirements, and estimates how these requirements affect a perceived level of quality in a requirements specification, called a scenario. The MQM process starts with ad hoc bootstrapping of scenarios using factorial vignettes, a social science method where scenarios are constructed using a template consisting of factors of interest. By treating the factors as variables and manipulating the variables and their levels, we generate different instantiations of the template. Generating multiple vignettes allows us to elicit more information from a smaller number of experts. This study design has greater statistical power (increasing power reduces the probability of errors) because it includes both within-subject and between subject effects. In addition, the data is analyzed later with multi-level modelling which limits the biased covariance estimates, and hence, increases power. The manipulation of factors/levels allows researchers to study the effect of changing security requirements on adequacy ratings, to identify dependencies, and to prioritize requirements based on the factor contribution to the overall effect. For example, results of the multi-level modeling suggests that, although experts realize that displaying detailed error messages to end-users is an insecure approach that exposes internal vulnerabilities to hackers, their overall security ratings were slightly improved when the scenario had a stronger logging and monitoring mechanism. To close the security knowledge gap, the MQM approach helps analysts elicit new requirements from experts that have been experimentally shown to monotonically increase security. Researchers and requirements analysts can benefit from applying this process to their case studies to measure security or any quality of interest. I present this work in detail in Chapter 5.

Data-driven Approach for Modeling Expert Knowledge

Formal Modelling of security knowledge is necessary to build a decision-support system or an intelligent system in general. The security expert judgments will always contain a degree of interpersonal uncertainty e.g., in which two experts providing different judgments and intrapersonal uncertainty where the same expert provides different judgments over different times. The uncertainty in the data is a characteristic that cannot be ignored, but rather need to be modeled, because it represents the diversity of opinions of experts. Type-2 fuzzy sets model the uncertainties, both interpersonal and intrapersonal. To use this logic to reason over uncertain decisions, I introduce a method to build a type-2 fuzzy logic rule-set with reduced size. I use the expert data to retain realistic permutations of input/output, and exclude unrealistic permutations. My contributions to the computational intelligence community is in the rigorous approach to eliciting and modeling real expert data, and in the application to cybersecurity, which I will explain in further detail in Chapter 6

Finally, I will provide conclusions from the work in this thesis along with future work and possible research directions in Chapter 7.

Chapter 2

Background and Related Work

Haley et al. describe security as a wicked problem [62]. Wicked problems are those difficult to solve problems due to unclear, ambiguous, or conflicting requirements [31, 62]. Wicked problems are challenging, because the space of possible solutions are difficult to enumerate [33], and this is the challenge that faces analysts when addressing security problems. Security analysts may respond differently to the same security problem, and they also may find different resolutions to discrepancies represented in the problem. For example, analysts can look at the same artifact that describes a network architecture, whereby one analyst might assess the security of the authentication mechanisms, while another is focused on encryption mechanisms. With such wicked problems, DeGrace and Stahl, and Detoit et al. suggest that the design of solutions should be aimed at reducing ambiguity by reaching a collective understanding of the problem representation [31, 33].

In this chapter I focus on background and related work that highlights the challenges that make security decision-making a wicked problem.

We believe that there are three factors that make security analysis a wicked problem: how security requirements work together, which we call composition; the varying levels of expertise maintained by experts themselves; and the uncertainty that is present to some level in security decisions. In the remainder of this section we will first explain the security risk quantification problem, because the goal of security analysis is to minimize the risk. Next, we will explain the problem with current security checklists. Lastly, we will discuss the role of security expertise in decision-making and how requirements composition, expertise differences, and uncertainty affect the analyst decision-making process.

2.1 Security Checklists

Security guidelines and best practices are widely available and documented in a checklist. For example, the U.S. National Institute of Standards and Technology (NIST) Special Publication (SP) 800 series describes best practice security requirements [93], and the Common Criteria describes a method to evaluate system security. In particular, the NIST SP 800-53 lists 256 security controls, which security analysts can apply in a checklist by deciding whether the control applies to their system. To make this decision, the analyst must reason over potentially millions of scenarios that account for various permutations of network type, services offered, threat type, etc. Hence, the problem is not the lack of security guidelines but that they are not usable. When requirements change by adding new components and features, these risk calculations must be updated. What is not known is how changes in threats and requirements affect the analyst's ability to perceive changes in risk or their ability to identify and prioritize security requirements. Checklists only list the requirement that could decrease the risk, whereas mapping the requirement to certain threat scenarios or to other requirements must be done by the analyst independently. In addition, the context in which requirements exist in composition with priorities and dependencies among each other is also missing from the checklists and it is the security analyst's job to figure out the context and the underlying dependencies [51].

Repeatable solutions in security require a certain level of abstraction. An abstract solution exists regardless of the underlying technology(s), and this is what provides more stability for a system [51]. For example, the Open Web Application Security Project (OWASP) is an organization that provides software security checklists in its online materials that help developers to reduce the security risk by applying security best practices to their software [95]. However, the technical solutions here are fine-grained to the program-level, where it is challenging for the average developer to infer the abstraction. These specific solutions in guidelines are only applicable as long as the specific technology exist, and once new technologies appear, the solutions may not be applicable to the new technology. What is needed here is the abstract solution that can be applied in similar contexts independent of technical details so the solution will remain stable no matter how the technology changes. Software design patterns are a good example of abstract solutions [4, 50, 114], although, more work is needed to understand how analysts fit security patterns to problems.

2.2 Quantifying Security Risk

The U.S. National Institute of Standards and Technology (NIST) defines security risk to researchers as the product of likelihood and impact: the likelihood of a threat to occur on a resource, and the impact of the threat occurrence on the organization [116]. There has been a number of efforts where researchers suggest methods that help assess and quantify the security risk according to the NIST definition [2, 72, 78, 116]. However, these and other approaches are criticized for not solving the security problem [51, 52] as our systems continue to be compromised by attackers [52]; and some researchers question the feasibility of such approaches [52]. For example, Butler and Fischbeck propose a multi-attribute risk assessment process that uses an additive value models that allows managers to rank order threats [17]. The authors argue that this approach helps prioritize requirements [17]. However, the proposed model relies on the input of one security manager and assumes that the knowledge of the manager is complete. As I explain in the upcoming sections, security experts knowledge varies and differences among experts can affect their judgment of requirements. Another limitation to Butler and Fischbeck's approach [17], is that it ranks requirements in an ordered list without accounting for dependencies and interactions among requirements.

The rapid growth in technology and data calls for new approaches for security risk assessment. Garfinkel argues that the existing different approaches to risk assessment are not feasible in practice, because we cannot put an exact number on impact and likelihood of adverse events and that is the reason why many organizations use catalogs of best practices as a way to minimize the security risk [51].

Research in risk quantification aims to define what is secure enough, but the challenge with risk remains: measuring to what extent a requirement or mitigation is *sufficient*. Chung, and Mylopoulos et al. suggest that security requirements could only be satisficed as opposed to satisfied [25, 92]. Since organizations are in need for risk assessment, they rely on security best-practice checklists to perform their analysis [51], and they probably consider the checklist to be their *sufficient* or *enough* threshold.

Checklists, however, as Garfinkel points out, lack the context or situation where the security best practices exist in [51]. Putting security requirements in a context impacts the analysts risk perception [61]. Haley et al. asserts that it is more feasible to assess security risk and reason about satisfaction of a security requirement in the context of a given situation as opposed to reasoning in a broader context, because it is harder to claim that a negative event is never going to happen [61]. In the upcoming Chapter 5 of this thesis, I will show the effect of composing security requirements in scenarios on experts risk assessment.

2.3 Security Risk Assessment as a Wicked Problem

Security problems are often assessed by experts who are responsible for reviewing a system specification, and deciding what mitigations will mitigate security threats. Experts are also responsible for making sure companies are in compliance with security guidelines, such as NIST 800-53. This practice is affected by the analyst expertise and their ability to make decisions about security requirements that exist in composition. Composition means that the requirements do not exist independent of one another; instead they exist in a context with dependencies and priorities among related requirements. Adding or removing a requirement affects other requirements in that context. For example, if an organization decides to open a web access port to its in-house system that was closed in the past, this would affect the authentication mechanisms, the access control policy, passwords, and so on. As we have mentioned earlier, the composition of requirements and context in which they exist is missing from conventional representations of guidelines and it is addressed later during the security risk assessment.

In addition to composition, security risk assessment is affected by the analyst own expertise, and the level of uncertainty that might exist in the decisions they make. Below, we will explain the four factors that affect security analysts risk assessment: expertise, scarcity of experts, composition, and uncertainty. We believe that these factors contribute to making security risk assessment a wicked problem:

2.3.1 Security Analysts Knowledge and Expertise

Experts rely on tacit knowledge to conduct an analysis. Security experts are not all equal in their knowledge and skill set. For example, security knowledge can be acquired from specialized courses, on-the-job training, or self-study. In addition, some experts may be more specialized in certain areas of security, such as web-security or mobile security. Ben-Asher and Gonzalez

[14] examined how the knowledge gap between novices and experts affect the analyst ability to detect cyber attacks as the experts performed significantly better than novices. To detect attacks successfully, cybersecurity experts need: 1) domain knowledge [22, 42, 59] that is obtained through formal academic learning and practical hands-on experience with tools; and 2) situated knowledge which is organization dependent and which analysts tend to learn through continuous interaction with certain environments [58, 59, 111]. We elaborate more on security expertise in Chapter 3 as we show our results from 11 interviews of security experts during the conduct of security assessments.

Defining a Security Expert

Distinguishing experts from novices is not a straight-forward process [43]. Finding the proper metrics to defining experts, measure their expertise, and distinguish them from novices remains a challenge in research and is affected by many factors including the domain of interest [43]. Expertise refers to the set of skills or characteristics that distinguish knowledgeable individuals in certain domains from the general population [43]. Self-assessment of expertise is not a reliable approach, because of the risk of overconfidence bias or the “Dunning-Kruger [77]" effect, wherein individuals with less expertise may fail to recognize their own weakness or knowledge gaps, which may result in inflated self-assessments (also known by social psychologists as illusory superiority) [77].

It is important, however, to assess the knowledge of experts who are providing input in empirical research because the level of expert knowledge could affect their judgment. Edwards and Tversky suggest that a decrease in knowledge could lead to an increased judgment uncertainty [35].

Using knowledge tests in any domain requires researchers to be cautious of some confounding factors that could skew or bias the results. For example, age could be one confounding factor; a study has shown that older physicians are morel likely to score lower on knowledge tests compared to younger physicians [24]. Knowledge tests that heavily focus on theory could also introduce some bias; a study performed on military technical personnel, found low correlation between the scores on the knowledge test and the actual performance in troubleshooting the technical problem.

In the security field, researchers assessed the knowledge of participants in security-related studies and there has been some evidence that security knowledgeable participants are not always following the most secure behavior on their own devices [109, 127]. Wash and radar surveyed 1993 Internet users in the United States and found that users might hold to security beliefs that are not correlated to their security knowledge, which means one cannot rely on using security knowledge to predict secure online behavior [127]. In another study that investigates users security behavior, Sawaya et al. used a set of 18 security knowledge questions to assess the security knowledge of 3,500 participants from seven countries [109]. The study results suggest that users confidence in their security knowledge had a more positive effect on their security behavior compared to their actual security knowledge [109]. As I explain later in Chapter 5, we use security knowledge tests to assess the security knowledge of experts participating in our studies. Being aware of possible biases highlighted above, we avoid screening participants based on the knowledge tests. Instead, we use the knowledge tests to as a tool to measure experts’ security knowledge and to provide descriptive statistics about the experts in our sample population. In the upcoming chapters of this thesis, I will explain our approach for the knowledge assessment of experts recruited in our different studies.

2.3.2 Security Requirements Composition

Expert tacit knowledge in security includes domains, such as cryptography, network security, web security, mobile security, database security, and malware analysis, among others. It is challenging to find one expert in all these areas, combined. Understanding complex attacks, for example, requires knowledge combined from a number of security fields and understanding how the “pieces of the puzzle” compose together [14, 68]. Stuxnet is a good example where the attack targeted networks with hosts that run the Windows operating system and Siemens Step7 software [21]. This attack, which targets vulnerabilities found on network hosts, proves that focusing on strengthening the security of the network alone is not sufficient as other factors, such as the hosts, their operating systems, and other connected components, need to be taken into consideration when performing the security risk assessment [52]. This broad understanding helps analysts to determine the proper requirements that work together to mitigate attacks. For example, stronger passwords with rules of 16 alphanumeric and special characters could be considered a good security requirement, but this cannot be an absolute rule. The type of password relies on other factors such as: the type of network where the connection is made, the sensitivity of the data involved, and so on [68]. In Chapter 5, we elaborate on this effect when we report the results of our studies.

2.3.3 Uncertainty in Security Decisions

The research paradigm in software engineering is shifting towards recognizing uncertainty as a first-class concern that affects design, implementation, and deployment of systems [53]. Garlan argues that the human in the loop, mobility, rapid evolution, and cyber physical systems are possible sources of uncertainty [53]. These sources of uncertainty affect the analyst security assessment. In this thesis, the focus is on the uncertainty in expert security assessments that could be interpersonal and intrapersonal. Interpersonal uncertainty exists between different experts as experts can judge the same situation differently. Intrapersonal uncertainty is the uncertainty within an analysts own judgment [88]. For example, an expert might describe a security requirement to be adequate. The uncertainty that this expert has about whether the combination of factors are themselves adequate is intrapersonal uncertainty, because the same experts might provide different judgments in two different times. The interpersonal uncertainty would be between two different experts would have different judgments of the situation and could disagree on the efficacy of the security requirement to mitigate an attack.

2.3.4 Security Experts Scarcity

The number of security experts in the world is scarce. According to the U.S. Bureau of Labor statistics, there is around 100,000 information security analysts in the U.S. in 2016, earning a median income of $95,510 a year [120]. Employment is projected to grow by 28% by 2026, which is a faster growth rate then average [120], and 56% growth in demand for security analysts is projected by 2026 [120].

Chapter 3

Exploring Challenges in Security Decision-Making

It is important to first explore the problem of security-decision making, and identify how different factors like uncertainty and expertise level could affect the decision-making process. Qualitative data that are rich with details and context are a good fit for exploratory research.

In this chapter, I describe expert interviews as part of an exploratory study (published in the Journal of Cybersecurity [69]) that is analyzed qualitatively using *grounded analysis* [29, 55] and the *theory of situation Awareness (SA)* [39, 41]. The study was conducted with 11 security experts [69] to understand how experts form decisions. We examined responses to the same artifacts with and without checklists, a prominent security requirements analysis method. We developed a novel coding method to apply Situation Awareness (SA) to interview data, to understand how security experts choose appropriate security requirements. The results include decision-making patterns that characterize how analysts perceive, comprehend and project future threats against a system, and how these patterns relate to selecting security mitigations. Based on this analysis, we discovered new theory to measure how security experts and novices apply attack models. The results also highlight the role of expertise level and requirements composition in affecting security decision-making.

3.1 Motivation

Security analysts review different artifacts of a system and decide on proper mitigations based on their security risk assessment. As I have pointed out in the introduction and related work, these reviews are affected by the analysts own expertise and differences among analysts could lead to different security decisions. In addition, the lack of information system security is unlikely due to an absence of documented security requirements. The ISO/IEC 27000 Series standards and the U.S. National Institute of Standards and Technology (NIST) Special Publication 800 Series are examples of documents that contain best practice security requirements. Combined with the

Excerpts from this work were previously published as H. Hibshi, T. D. Breaux, M. Riaz, and L. Williams, *A Grounded Analysis of Experts Decision-Making during Security Assessments*, Journal of Cybersecurity, 2016.

wealth of available security knowledge, we hypothesize that insecure information systems persist because security analysts experience two challenges: a) they experience difficulty in perceiving relevant risks in the context of their information system designs; and b) they experience difficulty in deciding which requirements are appropriate to minimize risk.

In the upcoming sections of this chapter, I will provide background information about the theory of situation awareness, explain the research methodology, and present results followed by a discussion on the impact of this research.

3.2 Situation Awareness and Security Risk Assessment

Situation Awareness (SA) is framework introduced by Mica R. Endsley in 1988 [39] that distinguishes between a user's "*perception* of the elements in the environment within a volume of time and space, the *comprehension* of their meaning, and the *projection* of their status in the near future" during their engagement with a system. Perception, comprehension and projection are called the *levels* of SA, and a person ascends through these levels in order to reach a decision. To illustrate, consider an SQL injection attack, in which an attacker inserts an SQL statement fragment into an input variable (often via a web form) to gain unauthorized database access. When an analyst conducts a source code vulnerability assessment, they look for cues in the code for where to place input sanitization, which is a kind of mitigating security requirement. Upon finding such cues (perception), the analyst proceed to reason about whether the requirement has or has not been implemented (comprehension). Once understood, they can informally predict the likelihood of an SQL injection attack and the consequences on the system (projection) based on their experience and understanding of the threat and attack vector.

We believe SA can be used to explain how analysts perform risk assessments. The NIST Special Publication 800-30 [93] defines risk as the product of the *likelihood* that a system's vulnerability can be exploited and the *impact* that this exploit will have on the system. The ability to predict likelihood and impact depend on the analyst's ability to project events based on what they have perceived and comprehended about the system's specification and its state of vulnerability. If the expert succeeds in all three SA levels, then they have *good* SA and they should be able to make more accurate decisions about security risks. Failure in any level results in "poor" SA that leads to inaccurate decisions or no decisions at all. We will describe below our method to detect the SA-levels in security expert interviews.

Endsley and other researchers [39, 40, 41]go beyond the SA definition to establish a holistic framework that scientists in other fields could benefit from and apply. This framework entails details and relationships to other concepts such as: expertise effect, goals, mental models, automation, uncertainty, and requirements analysis. A *schema* in cognitive psychology is defined as the mental framework in human cognition to prepossess ideas that represent some aspects of the world [6, 7, 12]. *Schemata* are a group of schemas organized in cognition that improve a human's ability to retrieve knowledge or acquire new knowledge [6, 7, 12]. For example, when we solve new problems using a computer programming language, schema theory suggests that our cognition matches the new problem structure with existing schemata for solving past problems and this process is what cognitive psychologists call: *schema abstraction* [71]. Rao et al. found that the number and variety of training examples in programming language experiments had minimal effect on schema abstraction [103]. Thus, we may conclude that schema abstraction is an expert ability that is acquired over multiple, repetitive examples across different contexts. Endsley explains how expertise can help a person to build and enhance mental *schemata*, which facilitates the person's ability to interpret their perceptions and make necessary projections that lead to better decisions [41].

3.2.1 Related Work in SA

The SA framework is flexible and could be customized according to the needs of a system. Examples of fields in which SA has been applied include military operations [32], command and control [46], cybersecurity [20, 34, 57, 74, 98] and others [41, 110]. Researchers have modeled SA in intelligent and adaptive systems [32, 46, 110]. Feng et al. proposed a context-aware decision support system that models situation awareness in a command-control system [46]. Their approach was to have agents based on "rule-based inference engines" that provide decision support for users. They applied Endsley's concepts and focused on "shared situation awareness" along with a computational model that they applied to a case study of a command and control application.

In the field of requirements engineering, Alkhanifer and Ludi [5] followed a recommendation by Endsley and Jones [41] to use the Goal-Directed Task Analysis (GDTA) for user knowledge elicitation. The authors applied the GDTA to user goals and sub-goals during elicitation about a system to improve orientation of the visually impaired while they navigate unfamiliar buildings [5]. Our approach of applying SA is different, as we are using the SA stages to code interview scripts to draw relationships that explain how requirements analysts make decisions early in design. An approach to SA, that to our knowledge has not been widely adopted in requirements engineering specifically.

Cyber SA Related Work

There has been multiple research efforts to use Situation Awareness to study the cybersecurity field. Chen et al. extended a cyber intrusion detection system using a formalization of SA concepts; the logic formalization is derived from expert experiences [20]. Jakobson proposed a framework of situation aware multi-agent systems that could be cyber-attack tolerant [74]. A *Cyber Situation Awareness* model [34] was introduced to simulate a security analyst in a network. The proposed model relies on using Instance-based learning (IBL) and starts by recognizing events in the network, and compares these events to past events stored in the analyst's memory. The model relies on the past threat experience to predict and detect network threats [34]. The simulation results had shown that the model was affected by the defenders approach to risk (risk-seeking/risk-aversion), and the authors note the difficulty of validating the simulation results against real human data on real networks that could contain proprietary data [34].

Paul and Whitley proposed a "taxonomy of cyber situation awareness questions" that represents the analyst's mental model. The authors conduct interviews to elicit questions that analysts ask themselves, then a card sorting activity was used to help categorize the questions into groups. The authors used qualitative methods to analyze the data and graph co-occurrence visualizations to represent the results [98]. The results of the analysis was used to build a taxonomy of cyber situation awareness questions where questions were categorized based on their co-occurrence score.

Gonzalez et al. [57] argue that to computationally represent human situation in cybersecurity, it is essential to develop cognitive models that are capable to dynamically adapt, adjust, and learn from experience and predict unforeseen situations. The authors further argue that cognitive models offer an advantage over statistical approaches (e.g. machine learning), that have the limitations of being confined to information derived from existing data and their dependencies, without the capability to adapt to the dynamics of human cognition, such as learning processes and short-term sequential dependencies [57]. In our SA study, we study the security analyst to understand factors that affect their cognitive mental model, and how an analyst analyzes dependencies among multiple components of a system to reach a decision in a dynamic, risky cybersecurity environment. The intention is to build on insights and hypotheses derived from this exploratory study, to learn how to represent the human reasoning computationally.

3.3 Using SA to Explore Security Decision-Making

We chose the definitions of SA levels to be our basis for the grounded analysis that we perform on the interview data of 11 security experts [69]. Below we provide an overview of our approach that consist of three phases:

- The *preparation* phase, in which we developed the research protocol, including tailoring SA to security analysis, selecting the system artifacts to use in the analysis, and recruiting the security analysts to be interviewed;
- The *interview* phase, wherein we elicited responses from selected analysts; and
- The *qualitative data analysis* phase, in which we coded the interview transcripts and systematically drew inferences from the data.

We applied grounded analysis using coding theory [107] to link SA concepts to the dataset and validate whether our observations are consistent and complete with respect to that dataset [29, 107]. In the first cycle, we applied the *hypothesis coding* method to our dataset [107] using a predefined code list derived from Endsley's SA levels; this method tests the validity of the initial code list. In the second cycle, we applied theoretical coding to discover decision-making patterns from the dataset. We now discuss the three phases.

3.3.1 The Preparation Phase

The SA framework can be tailored to a field of interest by mapping SA levels to statements made by domain analysts. We tailored the framework by verbally probing the analyst during the interview as they were asked to evaluate the security risk of information system artifacts. We expected the dataset to show how analysts build situation awareness. We also expected it to help us further discover how perceptions of security risk evolve as the analysts' awareness of both potential vulnerability and available mitigations increases. The inability to perceive risk may be due to limitations in analysts' knowledge or ambiguities in the artifacts. We map Endsley's SA levels to security analysis as follows:

Level 1: Perception:the participant acknowledges perceiving security cues in the given artifact. Examples include:“there is a picture of a firewall here” or “there are SQL commands in the code snippet.” Each observation excludes any deeper interpretation into the meaning of the perception.

Level 2: Comprehension: the participant explains the meaning of cues that they perceived in Level 1. They provide synthesis of perceived cues, analysis of their interpretations, and comparisons to past experiences or situations. Examples of comprehension include: “the firewall will help control inbound and outbound traffic...” and “the SQL commands are used to access the database which might contain private information, so we need to check the input to those commands, but this is not done in the code...”

Level 3: Projection: the participant has comprehended sufficient information in Level 2, so they can project future events or consequences. In security, projections include potential, foreseeable attacks or failures that result from poor security. Examples include: “this port allows all public traffic, which makes the network prone to attacks... ”, or “unchecked input opens the door to SQL injectionâĂę”

Finally after Level 3, we expect participants to make security-related decisions. Decisions include steps to modify the system to mitigate, reduce or remove vulnerabilities. Continuing with the SQL injection example, one decision could be: “this port should be closed” or “a function should be added here that checks the input before passing it to the SQL statement.” Closing the port prevents an attacker from exploiting the open port in an attack, whereas checking the input can remove malicious SQL in an SQL-injection attack.

Selection of Security Artifacts

We presented each participant with three categories of security-related artifacts: source code, data flow diagrams, and network diagrams (artifacts are listed in the AppendixA.1). We chose these artifacts to cover a broad range of security knowledge, from low-level source code to high-level architecture, noting that security requirements should be mapped to each artifact in different ways and analysts require different skills to do this mapping. Based on our own experience and knowledge of security expertise, we considered the effect of specialization in areas such as secure programming, network security, and mobile security in selecting these artifacts. Hence, the selection aims to satisfy two goals: 1) to account for diverse background and experience; and 2) to assess whether different artifacts show differences among SA levels. We selected artifacts that are typical examples comparable to what is generally taught in college-level security courses. We now describe the artifacts used in this study:

(a) *Source Code (SC)*. We present participants with JavaScript code snippets, corresponding SQL statements, and a picture of a web user interface related to the snippet. The SC contains two vulnerabilities, an SQL injection attack and unencrypted username and password. JavaScript is a subset of a general purpose programming language, i.e., no templates, pointers, or memory management. Thus, we expect analysts with general programming language proficiency and knowledge of SQL injection to be able to spot these vulnerabilities in the SC. We also list a high-level security goal to prompt participants and we ask participants if the goal has been satisfied.

(b) *Data Flow Diagram (DFD)*. We present participants with a DFD for installing an application on a mobile platform. As shown in Figure 3.1, the diagram contains high-level information about the data flow between the user, app developer and the market. The participants are asked about possible security requirements to ensure secure information flow, and whether they can evaluate those requirements based on this diagram.

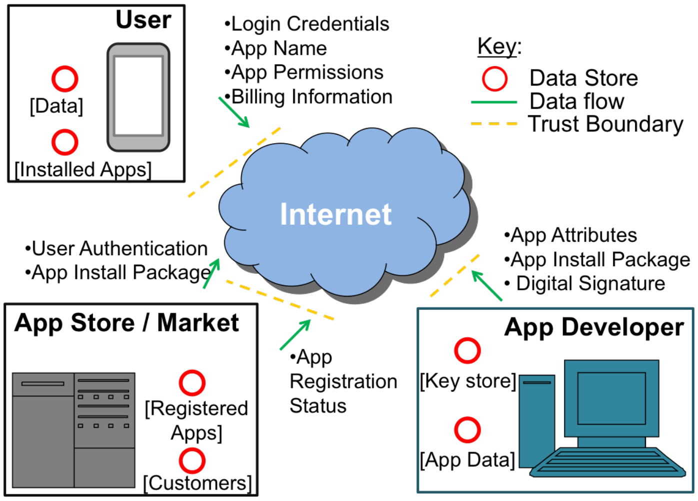


Figure 3.1: The Data Flow Diagram Artifact

(c) *Network Diagrams (ND)*. We present participants two network diagrams: ND1 shows an insecure network, and ND2 shows a network with security measures that address weaknesses in ND1. After participants are provided time to study ND1, we present ND2 and ask participants to evaluate whether ND2 is an improvement over ND1. After collecting data on participants evaluation of ND2, we present 15 security requirements to participants, which we explain to be part of a security improvement process, and we ask participants to assess whether the network in ND2 satisfies the 15 requirements (shown in Appendix A.2).

All of the selected artifacts are typical examples comparable to what is generally taught in college-level security courses. For example, the network diagrams were originally used in the Applied Information Assurance Class taught by Christopher May at Carnegie Mellon University [85].

Selection of Security Experts

In this study, we aim to observe how security expertise affects requirements analysis. However, security analysts are not all equal in expertise; some analysts have more experience than others in particular areas, and training in academia is different than hands-on practice. To cover a broad range of expertise, we invited industrial practitioners and Ph.D. students at different stages of matriculation, all working in security. We will present the demographics data later in this section.

3.3.2 The interview Phase

We designed the interviews to study how analysts reach a security-related decision, and not to study the correctness of the decision or degree of security improvement. We chose this design to reduce a participant possible anxiety about being personally evaluated. During our interviews, we only ask the following kinds of questions:

- What cues did the participant look at? *(Perception)*
- How were the cues interpreted? *(Comprehension)*
- Why did they interpret a cue that way? *(Comprehension)*
- What are the future consequences of each interpretation? *(Projection)*
- Based on those projected consequences, what is the best practice? *(Decision)*

Our approach differs from how SA is traditionally studied in human operator environments (e.g., airplane cockpits and nuclear power plants) that use the Situational Awareness Global Assessment Technique (SAGAT) [41], in that our participants are not immersed in a simulation per se. Rather, we present artifacts (SC, DFD, ND1 and ND2) to participants with prompts to evaluate artifacts for vulnerabilities asking them to act as the security analyst in this setting. We observe their ability to conduct requirements analysis, their proposed modifications or decisions, and their evaluation of security requirements satisfaction.

In addition, we ask participants to share information about their decision-making, such as unstated assumptions and what artifact cues led participants to reach a decision. We were careful not to guide participants in a particular direction by keeping our questions general. In addition, we avoided questions such as: what do you perceive, comprehend, or project? For example, if a participant identified an attack scenario, we would follow with "why would you think such an attack could occur", or "could you describe how it could happen?" Based on our approach to limit our influence on their responses, we found participants returning to the artifact to identify cues and to explain their interpretation.

We present ND1 before ND2, and we ask participants to draw on ND1 to improve this diagram. After this step, we show participants the secure diagram ND2 and ask them to compare this diagram to their own solution to ND1. Then, we ask participants to review the requirements list (shown in Appendix A.2), and to answer the following questions for each requirement:

- Is the requirement satisfied or not satisfied based on the information given in the diagram?
- How would the participant evaluate the security requirement: is it good, bad, unnecessary, immeasurable, unrealistic, etc.

The questions above are asked in a conversational style with an open-ended fashion where participants are free to comment, explain and elaborate in their answers.

Finally, given our interest in distinguishing novices from expert analysts, we asked participants to provide a brief description of their relevant background. Questions to elicit background information were asked twice: first, at the interview start, we ask participants about their security background, their education, industry experience, and security topics of interest. Lastly, at the end, we ask the participant about the analysis process they used during the interview and how it relates to their background. We audio recorded the interviews for transcription and analysis.

3.3.3 The Grounded Analysis Phase

Grounded analysis is used to discover new theory and to apply existing theory in a new context [29]. We apply grounded analysis in three steps: (1) we transcribe the interviews; (2) beginning with our initial coding frame (see Table 3.1), we code the transcripts by identifying phrases that match our codes, while discovering new codes to further explain phrases that do not match our preconceived view of the data; and (3), we review previously coded datasets to ensure the newly discovered codes were consistently applied across all transcripts. After piloting the initial study design on two participants, we observed uncertainty among participants so we added codes to capture the uncertainty. Table Table 3.1 shows the complete coding frame: the first eight codes (P, C, J, D, including the variants that account for uncertainty U*) constitute the initial coding frame and were inspired by Endsley's terminology for the *Situation Awareness* [41]; the remaining four codes were discovered during our analysis to account for the interview mechanics. We employed two coders (myself and a co-researcher) who first met to discuss the coding process and coding frame, before separately coding the transcripts, and finally meeting to resolve disagreements. The process to resolve disagreements led to improvements in the form of heuristics that explain when to choose one code over the other in otherwise ambiguous situations. To efficiently identify disagreements, we used a fuzzy string-matching algorithm [8] to align the separately coded transcripts. Finally, each coder recorded their start and stop times.

To ensure all statements are coded, we applied the null code `{NA}`to any statements that did not satisfy the coding criteria, such as when participants request a scrap of paper to draw a figure, or when they ask how much time is remaining for the interview, and so on. We code statements, such as: "I took a course in security..." or "I saw on the news a security breach related to this artifact" as background `{BG}`, which includes their personal experience and knowledge. If the participant compares and contrasts comprehended information from the artifact to their experience or knowledge, then that information is coded as comprehension `{C}`. To improve construct validity, the two raters resolved borderline cases by discussing and refining the code definitions and heuristics. The following heuristics were used to classify statements and draw clearer boundaries between coded data:

Perception: The participant verbally identifies a cue in the data (e.g. line number in code, an entity on the network diagram, a specific requirement in the text). Participants are only reporting what they see, and are not commenting or analyzing the cue.

Comprehension: The participant analyzes, makes inferences, or makes comparisons about what they see. This may include the name of the cue (e.g. firewall), but the statement at least includes an interpretation in addition to reporting the perception of the cue.

Projection: The participant forecasts future attacks, possible threats or any events that could occur based on the context found in the artifact.

Decision: The participant makes a decision with regards to the context. This includes deciding whether the system is secure or not secure, or if a certain requirement is satisfiable. Introducing

Table 3.1: Situation awareness annotation codes

Code Name and Acronym	**Definition and Coding Criteria Used to Determine Applicability of the Code**
Perception {P}	Participant is acknowledging that they can see certain cue(s)
Comprehension {C}	Participant are explaining the meaning of cue(s) and conducting some analysis on the data perceived
Decision {D}	Participant is stating their decision
Uncertain Perception {UP}	Uncertainty at perception level: participant is missing certain data that would help they need to analyze the artifact
Uncertain Comprehension {UC}	Uncertainty at comprehension level: participant is not missing data but they can't interpret their meaning confidently
Uncertain Projection {UJ}	Uncertainty at projection level: participant cannot predict possible future consequences confidently
Uncertain Decision {UD)	Uncertainty in decision: participant is not confident about the decision that should be made
Assumption {A}	Participant is stating assumption(s)
Ask Question {Q}	Participant is asking the interviewer questions
Probe {Pro}	Interviewer is triggering the participant's thinking with questions or guidance information
Background {B}	Participant is providing information regarding their personal background
Null code {NA}	Statement is not applicable to code criteria above

new mitigations of security threats are also considered decisions.

Uncertainty (at any SA level): To determine if the participant is uncertain, first examine the verbal cues that indicate uncertainty, including, but not limited to: "I guess", "I am not sure", and "this is not clear to me". For example, the participant may indicate that they do not know what an icon represents. Alternatively, if the participant acknowledges that they see a cue, but that they cannot understand its role in the artifact, then this is an uncertain comprehension.

Assumption: The participant here needs to explicitly express that they are making an assumption. Examples of such statements include: "I am going to guess that this means", "I assume", "Based on my experience this means, but it's not necessarily what the artifact tells me" and so on. To clarify how to distinguish assumptions from comprehensions, a comprehension is when the participant is explaining a certain cue's meaning based on the information given in the artifact. Assumptions, however, provide further explanation based on the participant's experience with similar systems to compensate for missing cues or missing information in the artifact.

After the first cycle coding, we conducted a second cycle or axial coding [107] to identify decision-making patterns. In grounded theory, axial coding is the process of relating codes to each other by finding relationships, themes and phenomena that exist among the codes and categories [29, 107]. We defined cut-offs between coded sequences by sequentially numbering each statement and then assigning group numbers to statements that address the same idea or topic the participant is discussing. The groups serve to delineate transitions between units of analysis. We programmatically extracted SA-level sequences (e.g., `P-C` for perception followed by comprehension) that we later associated with separate, named patterns, and we searched the dataset without the cut offs to assess pattern validity, i.e., to detect false-positives, wherein the SA-level sequence does not correspond to the pattern that we assigned. We used the false positives identification to compute pattern accuracy, which is the ratio of true positives over the sum of true and false positives.

The next step in our grounded analysis includes labeling interviewee statements with entity identifiers from the specifications, such as variables and functions in the source code or servers and firewalls in the network diagram. The labeled artifacts allow us to sort our results by entity to see how different participants react to and analyze the same entity and to link the decision patterns to corresponding entities involved in the pattern.

3.3.4 Pilot Study

We piloted the study on two human subjects: participant P1 is an expert with extensive hands-on and academic expertise in networks and systems security; and participant P2 is a novice who has only academic security experience. The purpose of the pilot study is to test our interview protocol and apply any needed modifications to the questions or protocol before conducting additional interviews.

Reliance on assumptions and searching for more information are both uncertainty resolution techniques that are explained in Endsley’s SA approach [41]. However, it is interesting to see in our pilot results that experts and novices apply these techniques differently. Both participants P1 and P2 analyzed the network diagram artifact, but P2 was unable to provide technical details of a the network configuration and reported a higher number of uncertainties. Another insight observed in the pilot study was the ability of the more experienced participant P1 to make assumptions when faced with uncertainty. When the novice participant, P2, was faced with uncertainty, their solution was to ask the interviewer clarification questions. The following excerpt below is an example of an assumption that participant P1 made when they analyzed the requirement R9 that states implementing time synchronization for logging and auditing capabilities. Note that each statement will have an opening and closing code tags (see Table 3.1 for codes):

```
      {UP}I don’t see an NTP server on this network{/UP} {C}but I know that
 Windows Domain Controller can act as NTP{/C}, {A}so I am going to assume
 that when they install it they’ll probably leave that box checked because it
 ’s a default option{/A}.
      {D}I think that is probably happening here{/D}
```

When P2 was faced with uncertainty, however, they turned to the interviewer and asked:

```
{Q} What kind of software does this thing has? {/Q}
```

Because of the conversational interview style, participants went beyond verifying security requirements in ND2 to check consistency between the requirements and the network diagram present in the artifact, and they actually performed requirements validation, where they assess if the requirement(s) actually meets the stakeholders' system security goals). An explanation may be that security experts rely on background knowledge and apply known security requirements. In addition, we found experts often add missing requirements, explain how to apply a requirement, evaluate whether a requirement was feasible, list some needed specifications, and prioritize requirements. For example, consider the following excerpt as participant P1 is evaluating R2 in the context of diagram ND2 and pointing out that this requirement is less critical than requirement R1 that they had evaluated earlier:

```
{C}but I don't think it's as critical as say the DMZ one, but I think its
 sort of whatever is the next tier of criticality{/C}.
```

Based on our pilot study experience and the participants feedback, we revised our study protocol. A major change was the order of the presentation of network diagrams ND1 before ND2, and asking participants to draw on ND1 to improve this diagram. After this modified step, we show participants the secure diagram ND2 and ask them to compare this diagram to their own solution to ND1. Finally, we ask participants to review the requirements list, and to answer the following questions for each requirement:

- Is the requirement satisfied or not satisfied based on the information given in the diagram?
- How would the participant evaluate the security requirement: is it good, bad, unnecessary, immeasurable, unrealistic, etc.?

The questions above are asked in a conversational style in an open-ended fashion where participants are free to comment, explain and elaborate in their answers. Since this study is based on a qualitative research method, pilot data from P1 and P2 is included in our full analysis of data.

3.4 Evaluation of the Qualitative Approach

We recruited a total of 11 participants. In grounded analysis, reaching a point of *theoretical saturation* is the main determinant of the number of participants (or cases) needed to complete a qualitative study [55], because determining the exact right sample size is context-dependent on the type of research being conducted. According to Glaser and Strauss, *saturation* means that the researcher can not find additional data whereby the researcher "can develop properties of the category [55]." This means that a researcher can stop collecting more data once the analysis *saturates* and keeps showing repeated results and no more new insights, theories, themes, or findings emerge from the new data. Atran et al. [9] estimated that a minimum of 10 participants is needed to show consensus, while Guest et al. [60] argued that a sample size of six could be sufficient if there is a homogeneity that exists among participants in the sample. In our sample, we reached saturation after 8 participants, but we continued to recruit 3 more participants to confirm theoretical saturation.

Below, we report the results from our empirical evaluation, which consists of the artifact assignment and inter-rater reliability.

3.4.1 Artifact Assignment

Due to self-perceived inexperience by participants and time limitations, not every participant analyzed all artifacts in the three categories we described above. The average total interview time per participant to complete each interview was 29 minutes. Table 3.2 presents the participant assignment to conditions: the shaded cells show the category of artifacts that participants attempted; cells labeled with “X” indicate that the participant spent at least 15 minutes analyzing the artifact. Because participants have varying skills and expertise, some participants invested more time than others analyzing certain artifacts. The order in which the artifacts were presented to different participants was randomized and the time allowed to complete the interview was limited to 60 minutes. Thus, not all participants reviewed all artifacts. The Sum column in Table 3.2 presents the total number of participants who reviewed each artifact.

Table 3.2: Participants’ assignment by artifact

Artifact	**Participant**											**Sum**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	
1) Source Code		▨	▨ X	▨ X	▨	▨ X		▨ X		▨ X	▨ X	
2) Data Flow		▨ X	▨ X	▨	▨	▨ X	▨ X	▨ X				
3) Network	▨ X	▨	▨ X	▨ X	▨				▨ X		▨ X	

3.4.2 Agreement and Inter-Rater Reliability

Two raters applied the coding frame from Table 3.3 to the transcripts of participant audio recordings. We measured inter-rater reliability using Cohen’s Kappa, a statistic for measuring the proportion of agreement between two raters above what might be expected by chance alone [26]. We calculated Kappa for each participant, which ranges between 0.51-0.77 with a median of 0.62. These values are considered moderate to substantial agreement [26]. The coding times were 19 and 8 hours for raters 1 and 2, respectively. Rater 1 spent more time documenting heuristics and developing the method. In addition to the above time, 6 hours were used for the resolution of disagreements between the two coders. Table 3.3 shows the breakdown of the total 2,595 coded statements in our final dataset by code (including the pilot participants P1 and P2).

Table 3.3: Final dataset frequencies by code

Code	Total Codes	Code	Total Codes
Perception	250	Uncertain Percept.	82
Comprehension	498	Uncertain Comp.	180
Projection	215	Uncertain Proj.	13
Decision	367	Uncertain Dec.	25
Question	95	Probe	535
Background	47	Assumption	45
N/A	243		

3.5 The Discovered Security Decision-Making Patterns

In this Section, I present the discovered decision-making patterns that ground the SA framework in the data. Acronyms introduced in Table 3.1 are used to express the patterns as a sequence of coded observations across the interview transcripts. Findings from this section are going to motivate the discussion, analysis, relationship to expertise, and impact on security analysis that is presented in the remainder of this Chapter.

3.5.1 The Classic SA Patterns

Endsley suggests that experts who assess risky situations engage in a process of perceiving information, comprehending the meaning of that information, and then projecting what might occur in the future. We call this pattern the*Classic SA pattern*, which proceeds from P→ C→J→D, where the "→" means the coded statement on the left-hand side appeared adjacent and before the coded statement on the right-hand side in the transcript. In addition to the Classic SA pattern, we searched for contiguous fragments of the Classic SA pattern while the order is maintained, such as P→C→J, and C→J that indicate when a participant is move to higher levels of SA.

Table 3.4 presents the pattern name, number of occurrences (Freq.) and the accuracy (Accu.), which is the ratio of actual, confirmed pattern instances among the total number of observations of the sequence, and, finally, the list of participants who exhibited these patterns. We believe the pattern J→D is interesting because in combination with other patterns, we see variation fragments of the order appear. The results indicate that the J→D pattern only appears 31 times with 10% false positives. This observation suggests that projections and decisions, as well as other SA levels, can occur out of sequence, which motivated our search for the other pattern fragments shown in Table 3.4; all of these fragments are variations of the full Classic SA pattern (P→ C →J→D). We observed that participants demonstrated the J→D pattern without the P→C pattern component, but this does not mean that participants did not perceive cues or comprehended those cues. Instead, participants may not be verbally reporting their perceptions and comprehension, or they may have automatized these stages of SA as part of their prior experience.

Table 3.4: Variations of classic SA pattern

Name	Pattern	Freq.	Accu.*	Participants
Classic w/o Decision	{P→C→J}	4	100%	P1, P3, P6
Projection-Decision	{J→D}	31	90%	All except P1
Classic Skip Projection	{P→C→D}	10	100%	P1, P3, P4, P6, P11
Classic Skip Perception	{C→J}	55	81%	All
Classic Skip Perception and Projection	{C→D}	56	83%	All except P2 & P5
Classic Perception Comprehension	{P→C}	61	81%	All except P10

*Excluding false positives

Except for the first two patterns, a common feature among the patterns in Table 3.4 is the skip factor. Participants could skip a level of SA before reaching the next expected SA level. Because we coded participants' verbal responses, and participants may not have verbalized each level of cognition, our dataset may be missing the expressions of some levels. Another explanation for skipping levels is the level of expertise and exposure to the problem. If the participant has seen several examples of a certain problem, they may jump to their decisions immediately without providing explicit verbal analysis of the perceived cues, meanings and possible consequences. The following is an example from P3's response to the source code artifact where they immediately projected an SQL attack without perceiving or comprehending a certain cue (we use brackets [] to explain the item of the artifact that the participant is speaking about):

> {J} this [speaking about the line of code that shows the unsanitized input] is just pure SQL injection here {/J}

By comparison, P11 articulated moving from perception to projection while describing the same attack scenario:

> {P}And thus, [speaking about the line of code that shows the unsanitized input], you use SQL query that explicitly say its inserting into the customer value {/P}{J}it may suffer from the SQL injection attack. {J}

In contrast, the pattern (P→C→D) from Table 3.4 describes how a participant moves from perception to comprehension but jumps to the decision phase without describing the projection.

The patterns (C→J) and (C→D) bypass the perception level, where participants move from comprehension to either a projection or a decision phase. Based on our analysis, it is not unusual for participants to begin verbalizing at the comprehension level. In this case, participants begin by describing the meaning of a cue without explicitly identifying the cue. Consider the following excerpt from the coded response of P9 when they were analyzing the Demilitarized Zone in the network artifact:

> {C} ...people can access this part [speaking about the DMZ subnet in the network diagram] but it means de-militarized zone.{/C} {J}If these machines are hacked, they can't affect other inner parts{/J}

The last pattern in Table 3.4 reflects that participants move from the perception to the comprehension level, but without going immediately into projection or decision levels. We find this pattern interesting because it shows that someone could move back and forth between perception and comprehension without moving higher to projection or decision. This movement could indicate that a participant found themselves "stuck" at comprehension where they could not proceed further, because they lacked the needed cues, understanding to envision what comes next or how to mitigate a threat.

3.5.2 The Reverse SA Patterns

In our dataset, we observed that SA patterns might occur in reverse order. This difference may be due to the participant using an inductive vs. deductive reasoning style. Up until now, we assumed that participants used a deductive reasoning style: they first report perceiving a cue, comprehending the meaning, and from this information, they deduce and report what may occur in the future (projection). In an inductive reasoning style, the participant verbalizes the possible consequences and from this information, they work backward by inducing the cues that led them to this conclusion. To accommodate the inductive reasoning style, we checked the dataset for patterns in the reverse direction of the classic SA pattern. Table 3.5 presents the reverse SA pattern names, their frequencies, accuracy and participants who exhibited these patterns.

Table 3.5: Reverse SA patterns

Name	Pattern	Freq.	Accu.[*]	Participants
Reverse SA w/ Decision	{D→J→C→P}	None	None	None
Reverse SA w/o Decision	{J→C→P}	1	100%	P6
Reverse SA skip projection	{D→C→P}	3	67%	P6, P9
Reverse SA no perception	{J→C}	35	67%	All
Reverse SA no perception no projection	{D→C}	46	75%	All

[*] Excluding false positives

The following excerpt illustrates the reverse pattern exhibited by participant P6 who is analyzing the source code; the participant first reports their decision to prioritize a particular part of the diagram, followed by their understanding of this part and their perception of the part's character that led to the prioritization decision:

```
    {D}It's very important [speaking about using encryption for
communication over the Internet] {/D}{C} you're sending the SSN over the
Internet{/C} {P}The SSN is in plain-text. {P}
```

3.5.3 Patterns of Uncertainty and Assumptions

Uncertainty plays an important role in security, as many security risks are probabilistic and participants must estimate the likelihood of particular events when forming projections. Moreover, analyst experience is likely to play a role in interpreting ambiguity in a specification and then deciding whether that ambiguity includes an interpretation that may lead to a security exploit. Table 3.6 presents the uncertainty patterns that we identified in the data. These patterns consist of statements coded with uncertainty (UP, UC, UJ, and UD) and assumptions{A}, questions {Q }, and decisions {D}. The total coded subset relevant to this discovery is comprised of 440 statements across all participants. We categorized uncertainty into three categories:

- *Propagated Uncertainty* occurs in the first three patterns, wherein the uncertainty in perception or comprehension is propagated to a subsequent comprehension, projection or decision
- *Hedged Uncertainty* occurs in all patterns where uncertainty leads to assumptions (e.g.,U ⋆→A), in which case the analyst bounds the uncertainty by interpreting an ambiguity and concluding this interpretation in the form of an assumption; and
- *Uncertainty Transfer*, in which the analyst asks a question (e.g., U⋆→Q), to resolve uncertainty by seeking outside assistance.

With hedged uncertainty, 5 out of the 8 participants who made assumptions after their uncertain comprehension were able to make decisions. We found 9 instances of hedged uncertainty leading to decisions, which may involve unstated assumptions. Finally, we observed that participants could move from a certain state to an uncertain one. In our dataset we found participants transitioning to uncertain comprehension from perception (P→UC, 22 occurrences, 86% accuracy) or from comprehension (C→UC, 25 occurrences, 68% accuracy).

Table 3.6: Uncertianty patterns

Pattern	Freq.	Accu.*	Participants
{UP→UC}	8	100%	P1, P3, P5, P6, P9
{UC→UJ}	2	100%	P2, P5
{UC→UD}	2	100%	P1, P4
{UC→A}	8	75%	P1, P2, P3, P9, P11
{UC→A→D}	5	100%	P1, P3, P9, P11
{UC→Q}	7	100%	P2, P3, P4, P5, P7, P9
{UP→A}	5	60%	P1, P3
{UP→Q}	3	67%	P1, P3, P5
{UC→D}	9	67%	P1, P2, P5, P6, P8, P9, P11

*Excluding false positives

3.5.4 Patterns Showing Redundant SA Levels

In addition to the patterns we discussed so far, we identified several patterns that appear to show the analyst is working harder to reach a decision. This includes patterns with accuracy rates above 60%: (C→C→C→C), (C→C→D), (P→C→C→J), (P→C→C→D), and (P→C→P→C). These patterns appeared 21, 26, 3, 5, and 12 times, respectively. The patterns show that participants are working harder to comprehend (note the redundancy in the C code) and interpret meanings to make more informed decisions. The patterns and corresponding text indicate that, the more detailed and thorough participants comprehensions were, the better and clearer their future projections or decisions. This may explain why a participant needs more than one comprehension to reach the projection or decision levels. Moreover, there could be situations where complex security projections rely on multiple cues and multiple comprehensions. Moreover, the comprehension level is where the analysis and interpretation begins, and projecting or forming a decision relies heavily on how well the analyst understands the vulnerability. For example, when an analyst comprehends the meaning of a firewall on the network, they consider different factors, which could lead them to verbalize more than one comprehension. Consider the following example as P3 was trying to analyze the network diagram ND2 against the first security requirement from the requirements list provided:

> **{P}**your firewall**{/P}** **{C}**which is your first point of entry to both DMZ traffic and intranet site traffic and also to your users**{/C}** **{P}**has all of these on separate subnets**{/P}****{D}**the first rule here about stuff being unavailable [speaking about the requirement R1]comes down to whether this firewall is properly configured.{\D}

Participant P3 in the example above cannot reach a decision without comprehending two cues: 1) the firewall is the first point of entry to multiple network segments, and 2) the firewall places the segments on different subnets. Therefore, this decision is dependent on a composition of multiple comprehensions, which explains the redundancy in the above pattern.

3.5.5 The SA Path to Security Analysis

From our analysis results, we extended Endsley's SA model to account for uncertainty, the role of assumptions and participant inquiry that results from uncertainty. Endlsey defines the stages of SA as they occur in the human mind, but since we are annotating participant articulations of those stages based on their verbal statements, there will be no guarantee that we will observe patterns in the data that will exactly reflect the classic or reverse SA work-flow ({P→C→J→D}).

In the upcoming sections of this chapter, we will continue exploring security decision-making by analyzing the SA patterns found in our dataset.

3.6 The Security Expertise Effect

We are interested in investigating whether more experienced participants would exhibit better SA and, thus, be able to form more confident decisions. I have explained earlier in Chapter 2, the challenge of defining metrics to distinguish experts from novices. In the SA study, we choose to report multiple factors as background data of our participants taking into account the both industry and academic aspects of ones experience. In addition, we use the analysis that is based on the SA patterns discovered in the data to explore differences between participants performance and link that back to their background information to help distinguish between novices and experts. Ericsson suggests that focusing on experts performance and understanding how they utilize their knowledge could help develop improved learning and training approaches that can help increase the novices performance in a domain [43].

In this section, I will present participant's expertise and background data, followed by participants performance and ability to demonstrate an attacker model.

3.6.1 Participants Background and Expertise

Herein, we report our findings drawn from demographic data including participants background and experience. The participants experiences reported as remarks during their interview that we coded as {BG}. Next, we examine the role of expertise in forming more confident decisions. According to Endsley & Jones [41], an increase in experience may affect a participant's ability to project future consequences and, hence, may lead to more confident decisions.

Table 3.7 summarizes participant backgrounds: the participant number *P#* which is used consistently throughout this paper; *Years* is the number of years of industry experience, including internships; *Security Areas* are the general topics that best describe their industry experience; *Research Focus* are the topics that best describe their research experience; and *Degree* is their highest degree earned, or in progress; Among the total eleven, four participants (P1, P3, P4, P5) have extensive industry experience in security (4-15 years) with diverse concentrations.

The PhD. students in our sample had varying levels of experience, from a student who completed security courses, but who did not apply these lessons in practice beyond class projects, to students who had completed internships with a reputable company working on infrastructure security and log visualizations. P1, and P4 hold a Ph.D. in security and specialize in systems and infrastructure. These two PhDs and P5 have teaching experience in which they taught ad-

Table 3.7: Summary of participants background

P#	Industry		Research	Degree
	Years	Security Areas		
P1	5+	Network, systems, forensics and more.	Mobile computing, forensics, systems security	Ph.D.
P2	< 1	Security protocols, social networks.	Global cyber threat	Ph.D.*(5th yr)
P3	15+	Systems, Networks, programming, and more.	NA	B.S.
P4	5+	Systems, Networks, architecture, and more	Security for real-time critical systems & architecture	Ph.D.
P5	10+	Software Architecture, Secure Programming	Software Architecture	M.S.
P6	0	NA	Cyber & system security	Ph.D.*(4th yr)
P7	0	NA	Android security, malware, static analysis.	Ph.D.*(4th yr)
P8	1	Infrastructure security, log visualization	Security and Privacy	Ph.D.*(5th yr)
P9	0	NA	Security analysis, network traffic	Ph.D.*(2nd yr)
P10	0	NA	Anomaly Detection	Ph.D.*(1st yr)
P11	0	NA	Network traffic	Ph.D.*(4th yr)

*PhD student, followed by year of matriculation in parentheses

vanced security courses. The remaining seven participants were all PhD. students with research specialties in security.

Industry experience data in our sample, shows some correlation to participants performance. In our study, we observe that participants with more industry experience were able to make more assumptions compared to those with less experience. For example, participants with more than 5+ years of industry experience made an average of 7 assumptions, while participants with less than 5 years of experience made an average of 1 assumption. We coded statements with assumptions when the participant explicitly mentions that they are missing relevant details and that they have to assume or guess to complete their understanding.

Difference in artifacts presentation and notation could possibly affect situation awareness, and could help reveal different sub-domain expertise among experts. Certain portions of an artifact were likely more unclear than others, so we may only expect to see assumptions when participants encountered less clear portions of the artifact. The pattern (UC→A→D) was observed for experts P1, P3, and P9, when they analyzed the network artifact, and was observed for P11 when they analyzed the source code artifact. Participant P11 demonstrates advanced understanding when analyzing the source code artifact by reaching 24 decisions and this participant was the only participant to make 2 assumptions in that artifact. Recall from Section 3.5 above, assumptions were a sign of more experience in the area as the participant is making informed assumptions based on similar past situations. Hence, P11 demonstrated more expertise when analyzing the source code artifact.

3.6.2 Expertise Role in the Attacker Threat Model

Expert security analysts project future attack scenarios, and then decide how to mitigate these attacks. In security analysis, projection and decision are closely related, because security analysts may be trained to think like an attacker and have an attack model in mind [100, 123]. With an attack model in mind, the analyst decomposes a future attack scenario into multiple steps that exploit vulnerabilities. Under SA, we expect this decomposition to first appear as perceptions and comprehensions of the vulnerabilities, which then lead to the conclusion of projected exploitation, and finally a commensurate decision to mitigate the vulnerabilities. For example, Participant P3, notes:"what could I do since I am looking at this code to do bad stuff", which is their reflection on trying to walk through threat models that could be relevant to the code segment under review. P3 further stated:"it's critical if you're trying to design something secure to try and get into the mind of an attacker. If you can't think like an attacker, then you don't know how to defend against an attacker"

The anlaysis of the dataset measures how often security analysts employed the attacker perspective. In the study, five participants (P1, P2, P6, P8, P10) demonstrated the need to think like an attacker as demonstrated by the word *attack* in their statements while referring to how an intruder would act.

The study results show 45 instances of attack words used where participants demonstrate knowledge of an attack; out of which only 29 instances describe an application of the attacker model where participants describe how the attack is taking place. The remaining 16 instances out of the 45 statements include instances where participants are explaining attacks that they knew about from their background, but without relating that knowledge to the artifact being analyzed. For example, the word attack could show up in a `{BG}` statement without a relevant SA pattern. For our analysis, we are interested in the 29 instances where participants are actually *thinking like an attacker* by demonstrating an attack scenario. Table 3.8 shows our results from this analysis: the participant number (*P#*) who described the attack scenario; the frequency (*Freq.*) that the term attack appears, the security artifact (*Art.*); and the relevant *in-context patterns* associated with the word âĂŞ the SA code of the statement containing the attack word is highlighted in bolded text to show the position within the pattern. Each participant can exhibit multiple, separate instances of thinking like an attacker, which we separated by artifact and in-context pattern.

Among the 29 instances of the word *attack*, we observe that most instances (25/29) occurred in the projection stage of SA. In less than half of the instances (12/29), the projection was observed after the interviewer probed the participant to explain why they were perceiving, comprehending or projecting prior to describing the attack scenario (coded as `Pro`→`[*J*]`). Participants P2, P5, P7 are absent from Table 3.8, as they failed to demonstrate the attacker model.

Attack scenarios can be simple, meaning a single vulnerability is exploited to achieve an attacker's goal, or complex, meaning that multiple exploits are needed. In our results, we can observe and measure the complexity of attack scenarios as a series of different SA stages needed to demonstrate how an attack occurs within an artifact. For example, P9 projects a password brute force attack by looking at one item: requirement R7 on the list that reads: "Company X will require strong passwords (8 characters with complexity) for all user accounts." Based on the brute force projection, P9 decides that 8 characters alone are insufficient for a secure password policy. Alternatively, consider the attack pattern that P1 and P4 found in ND1: our entity analysis

Table 3.8: Participants use of the term attack

P#	Freq.	Art.	In-Context Pattern
P1	5	ND1	{P→C→C→Pro→[⋆J⋆]}
		ND2	{P→C→[⋆J⋆]→C}
		ND2	{D→D→Pro→C→C→[⋆J⋆]→C→C}
		ND2	{U→J→Pro→[⋆UJ⋆]→Pro→J}
		ND2	{Pro→UJ→Pro→[⋆J⋆]}
P3	3	ND1	{P→C→D→Pro→C→D→C→D→J→D→D→Pro→[⋆J⋆]}
		ND2	{D→J→Pro→J→Pro→[⋆J⋆]→Pro→[⋆J⋆]→C}
P4	2	ND2	{D→C→C→[⋆J⋆]}
		SC	{D→C→Pro→[⋆J⋆]→Pro→C→C→P→C}
P6	4	SC	{[⋆J⋆]→D→[⋆J⋆]→J→C→C→J→Pro→C→C→Pro→P→J}
		SC	{C→C→[⋆J⋆]}
		SC	{D→[⋆J⋆]→D→D→J→Pro→C→P→J}
P8	3	SC	{C→Pro→[⋆J⋆]→Pro→J→D}
		DFD	{C→C→D→[⋆J⋆]→Pro}
		DFD	{C→J→[⋆J⋆]→C→C}
P9	1	SC	{Pro→[⋆J⋆]→J→D→UP→D}
P10	7	SC	{D→Pro→[⋆J⋆]→J→[⋆J⋆]→D→C}
		SC	{[⋆J⋆]→Pro→Pro→J→[⋆J⋆]→D}
		SC	{[⋆J⋆]→J→Pro→[⋆J⋆]→J→[⋆J⋆]}
P11	4	SC	{P→[⋆J⋆]→J→[⋆D⋆]}
		SC	{C→C→[⋆D⋆]}
		ND2	{D→[⋆C⋆]→C→UC}

shows that in order to demonstrate the possible attack on the insecure network, both participants where analyzing multiple items in the ND1 diagram: allowed inbound ports on the router, the web server, the DNS controller, and the mail server. P1 further explained:

> **{J}** From an attacker that has no other entry point he is going to look at these three things [speaking about the 3 allowed inbound ports shown on the router], and if they didn't have any DNS server inside, there will be no reason to have port 53 open **{/J}**

Using SA patterns, we can compare participants analysis when looking at the same entity (see our explanation of entity analysis earlier in this chapter). For example, In Table 3.8, participant P1 presents the pattern (P→C→J→C) in ND2 by first perceiving server names (entity code: NAME), such as Alpha, Lima, Bravo, etc. Participant P1 comprehends the server naming scheme and subsequently projects that an attacker discovering these names alone cannot tell the role or function of the servers. Based on our entity analysis that links SA codes to these servers across participants, we found that participant P11 perceived the same naming scheme in their analysis (Q→P→C→UC→C), but they were unable to project based on the meaning of the scheme and thus were unable to see the attack scenario. Instead, P11 asks questions, experiences uncertain comprehension due to the meaning of the naming scheme and whether the scheme has any relevance to network security. Unlike P1, participant P11 stops at comprehension and does not proceed to projection or decisions. This is an example of how the same cue could be interpreted differently by experts of different expertise levels.

Our SA attack model shows how we can use SA to detect a certain expertise skill: *thinking like an attacker*. A conclusion that is based on the background data alone that is shown in Table 3.7 above, might indicate that participants P1, P3, P4, and P5 have more expertise with respect to these artifacts compared to the remaining participants in the table who could be treated as novices. This classification, which could be referred to as *industry classification*, is based on participants clearly combining years of practical industry experience along with academic degrees. However, this classification does not take into account the personal skills that a security analyst might acquire through their job or academic learning. Our attack threat model, on the other hand, help address this limitation by identifying the experts who demonstrate who can *think like an attacker*. Table 3.8 shows that in addition to P1, P3, P4, who are already identified experts based on their industry experience, P6, P8, P9, P10 P11 can also demonstrate the skill of thinking like an attacker.

Going back to Table 3.8, we observe that except for P11's ND2 pattern, all participants had their *attack* keyword appearing in a projection or a decision statement, which resonates with the definition of our projection statements where a future attack is described, and our decision statements where mitigations to an attack is explained. By looking into the details of P11's pattern (D→C→C→UC), we observe how the participant is stuck at the comprehension level where they demonstrate a level of uncertainty.

3.6.3 Expertise Role in Security Requirements Mapping

After presenting the diagram ND2 to the participants, we presented the security requirements checklist. We observed individual differences among experts and novices when assessing a single requirement and linking it to the diagram entities. In general, 5 out of 7 participants who were presented with ND2 exhibited an improved ability to discuss items in the checklist that they previously missed, as compared to the two modes above. Analysts made an effort to connect each requirement to entities in the diagram. Table 3.9 below shows the results of mapping requirements to entities in the diagram by the 5 participants who were presented with ND2 and were successful in the mapping exercise. Participant P2, and P5 are absent from the table as they have stated that they could not see how to do the mapping. None of the participants shown in Table 3.9 managed to map requirement R3 (shown in Appendix A.2), which is about âĂIJhardeningâĂİ the network. Participant P4 stated that the rule makes no sense, as it cannot be qualified nor quantified. Participant P3 commented: "that's not uncommon for compliance to do that, to just state in very general terms a requirement, and then it's a little loose interpretation as to whether or not you've met that compliance or not." Highlighted cells in the table indicate that participants stated that dependencies exist among the highlighted requirement. Participant P1 found the requirements R11 and R12 to be related. Participants P1, P3, and P11 agreed that R9 and R10 are related, but P11 failed to point out the entities on the diagram that map to the requirements.

Mapping the requirements-entity matching data in Table 3.9 to experience and background data in Table 3.7, it can be observed that P1, P3, P4 who has more industry experience then P9 and P11, were able to match more requirements on the list.

Using entity analysis, participants' responses are compared across entities in diagram ND2. The analysis results indicate that the requirements list could help both experts and novices: the

Table 3.9: Participants requirements mapping to entities in ND2

R#*	P1	P3	P4	P9	P11
R1 *DMZ*	Firewall-1	Firewall-1	Firewall-1	DMZ	
R2 *Proxy*	Proxy (Squid)	Firewall-1, DNS-1	Proxy (Squid)		
R3 *Harden services*					
R4 *Web filtering*	Proxy (Squid)		Proxy (Squid)		Snort1, Snort2, ArpWatch
R5 *Windows group policy*		Windows DC	Firewall-1, Firewall-2, Exchange Mail Server		
R6 *Electronic mail relay and filters*	Firewall-1, Firewall-2	Exchange Mail Server	Exchange Mail Server, DMZ Mail Server, Firewall-1	Exchange Mail Server	
R7 *Strong passwords*		Windows DC	Exchange Mail Server		
R8, *Network segments*	Firewall-2		Firewall-1, Firewall-2		
R9 *Logging*	Syslog	Syslog	Nagios, ArpWatch	Syslog	
R10 *Time synch*	Windows NTP	Windows NTP	Windows NTP		
R11 *IDS*	Snort1, Snort2, ArpWatch	Snort1, Snort2	Snort1, Snort2, ArpWatch		
R12 *Split DNS*	DNS-1, DNS-2, DMZ	DNS-1, DNS-2, DMZ	DNS-1, DNS-2, Firewall-1, Firewall-2	DNS-1, DNS-2	
R13 *Packet Sniffers*		ArpWatch	Snort1, Snort2, ArpWatch		
R14 *Centralized System Monitoring*	WinMRTG, Nagios	Syslog	WinMRTG, Nagios		
R15 *Isolated admin network*		Firewall-2			

experts attention was focused towards a specific security component and help them reach better-informed decisions, and the novices became aware of a requirement and/or its security justification. Consider requirement R12 that requires a split DNS policy: expert participants P1, P3, P4, and P9 were able to map requirement R12 to the split DNS servers shown on the diagram and to state that the network satisfies the requirement, and they were also able to explain why such requirement is important from a security standpoint. Participants P1, P3, P4, P9 demonstrated the patterns: (P→P→UP→P→UP→D),(P→Q→Pro→D→J→J→J→A→J), (Q→C→C→C→J→J),(C→P →J→D→Pro→D→UC→C→A→C→C→J→C→D) respectively.

By investigating why P3 and P9 had longer patterns, it is found that they were demonstrating an attacker's attempt against the DNS server and how the split DNS increases the difficulty for attackers to break into the system. Towards the middle of participant P9's pattern, the participant exhibits uncertainty about why this requirement in needed for the system's security and thus they made an assumption in order better comprehend and project before reaching their final decision. Participant P11 was able to state that the requirement R12 is satisfied based on the diagram, but was unclear why a split DNS policy is needed. This is an example of how introducing structure to security analysis, could help analysts become aware of essential security requirements.

Table 3.9 suggests that participant P4 provided more mitigations among all participants. We found that P4 employed a matrix-based analysis approach by drawing a table on a blank piece of paper, listing the requirements numbers, and documenting how the requirement could be satisfied given the information shown on the diagram. During the interview process, P4 exhibited more depth in their analysis and had greater confidence as evidenced by the absence of uncertainty patterns in his analysis of the ND2 and the requirements mapping. The word depth is used here because P4 was able to refine requirements into specification levels and write down system specification and software configurations that are essential to satisfy the requirement, and this observation did not occur with any of the other participants.

3.7 Summary Observations

The three categories of artifact - source code, data flow diagram and network diagrams were chosen to vary specificity in system design and operation in order to surface variations in analyst performance. For example, the source code artifact examines participants responses when presented with details at a code-level, while the data-flow diagram is a high-level representation of an architecture that can introduce ambiguity. We now discuss those variations based on our SA results.

3.7.1 The Source Code (SC)

Eight participants were presented with the source code artifact, of whom seven agreed to analyze it (see Table 3.2). Six out of the seven participants identified at least two major concerns: the risk of SQL injection attack and of unencrypted user data. The remaining participant (P10) could not spot the SQL injection vulnerability although he was reminded by the interviewer more than once to look at the artifact and provide any possible security concerns they might have, or if they have further comments.

The level of analysis and the proposed solutions varied in detail between the participants. While some were able to explain what languages to use and what libraries to call, some found it sufficient to make a general statement that "there are more secure measures that exist" and good programmers should know about these secure measures. To investigate this more, we looked at the coded statements of participants; and compared participant P10 to others who were able to spot the vulnerabilities. For this specific source code artifact, P10 had only 4 perceptions compared to 12, 9, 13 perceptions for P6, P8, P11 respectively. However, P10 had 30 comprehension statements, which is the same as P11 who had more perceptions. When we read some of the statements, we found that P10 spent more time comprehending the 4 perceptions and thus deviating away from the intended attack to demonstrate other types of attacks that could occur, such as phishing. Although Table 3.8 indicates that P10 can actually demonstrate thinking like an attacker, results from our entity analysis showed that P10 was demonstrating possible attacks other than the SQL injection attack, which is the main weakness of the scenario presented in the artifact.

3.7.2 The Data Flow Diagram (DFD)

We found 4/7 occurrences of the (`{UC→Ask}`) pattern in the data flow diagram (DFD), as participants have reported being confused about the chronological order of diagram entities. In addition, the DFD shows higher comprehension uncertainty (49 `{UC}` statements compared to 24 `UC` statements for source code). From the participant responses, we infer that all seven participants agree that the diagram lacks specific details needed for analysis. This result was expected when we designed the artifact: we deliberately created the diagram with less details to assess how ambiguity could affect the results. In our data, we observe two participants (P2, P5) responding differently to the ambiguity although they have perceived the same cue. Participant P2 states that they do not understand the role of the digital signature shown on the diagram (`{UC}`). In contrast, the participant P5 responds to the same entity by challenging the uncertainty with a perception and scaffolding their analysis with an assumption to reach a decision:

```
    {UC}Okay. So presumably I'm not sending my digital signature in the
clear. It's an encrypted session, right?{/UC}{P} But again that doesn't
really show that here{/P} {A}so if we assume that's an encrypted session and
 that I am not sharing my digital signature with somebody{/A} {D}then this
is trusted{/D} {J}but if my machine's been compromised and someone has my
digital signature they could potentially publish things as me, right?{/J}
```

3.7.3 The Network Diagrams (ND1 and ND2)

The network artifacts illustrate how expertise areas and job role affect decision-making. Recall from Section 3.6.2 how participant P1, and P11 reacted differently to the same perceived cue of the server-naming scheme. When we matched participant background information from Table 3.7 with their decision-making patterns, we observed that a job role, such as P1's hands-on experience in networking, might improve the participant's comprehension of cues and lead them to better decision-making.

Contrary to the SC artifact, where participants look at a code snippet showing one distinct vulnerability: the SQL injection, network diagrams describe a composition of IT components (servers, routers, etc.) in which each component may have its own vulnerabilities. Thus, participants must view these vulnerabilities together to reach certain categories of decision. These interactions can be overwhelming for participants, if no structure is imposed on how they conduct their analysis. We observed three modes of security analysis: unstructured, semi-structured and structured, which we now discuss.

The Unstructured Mode

Participants were provided the least amount of structure when they were presented with the insecure network diagram (ND1) that had minimal cues, text and legends. Every participant began their analysis with a different cue or entity, and each participant arrived at their own concerns and threat models. Table 3.8 shows that P1 and P3 demonstrated an attacker threat for ND1, but the entity analysis shows that the two participants were looking at different entities and demonstrating different attackers. Participant P1 began their analysis from the firewall and its possible rules for open ports and participant P3 was more focused on the insecure layout of the DNS, e-mail and web servers. Both participants reached similar mitigation techniques, such as using a DMZ, and network segmentation in order to reduce the attack surface.

The Semi-Structured Mode

The diagram ND2 has more legends and cues. The icons are distinguished by type of entity and the text and legends provide more detail, such as IP address, server name, OS type, etc. When participants analyzed ND2, they showed more structured analysis than they did with ND1. Contrary to ND1, all participants here, novices and experts, started at the same cue: network segmentation. They recognized the network segmentation of users, administration, management and DMZ, and explained the security advantages of such designs. The diagram in ND2 clearly shows the segmentation using legends and color-codes that the network segmentation becomes very obvious. However, some participants weren't able to explain by the diagram alone some of the network design decisions such as the reason for having two separate DNS servers one of which is present in the DMZ. We will show next how structured analysis helped address this problem.

The Structured Mode

After presenting the diagram ND2 to the participants, we presented the security requirements list. We observed individual differences among experts and novices when assessing a certain requirement and linking it to the diagram entities, but in general participants had more insights compared to the two modes above. However, we observed that participants P1, P3, P4, who organize their thoughts and follow a more structured approach in their analysis of the requirements list, tend to provide more insights and recognize entities that affect security analysis that they did not mention before looking at the requirements list. Using our entity analysis, we compared participants' responses across entities in diagram ND2. Our analysis results indicate that the requirements list could help both experts and novices: the experts' attention was focused towards a specific security component and help them reach better-informed decisions, and the novices became aware of a requirement and/or its security justification. Consider requirement R12 that requires a split DNS policy: expert participants P1, P3, P4, and P9 were able to map requirement R12 to the split DNS servers shown on the diagram and to state that the network satisfies the requirement, and they were also able to explain why such requirement is important from a security standpoint. Participants P1, P3, P4, P9 demonstrated the patterns: ({P→P→UP→P→UP →D}),({P→Q→Pro→D→J→J→J→A→J}), ({Q→C→C→C→J→J}),({C→P→J→D→Pro→D→UC→C →A→C→C→J→C→D}) respectively. We investigated why P3 and P9 had longer patterns, and we found that they were demonstrating an attacker's attempt against the DNS server and how the split DNS increases the difficulty for attackers to break into the system. Towards the middle of participant P9's pattern, the participant exhibits uncertainty about why this requirement in needed for the system's security and thus they made an assumption in order better comprehend and project before reaching their final decision. Participant P11 was able to state that the requirement R12 is satisfied based on the diagram, but was unclear why a split DNS policy is needed. This is an example of how introducing structure to security analysis, could help analysts become aware of essential security requirements.

Participant P4 took an alternative and more highly structured approach to analysis by drawing a table on a blank piece of paper, listing the requirements numbers, and documenting how the requirement could be satisfied given the information shown on the diagram. During the interview process, P4 has shows more depth when analyzing the results and had confidence in their security analysis. We use the word depth here because P4 was able to refine requirements into specification levels and write down system specification that are essential to satisfy the requirement, and this observation did not occur with any of the other participants.

3.8 Threats to Validity

In this section, we address threats to construct, external and internal validity.

Construct validity is whether measures actually measure the construct of interest [131]. In our study, the construct of interest is SA, which is comprised of the four levels previously mentioned. One threat to construct validity is the definitions of the codes for each level in the coding frame are ambiguous and not mutually exclusive, such that the codes are inaccurately applied to the wrong statements (i.e., the perception code, if misapplied, may not be measuring instances of perception). To address this threat, we had two researchers meet to first discuss the coding frame before applying it to the dataset, after which we identified points of disagreement and reconciled these differences in a subsequent meeting. Recall from Section 3.4, we computed the inter-rater reliability statistic Cohen's Kappa that showed a moderate to high agreement. Unfortunately, we cannot know when participants are making implicit or unstated assumptions before reaching their decisions. Personality may be a co-factor that can effect whether or not participants make assumptions, since assumption making may be related to over-confidence.

External validity refers to the extent to which the results of this study can be generalized to other situations [131]. This study is based on grounded analysis, which limits generalizations beyond the data set. While some might argue that our findings are thus too limited, we identified several prospects for future research. This includes whether we can transfer expert assumptions to novices to facilitate transitioning novices from comprehension to decision-making, or how can we improve perception to reduce uncertainty. These questions can be further examined in future generalizable, controlled experiments. In this section, we discuss our results in the context improving the evaluation of security notation in artifacts used in security analysis, and provide suggestions moving forward explaining how hour method could be adapted to improve the design of security training.

Internal validity refers to whether the conclusions drawn from the data are valid [131]. Based on our coding of the data, we inferred several decision-making patterns in the data set that we report in Section 3.5. The completeness of the data threatens internal validity, because participants have unspoken perceptions, comprehension, etc. To address this threat, we employed probing questions to prompt participants to make explicit their SA levels, and we checked our observed patterns for accuracy across the dataset, i.e., how many instances of the pattern were consistent with our definition of the pattern. This process led us to discover the reverse SA patterns (see Section 3.5 above), which corresponds to differences between western deductive and eastern inductive reasoning styles previously studied in psychology [6, 23, 99].

3.9 Discussion and Future Work

In this section, I will discuss our results in the context improving the evaluation of security notation in artifacts used in security analysis, and provide suggestions moving forward explaining how our SA method could be adapted to improve the design of security training.

3.9.1 Identifying Effective Cues

Throughout this chapter, we discussed how certain analysts were able to perceive certain cues in the artifacts, comprehend them, and then, project and decide on mitigations, accordingly. However, we also showed cases where novice analysts were facing uncertainty during comprehension about a cue, e.g., trying to make sense of its meaning or its possible consequences. In Section 3.7.1, we showed how one analyst, P10, did not even reach perception; P10 failed to perceive the cue that leads analysts to project the SQL injection attack.

In addition to measuring where analysts struggled to move past perception and comprehension, we assessed the effect of improving notations and visual cues by comparing performance between the two network artifacts, ND1 and ND2, and also by comparing the analysis results of the DFD artifact. Recall from Table 3.8 how only one participant P8, was able to demonstrate an attack on the diagram. In Section 3.7.2 we showed how participants exhibited increased uncertainty analyzing the DFD artifact, which indicates how notational elements (or lack thereof) introduce ambiguity, which has a negative impact on the analysis.

These observations lead to the following question: *How can we avoid situations where experts fail to perceive or comprehend a cue?* The SA methodology that we applied helps surface the cues that likely to need support. While experts may have little difficult reaching projection and decision, novices may need additional information to aid them in reaching these higher levels. In addition to identifying the cues, comparing the results could help find ways to redesign the artifacts in a way that makes the cues either more explicit (improve perception) or more meaningful (improve comprehension). We even envision an adaptive security analysis system that can adapt to the training needs of a security trainee based on their perception and comprehension of cues. If a trainee fails to identify a cue, then the system could provide deeper training with further cues in order to help the trainee perceive vulnerabilities, comprehend its risk, project the impact, and decide on the proper mitigation.

In addition, deciding the appropriate cues could help inform future security experimental designs. For example, consider a study that tests how security analysts evaluate a certain system artifact for threats. In order to draw correct conclusions from the experiment, first we need to evaluate the cues used in the experiment materials (online application, paper, etc.) during a pilot study. Cues can be selected that participants perceive and understand well, and others can be improved if they are misleading or ambiguous.

3.9.2 Structured Analysis Trade-offs

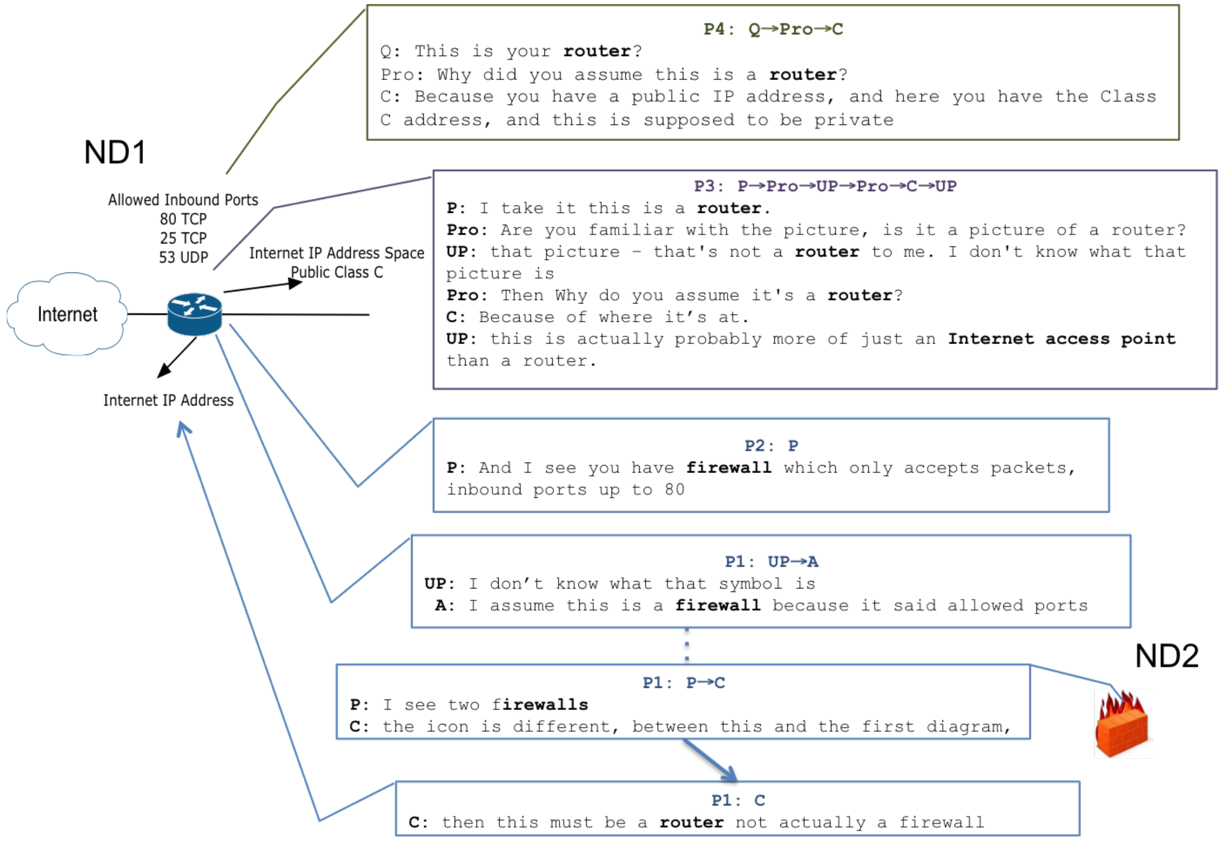
It is arguable whether or not to provide structured approaches to security analysis. Although our findings in this work are in favor of structured analysis, we think that the decision of favoring structured vs. unstructured analysis is based on realizing the trade-offs between the two approaches, and future research examining those trade-offs is beneficial. The structured approach improved the experts' security analysis of ND2. Only after going through the requirements list, participants P1, and P2, P3, P4 noticed the split DNS design in ND2, which was an improvement over the insecure diagram shown in ND1, but they did not point it out by looking at the diagram alone.

3.9.3 Ambiguity and Resolution

We intentionally chose the ND1 with minimal cues and information displayed to study the role of ambiguity in decision-making. Consequently, participants interpreted a router icon differently, as a router or firewall. Figure 3.2 shows the different interpretations of the same entity by four participants, including their statements in order of articulation coded by the SA method. When the notation was improved in ND2, we observed a positive effect on P1 for example. After later seeing the firewall icon in diagram ND2, participant P1 returned to ND1 to correct their prior interpretation to conclude that the ND1 icon was a router.

Participants could not comprehend effectively if they did not perceive appropriate cues that lead to a comprehension, and that could explain having uncertainty patterns appear in our dataset, which leads an expert to transition to an uncertainty stage as shown in Figure 3.2. When analyzing the DFD artifact, for example, one participant attempted to think of all possible interpretations given the absence of specific details from the diagram. In the excerpt below, we show how participant P3 assumed that encryption existed:

> **{UC}**that doesn’t really show that here [speaking about encryption session for sending the digital signature] **{/UC}**, **{A}**so if we assume that’s



P4: Q→Pro→C
Q: This is your **router**?
Pro: Why did you assume this is a **router**?
C: Because you have a public IP address, and here you have the Class C address, and this is supposed to be private

P3: P→Pro→UP→Pro→C→UP
P: I take it this is a **router**.
Pro: Are you familiar with the picture, is it a picture of a router?
UP: that picture - that's not a **router** to me. I don't know what that picture is
Pro: Then Why do you assume it's a **router**?
C: Because of where it’s at.
UP: this is actually probably more of just an **Internet access point** than a router.

P2: P
P: And I see you have **firewall** which only accepts packets, inbound ports up to 80

P1: UP→A
UP: I don’t know what that symbol is
A: I assume this is a **firewall** because it said allowed ports

P1: P→C
P: I see two f**irewalls**
C: the icon is different, between this and the first diagram,

P1: C
C: then this must be a **router** not actually a firewall

Figure 3.2: Participants’ perceptions of the router icon in diagram ND1

an encrypted session and that I am not sharing my digital signature with somebody{/A} {D}then this is trusted{/D}

In a few cases of uncertainty, assumptions helped participants resolve the ambiguity and reach their decisions. Those assumptions were not arbitrary; they were based on former experience and best practices adopted for network security that experts had been exposed to.

The following coded excerpt that was taken from participant P1 and illustrates such an assumption:

{UP}I don’t see an NTP server on this network{/UP} {C}but I know that Windows Domain Controller can act as NTP{/C}, {A}so I am going to assume that when they install it they’ll probably leave that box checked because it ’s a default option{/A}. {D}I think that is probably happening here{/UD}

The above assumption is an example of a trust assumption that were applied to security requirements by Haley et al. [63]. Trust assumptions describe desired behaviors and may be outside the control of the system designer. Based on the background-coded data BG (see Table 3.1 for a definition of this code), participant P1 has extensive hands-on experience in network security, which could explain why P1 was comfortable making assumptions about the system. The example above shows an interesting pattern ({UP→C→A→UD}). Although we did not observe the exact same pattern with other participants, we were able to observe the latter half of the pattern:

`{A→UD}` as it occurred once for P5 and P11, and twice for P3 and P9. These participants reported significant experience in network security, so one would expect them to be more confident in reaching certain decisions with respect to network artifacts. However, we must not ignore the personality effect: an expert may hesitate to make confidant decisions based on assumptions, so they express a level of uncertainty with their decision to be more cautious. As explained by the "Dunning-Kruger effect", more competent participants may have a cognitive bias towards underestimating their abilities [77].

Trust assumption reported by Haley et al. [63] help restrict the domain by narrowing the attention span of the analyst. In SA, a narrowed focus is beneficial for projection, but it can also lock-in the analyst and prevents them from perceiving alarming cues in the environment [41]. Our work could be extended by distinguishing which assumptions are trust assumptions to distinguish the volatility of decisions that depend on assumptions about actors that are outside the system boundary. If those trust assumptions turn out to be untrue, then the security analysis that depends upon those assumptions should be revisited for possible inconsistencies

While our dataset is small in the number of participants, we did observe that experts were more likely to use assumptions to control uncertainty and to reach a decision. In future experiments, we could test if assumptions could provide another metric to distinguish between novices and experts. Being able to distinguish users based on expertise level could have an important impact on designing intelligent and interactive tools to help novice analysts cover more security scenarios in a problem description or specification.

3.10 Conclusions of the "SA Study"

The SA study described throughout this chapter, has shown a new empirical research approach to assess security expertise and decision-making. So far, we have shown a systematic method to apply the Situation Awareness (SA) framework to distinguish security experts effective analysis based on their differences in recognizing attack threat models. The results were presented to show traces across the SA levels in the form of patterns that could be used to distinguish experts from novices. We believe that other researchers can use insights from this methodology and adapt it to evaluate their technical solutions to security analysis by improving notation, presentation, training materials, and most importantly understanding how those solutions improve novice decision making in comparison to experts.

In short, the SA study highlights the following [69]:

Security requirements exist in composition Deciding on security requirements that mitigate threats relies on: the context of the attack and the composition of requirements. Attack patterns found in the data confirm this finding as participants need to understand and comprehend how perceived cues interacts forming a context where a future attack can occur. This thorough understanding can lead to deciding about proper attack mitigations

Security decision-making involves uncertainty Uncertainty in this context means missing information that is essential to the get the full picture, or ambiguity in presentation where analysts could have different interpretation of the same item. Experts differ in handling uncertainty based on their past experiences, but even when highly confident, experts tend to quantify security with "it depends," which means: 1) that security decisions rely on the context, and 2) we need better linguistic expressions that accommodates the uncertainty present in security decisions.

Security expertise is broad and stove-piped The results of the SA study suggest that experts vary in their domains of security expertise and that variation impacts their security analysis. Participants P11 for example, performed better when analyzing the SC artifact, as they were able to detect and mitigate the SQL injection vulnerability, but performed poorly trying to map ND2's entities to the requirements list. We need to measure security by evaluating domains together and independently with respect to analyst expertise.

These findings from the SA study that listed above, had led me to design user-experiments that instruments scenarios to present security requirements to experts. This scenario-based approach accounts for the effect of context and helps measure the effect of composed requirements in a scenario on the overall security assessment of that scenario. When instrumenting the user-study, I collected demographic information and used a security knowledge test to examine the effect of an expert's background on security assessment. More details to follow in the upcoming chapters.

Chapter 4

Establishing the scientific validity of Ad-hoc Security Measures

To ask experts about security decision-making, we must first decide on the measure to be used for security. The interviews with experts that I described in Chapter 3 show that experts were hesitant to describe a feature or a component as secure or insecure, and they preferred to say “it depends”. Decision scientists investigate which constructs can measure human judgment, and this requires researchers to design and validate new scales.

Since I am asking security experts to provide security assessments in a survey, I found it necessary to communicate security levels using a labeled scale. There has been a number of efforts to represent security on a scale, for example, the National Institute of Standards and Technology (NIST) special publication 800-30 recommends three levels to represent security risk: low, medium and high. In our work, we created new labels to measure security on a scale of adequacy.

In this chapter, I explain how to investigate a new scale. We created a new scale to measure a construct of security adequacy. We found this to be necessary because there are no existing, empirically valid scales to measure this construct in the security requirements context. This type of scale is usually described in the psychometric literature as an *ad hoc scale* due to the lack of valid or reliable scales [49]. Creating ad hoc scales requires evaluation to examine the reliability of the scales rather than relying on the face validity, alone [49]. In contrast to construct validity, face validity is subjective: if a test or measure “looks like” it will measure what it is supposed measure, then it has face validity [91].

Below, I will describe our approach to selecting the adequacy scales to be used to describe security requirements, and the user studies conducted for the empirical evaluation of the scales.

Excerpts from this work were previously published as: H. Hibshi, T. Breaux, and S. B. Broomell, *Assessment of Risk Perception in Security Requirements Composition*, 2015 IEEE 23rd International Requirements Engineering Conference (RE), pp. 146-155, Aug. 2015; H. Hibshi, T. D. Breaux, and C. Wagner, *Improving security requirements adequacy: an interval type 2 fuzzy logic security assessment system*, in 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1âĂŞ8.

4.1 Approach

In this section, I will describe the details of our research methodology including, how the initial scale labels were selected and evaluated.

4.1.1 Selecting the Initial Labels

At this stage, my goal was to answer the following question:

When given a scale, what kind of word labels best describe an increase or a decrease in security?

The linguistic labels that we are interested in should describe a security assessment scale for survey participants who are evaluating security requirements in a scenario. The choice of such labels is context dependent by application, and relies on the background knowledge supported by empirical evaluation using experiments [88, 90]. We conducted a focus group of five researchers in our lab to discuss the initial set of labels that we are considering for security assessment. We used the context of a concrete scenario to be able assess labels that mostly came from prior research in fuzzy logic [88, 90], and in NIST special publications. An example of some of the labels used in prior research include: low, medium, high, and moderate. The focus group discussion concluded that these labels are not very well suited to describe security because it is unclear how to define low security vs. high security?

In the focus group discussion, members explained that experts analyze threats in a security scenario as they are concerned with risks involved with the threat. Security requirements are intended to mitigate the threats and decrease the risk. With the goal of mitigating threats, security requirements can be described as: inadequate, adequate, or excessive, because security requirements are often viewed by companies as cost requirements, meaning the value is not so obvious to achieve primary system goals and stakeholders often have difficulties seeing the benefits. Furthermore, excessive security has negative financial and usability effects, while adequate security is what an organization might settle for. Hence, we developed three labels to describe security adequacy: inadequate, adequate, and excessive. This understanding of security requirements enables us to describe requirements in terms of "adequacy" of a security requirement to mitigate a threat.

Hence, the focus group proposed with choosing the three labels: inadequate, adequate, and excessive. Next, I will describe the experimental evaluation of the labels.

4.1.2 Experimental Evaluation of the Labels

For the purposes of my research, I evaluated adequacy scales to be used in a security context, and asked experts to assign intervals to word labels. My purpose was to investigate how many labels are needed on a scale to represent security adequacy? Answering this question involves obtaining numerical intervals for the words to make sure that we select the minimum number of word labels that covers an entire given scale (e.g. from 1-10).

We obtained an 18-word data set using a standard English dictionary by looking for synonyms of the words: inadequate, adequate and excessive. The 18-word data set includes the original three labels and the additional synonyms. We presented the words from the dataset to security experts using a survey in which we asked participants to represent these words using as an interval on a predefined range. This approach is commonly accepted and adopted by the fuzzy logic research community [88, 90]

Participants were asked to specify the start and end points of intervals of the 18 words using the text template shown below, replacing *Adequate* with each of the other 17 words. The word order in the survey was randomized. Since human perception of adequacy can vary by scenario and context, we include a security scenario to add context to each word as follows:

```
     A security expert was asked to rate a security scenario with regards
to mitigating the Man-in-the-Middle threat.
     The expert would give an overall security rating using a linguistic
term.
     In the next sections of this survey, we will present 18 linguistic
terms describing the overall security of a scenario. We would like you to
mark an interval between 1-10 that represents each term.
     Note: Intervals for different terms can overlap.
```

For each word (e.g., "adequate"), participants were asked:

```
     Imagine "Adequate" represented by an interval on a range from 1-10.
Where would you indicate the start and end of an "Adequate" security rating?
```

Participants Demographics

At the end of the survey, we ask participants to provide demographic information where they answer questions about job experience and security training, in addition to their age, gender, and income level. It is recommended to place background and demographic questions at the end of surveys to increase participants response rate, because research has shown that placing these questions at the beginning may detract the participants attention from the intended survey topic [105].

Expert Recruitment

Participants were recruited by sending out email invitations to mailing lists of security research groups at Carnegie Mellon University. We anticipated that the survey will take a participant between 15 and 30 minutes (based on a pilot test with 4 participants), so we offered each participant a $10 Amazon gift card.

4.2 Results

Intervals were collected from 38 security experts who consist of 74% males, 18% females, and 8% unreported. For each word, we calculated the mean and standard deviation for the interval end points that we collected from participants. The results show that the three words, inadequate, adequate, and excessive are sufficient to be used as fuzzy sets covering an interval from 1-10. Figure 4.1 shows all the labels and their coverage over the 1-10 interval. The solid region represents the interval between the mean values of the start and end points collected from the experts. The shaded region on each side of the solid region represents the standard deviation for that point, which represents the *uncertainty* surrounding the mean value. It is only possible to cover the entire region from 1-10 using only three labels (inadequate, adequate, and excessive) due to the uncertainty, which yields overlapping intervals for the three words. I will explain more in Chapter 6 about how type-2 fuzzy sets can represent a linguistic label, while maintaining the uncertainty that is present in the data.

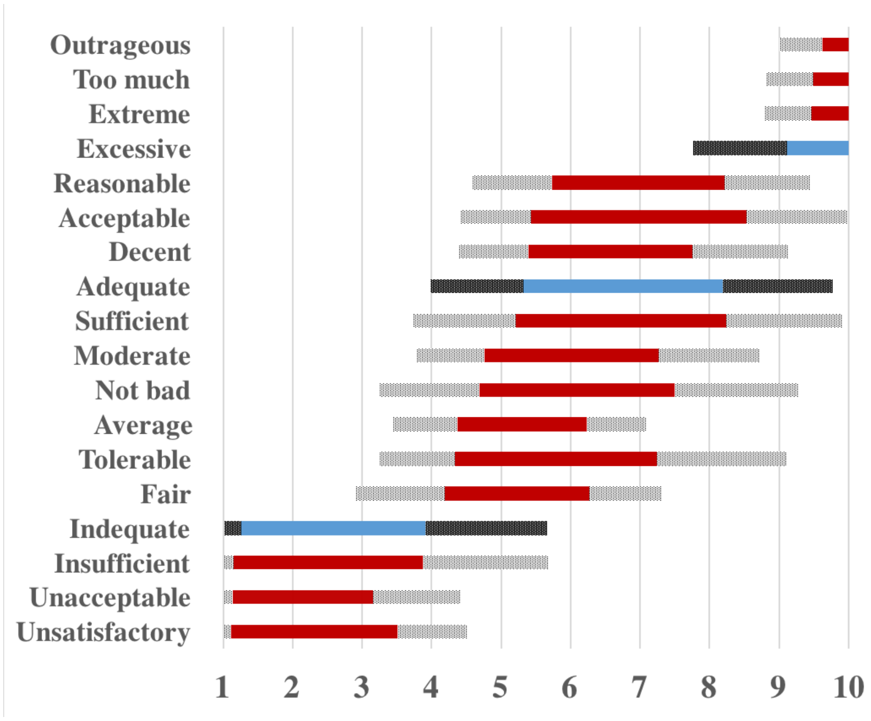


Figure 4.1: The fuzzy sets with the start and end means, and standard deviation

4.3 Threats to Validity

In this section, I will discuss how we addressed threats to construct, external, and internal validity.

Construct validity is whether a measure actually measures the construct of interest [131]. In the adequacy labels study that we discussed above, one threat to validity could be that a participant might think of the words in a context different than security requirements. For example, we have examined in a separate online study, how participants could rank the adequacy labels differently in four different contexts such as: describing meal portions, waiting time for a bus, distance to parking lot, and amount of privacy protections against government surveillance [65]. To reduce the effect of such threat, we provided a security context with each word, so participants would provide the intervals while thinking about security requirements.

External validity refers to the extent to which the results of this study can be generalized to the population and other situations [131]. Our target population is security experts and we targeted participants by recruiting from security mailing lists that include professors, post-doctoral researchers, and graduate level students. One possible sample bias is that our sample was drawn from two U.S. Universities. We also used a security scenario in the study, so it is unknown whether the intervals for the labels generalize to other contexts outside of security.

Internal validity refers to whether the conclusions drawn from the data are valid [131]. One possible threat to validity in the labels study is the ordering effect of the words that could bias the outcome. For example, the interpretation of *Decent* might shift toward 10, if it followed *Inadequate* instead of *Not bad*.To address this threat, we randomized the order in which the words were displayed to different participants.

4.4 Discussion and Conclusions

The results of this word-study served two purposes for my research. First, I will explain in Chapter 5 how I applied the labels: inadequate, adequate, and excessive as anchor points on a semantic-scale used to collect experts ratings of security requirements. Next, I will show in Chapter 6 how I used the intervals provided in this *adequacy labels study* to build type-2 fuzzy sets that will be used in a security assessment system.

The number of anchor points on a scale carries an important consideration as it affects that user-study design, and the modeling of the word using fuzzy sets. As the number of anchor points increases, the usability and user-friendliness of the scale could drop. The more anchors participants need to read and comprehend, the more there is an increase in their time to answer the questions on the survey. With regards to using fuzzy sets, I will explain in Chapter 6 increasing the anchor points means increasing the number of fuzzy sets and the number of rule combinations, which will increase the computational complexity.

One can infer from Figure 4.1 that the word excessive could be ignored as an anchor point, given the considerable overlap of intervals between *excessive* and *adequate*. When used in a security context, a security expert may feel one can never have excessive security. I will explain in the upcoming Chapter 5 how the mean value of respondents ratings in the user-study remained around the adequate anchor point even in scenarios with increased security, which supports the claim that experts hesitate to rank security requirements as excessive.

Other researchers in the security domain, can benefit from the results of this word study by using adequacy scales for studies where security ratings are obtained from participants. Because of time constraints, we did not use psychometrics to validate the scales, but we envision that future studies that uses psychometrics to validate the scales would be beneficial to the research community.

Chapter 5

Capturing the Effect and Priorities of Composed Security Requirements

In Chapter 3, I have described challenges involved in security decision-making based on collecting qualitative data through expert interviews. I have also presented background and related work describing the challenges in security decision making in Chapter 2. In this chapter, I will report results of multiple user experiments [66, 68] conducted on security experts to examine how changing threats and requirements affect an expert's ability to perceive security risk and make corresponding decisions to prioritize security requirements. I will describe how I conducted user studies on a larger scale using a mix of quantitative and qualitative analysis methods, and I will describe the research methodologies that I adopted to handle the following challenges:

- The experts varying level of expertise and their stove-piped security knowledge and background.
- The composition of requirements corresponding to components of a system.
- The security requirements varying priorities: some requirements have higher priorities than others, depending on their strength in mitigating threats.
- The uncertainty in security decisions, that could result from ambiguity in abstract terminology that could lead to different experts interpreting the same requirement differently.
- The scarcity of security experts.

I will first motivate the use of factorial vignettes in the design of the security experts user studies, before explaining and reporting results from two studies conducted on a sample of researcher, graduate and undergraduate students from CMU and NCSU [68]. The purpose of these two studies was to examine security requirements composition through the use of factorial vignettes that I adapted from social science. Results and insights from this work led me to develop the Multifactor Quality Measurement method (MQM) [66]. The MQM models dependencies

Excerpts from this work were previously published as: H. Hibshi, T. Breaux, and S. B. Broomell, *Assessment of Risk Perception in Security Requirements Composition*, 2015 IEEE 23rd International Requirements Engineering Conference (RE), pp. 146-155, Aug. 2015; H. Hibshi and T. D. Breaux, *Reinforcing Security Requirements with Multifactor Quality Measurement,* in 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 2017, pp. 144-153.

among requirements, and estimates how these requirements affect a perceived level of quality in a requirements specification, called a scenario. I also explain how I applied the method to conduct user experiments on 69 security experts with average 10 years of experience to evaluates security scenarios in the four different security domains of networking, operating systems, databases and web applications [66]. The MQM provides a defined framework for researchers and requirements engineers. Whether the research is for academic or industry purposes, a researcher who wish to study a problem, phenomena, or a quality of interest (not necessarily security) where a number of dependencies exist between different factors, and where there is a demand for input from domain experts who happen to be scarce in that domain of interest.

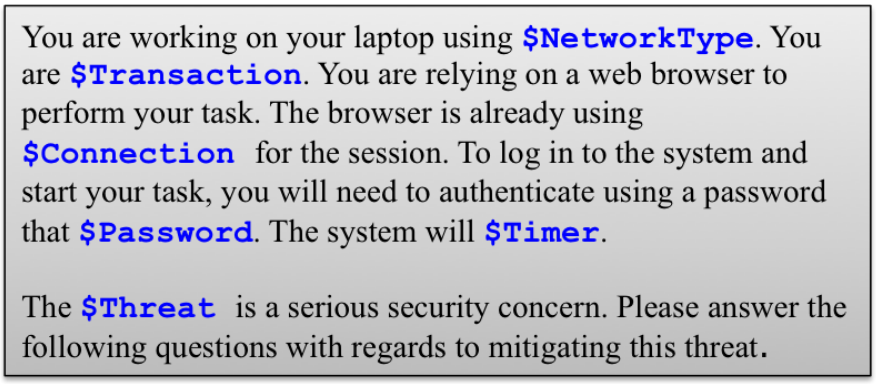
As explained earlier in Chapter 2, analysts still face challenges in the selection of the appropriate security requirements to mitigate threats. Security requirements exist in composition. The "composition in security requirements" means that any given security scenario could consist of multiple components, such as authentication and network type that can contribute differently to the security risk in the scenario. For an analyst to assess the risk and rate the overall security of a scenario, they would need to understand all the security requirements composed in the scenario, and how these requirements interact and contribute to increase or decrease the risk.

In the "SA study" described in Chapter 3, we interviewed 11 security experts and showed them artifacts that consist of multiple requirements to study how experts would analyze composable requirements and make security decisions. To study a larger scale of experts, it is more beneficial and practical to use controlled experiments. I will explain below how I designed my experiments using factorial vignettes, a methodology adopted from social science. I will also explain how the factorial vignette design helps to capture the composition effect of multiple security requirements that are arranged in a scenario. This approach allows us to isolate the effect of composition on security risk, and to address the limitations of differing levels of security expertise. To improve completeness and to help reduce ambiguity, the design asks analysts to report missing requirements.

5.1 Factorial Vignettes

The vignette experiments that I describe in this chapter are based on *factorial vignettes*, which are scenarios comprised of discrete factors that contribute to human judgment. Researchers systematically manipulate the factors to understand their composite and individual effects on a decision [106, 126]. Factorial vignettes are proven more effective to understanding decision making than direct questioning or single statement ratings that obscure the underlying contributions of different factors to the overall decision [3, 106, 126]. In addition, the use of factorial vignettes, increases experimental realism as participants react to scenarios that are similar to what a participant may experience in the real world [1].

Factorial vignettes are presented in surveys and user experiments using a basic template that contains multiple dimensions of the construct of interest. In our case, each dimension is a security requirement that influences the perceived level of security risk: some requirements increase risk, while others decrease risk. For example, Figure 5.1 shows the natural language text template that we used in our preliminary study to create the vignettes: a *vignette* is a standard scenario generated by the template, wherein each variable name (starting with a $) is replaced by a *level* in the corresponding dimension.



You are working on your laptop using $NetworkType. You are $Transaction. You are relying on a web browser to perform your task. The browser is already using $Connection for the session. To log in to the system and start your task, you will need to authenticate using a password that $Password. The system will $Timer.

The $Threat is a serious security concern. Please answer the following questions with regards to mitigating this threat.

Figure 5.1: A template used for vignette generation

In the study where we used the template shown in Figure 5.1, each level corresponds to a requirement or system constraint variant, which is either a quality requirement (e.g., a *weak* vs. *strong* password) or more concrete interpretation of an otherwise ambiguous requirement (e.g., *unencrypted* vs. *encrypted* Wi-Fi). Further in this chapter, I will be showing more vignette templates from our studies along with their dimensions and levels.

5.1.1 Related Work on Factorial Vignettes

Research methods using factorial vignettes have been applied in social and decision science, psychology, sociology, and marketing, to name a few [10, 126]. Factorial vignettes have also been used in security and privacy research. I will highlight below related work that uses factorial vignettes as a research methodology.

McKelvie et al. used factorial vignettes to investigate the effect of different types of uncertainty on the decision-making of entrepreneurs in software industry [1, 87]. Based on their results, the authors argue that entrepreneurs prefer to avoid uncertainty, but the extent of that avoidance is affected by the type of uncertainty, the magnitude of the decision, and the domain expertise [1, 87]. The authors argue that having their participants provide judgments to scenarios that consist of underlying factors, is an approach found to provide more accurate and less biased data when compared to other methods such as participant introspection [87]. In our work, we are also interested in investigating uncertainty, risk, and expertise, but in the security domain. We find factorial vignettes an approach that allows participants to rate scenarios composed of multiple security configurations. The analysis of participant data will help investigate the composed requirements in the scenarios and explain their effect on security expert decisions.

Factorial vignettes have been used in privacy research [15, 16, 82, 83, 84]. Martin used factorial vignettes on 1600 participants from Amazon's Mechanical turk to study how introducing privacy notices may impact consumer trust [82]. Bhatia et al. used factorial vignettes to study privacy risk and how it affects user willingness to share personal information [15, 16]. Martin and Nissenbaum use factorial vignettes to show that contextual elements highly affect individual privacy decisions which provides an explanation to the conflicting data found in privacy literature; that is when survey results show that people are concerned about privacy, while their real actions show that they do not care about sharing their personal information with websites [83]. Emami-Naeini et al. have used factorial vignettes to study user privacy expectations and preferences when using Internet of Things (IoT) technologies [38].

Researchers have applied factorial vignettes to study different factors that impact cybersecurity. To study compliance with cybersecurity policy, researchers used factorial vignettes to explore factors that lead to policy violations [76, 86, 118]. Gomez and Villar use factorial vignettes to study the effect of uncertainty on dealing with cyberthreats [56]. Factorial vignettes have also been used to examine end-user security decision-making (e.g. file download) and explore their security risk perception [64].

Except for the research conducted by Gomez and Villar [56], prior work mentioned above that used factorial vignettes in the usable security and privacy field have focused on end-users. Gomez and Villar recruited computer science university students and treated them as to be the experts in their online experiments [56]. In our research, we focus on security experts. I will show in the upcoming sections of this chapter how we focused on recruiting industry experts. In our preliminary study, we recruited graduate and undergraduate students who perform research in cybersecurity or who are enrolled in security classes. In addition to self-reported expertise questions, we include a security knowledge test in our studies to be able to assess a participant's security expertise.

We have also applied factorial vignettes to a new application domain: security requirements. We treat security requirements as factors and we manipulate these requirements by using specifications that should increase or decrease security in a vignette. Prior work mentioned above have focused on studying end-user related factors such as: effect of personality traits on employees compliance with company's information security policy, presence of compliance policies on the likelihood of playing online games[118], trust in cyberspace [56], and effect of gain-and-loss on end-user security decisions [64].

5.2 Preliminary Study on Security Requirements Composition

We conducted a preliminary study that uses factorial vignettes to study the composition effect in security decision-making. This study, which we will refer to as the *requirements composition study*, consists of two online experiments that elicit risk perceptions from multiple analysts and target the mitigating effects of specific requirements to the threats they address. This approach examines the effect of composition on security risk, and address the limitations of differing levels of security expertise. To reduce incompleteness and ambiguity, the design asks analysts to report missing requirements.

5.2.1 Approach for the Requirements Composition Study

I will now explain our research methodology for the preliminary study. This includes the research questions, vignette design, survey design, deployment and subject recruitment, and the analysis approach.

Research Questions

The survey used in the preliminarily study was designed to answer three research questions:

RQ1. Does requirements composition affect risk perception in a security scenario to cause varied ratings of the security adequacy level, or can requirements be treated independently in a checklist?

RQ2. Which security requirements in a security scenario contribute more weight to experts security adequacy judgment?

RQ3. Would experts be able to detect ambiguities in a security scenario and provide modifications to improve the security adequacy ratings?

To answer these questions, the survey instrument was designed with three parts: the security vignettes, a security knowledge test, and a demographics test. Below, I describe the factorial vignettes design, followed by the overall survey design.

Factorial Vignettes Design

We use factorial vignettes to design the study where we ask users to judge security scenarios that include multiple security requirements. Figure 5.1 that was introduced above, shows the template that we used in the study to create the vignettes. In Table 5.1, we present the dimensions and levels to Figure 5.1. Each level has a code (in parentheses) that we used to analyze and report our results. The level (`$Threat = Man-in-the-Middle`) occurs when an attacker intercepts the encrypted communication between two parties by decrypting the encryption. The level (`$Threat = Packet Sniffing`) is passive in that the attacker eavesdrops on network packets to steal information without interacting with any parties, directly.

The choice of dimensions and levels in factorial vignettes is determined by the researcher's judgment based on the research questions. We seek to evaluate the effect of changes in requirements composition and in threats where the composition spans a range of security knowledge, including network and application security, perceived sensitivity of information, and general *best practice* vs. threat-targeted mitigations. The dimensions that we chose are not the only dimensions that can be evaluated. In addition, the number of levels for each dimension is not the only number that exists.

In factorial vignette design, the space of all possible dimensions and levels is called the factorial object universe [106] and the factorial object sample is the sample across the universe that we use to instantiate the vignette template [106]. Sampling is random or systematic and the choice is based on prior theory, research, and reasoning [75]. Factorial sampling is used to eliminate unrealistic combinations of levels and to exclude scenarios that are likely to produce a predictable outcome [126]. Sampling from vignettes is more efficient than classic factorial designs, wherein all possible combinations of factors are tested [106].

Table 5.1: Vignette dimensions and their levels in the security requirements composition study

Dimension	Level Code	Level(s)
$NetworkType	EmpNetwork	Your employer's network at your office
	PublicWIFI	Public unencrypted Wi-Fi at a public area (restaurant, airport)
	VPNUnencrypted	Your employer's VPN that you connected to through public unencrypted Wi-Fi
	VPNEncrypted	Your employer's VPN that you connected to through public encrypted Wi-Fi
$Transaction	Email	Accessing your email account and replying to confidential email
	Financial	Performing a financial transaction using your credit card
$Connection		SSL
$Password	Weak	A password that is at least 8 characters long
	Strong	A password that is at least 16 characters and must include lowercase letter, a symbol, and a number digit
$Timer	Yes	Automatically log you off the session after 15 minutes of inactivity
	No	Never time-out
$Threat	M-i-t-M	Man-in-the-Middle
	Packet-Sniffing	Packet-Sniffing

The initial scenario about logging into a remote e-mail service was chosen because it crosses between novice and expert security knowledge, and this would allow us to measure the effect of security expertise on risk perception. We reviewed the universe and selected dimensions that had a sufficient number of levels to provide a rich space from which to sample; this includes network types and password complexity. Based on Table 5.1, we have 32 ($4 \times 2 \times 1 \times 2 \times 2$) conditions per $Threat type.

Our vignette selection is based on removing unrealistic and idiosyncratic scenarios. For example, the $Connection dimension consists of one level, only, which is called a blank dimension. While we can evaluate unencrypted HTTP sessions in a scenario, the prevalence of knowledge about the high risk of unencrypted sessions suggests this level would predictably lead respondents to rate this requirement as inadequate to protect against the chosen threats. Blank dimensions are included in the vignettes, but not as statistical variables in the analysis, because they have no statistical effect to be measured. That said, blank dimensions are not to be eliminated, because their presence and absence affect how participants make decisions. In our case, removing SSL introduces an ambiguity: some participants may assume it exists, while others may assume it is absent. To control for this variability, we made this requirement explicit.

Survey Design and Research Questions

As stated above, the survey instrument was designed with three parts: the security vignettes, a security knowledge test, and a demographics test. In addition, each participant receives a consent form noting that participation is voluntary. Participants were with the Man-in-the-Middle threat, where they answer all three parts of the survey. A week after taking the survey, participants were invited back for the Packet-Sniffing threat, where they do not repeat the security knowledge test or the demographic questions.

The Security Vignettes In this study, each participant rated four vignettes to observe all four network levels (see Table 5.1). Since we have a total of 32 vignettes per threat, we have 8 possible combinations of the dimensions and, thus, each participant is randomly assigned to one of eight conditions, where they rate four vignettes ($8 \times 4 = 32 \quad vignettes$). Each condition randomly assigns the participant to a single level of the `$Transaction`, `$Password`, and `$Timer` dimensions (between-subjects effect), which are the same across all the four vignettes that the participant rates. The four vignettes differ by the `$NetworkType` dimension (within-subjects effect) and are presented in a randomized order. For all four vignettes, a participant is asked to first rate the overall security level of the scenario within the context of the given threat. The rating levels are displayed in a random order from the following list:

- **Excessive** security measures that exceed the requirements to mitigate the threat
- **Adequate** security measures that are enough to mitigate the threat
- **Inadequate** security measures that are not enough to mitigate the threat

Next, we ask participants to rate the dimension levels based on the security requirement's ability to mitigate the given threat. This mitigation rating is applied to factors that represent a mitigation that can be modified to improve security: `$NetworkType`, `$Connection`, `$Password` and `$Timer`. Participants provide their rating on a 5-point semantic-scale, where point 1 is labeled *inadequate mitigation*, point 3 is labeled *adequate mitigation* and point 5 is labeled *excessive mitigation*. For each such dimension, we list the selected level for the vignette from Table 5.1. These ratings are used to test which requirements (or factors) affect the overall security.

Participants are also given the opportunity to list additional security requirements that they believe contribute to increasing the security level to adequate. These are open-ended responses that are later analyzed using grounded analysis [107].

The Security Knowledge Test Following the vignettes, participants are required to answer ten security knowledge questions. These questions were selected to cover from user to administrator-level security knowledge, including cryptography, firewall rules, encryption, hashing, file permissions, and network security. The questions cover security concepts, and are intentionally inconvenient to search for on the Internet to reduce cheating. The responses are used to calculate a score that serves as a proxy expertise metric.

Demographics Survey At the end of the survey, participants answer questions about job experience and security training. It is recommended to place background and demographics questions towards the end of surveys to avoid potential bias and to increase participants response rate [105].

Deployment and Subjects Recruitment

Security experts were recruited using e-mail invitations to participate in the Man-in-the-Middle study (32 vignettes, where each participant sees 4 vignettes). The invitations was sent to security class mailing lists at Carnegie Mellon University and North Carolina State University. We also sent invitations to security-research mailing lists at Carnegie Mellon University. Participants were compensated with a $10 Amazon gift card for participation. A week after taking this study, the participants were invited back to the Packet-Sniffing study (another 32 vignettes, where each participant sees 4 vignettes), and compensated with a second $10 Amazon gift card.

Analysis Approach

Now I will explain the multi-level modeling and grounded analysis used for the analysis.

Analysis of Multi-level Models Multi-level models are statistical regression models with parameters that account for multiple levels in datasets [54]. Our study design described above supports both within and between subjects effects (mixed-effects). We treated the data as two studies based on the two levels of the `$Threat` dimension, which we assume the participant responses to the two threats are independent due to the week delay between surveys.

The quantitative dataset consists of one major dependent variable: the `$OverallRating`, which is the security experts judgment rating of the overall security level. This variable has three possible values -1, 0, or 1 that correspond to inadequate, adequate or excessive security, respectively. The fixed effects independent variables are the dimensions: `$NetworkType`, `$Transaction`, `$Password`, `$Timer`, which we will refer to as requirements-mitigation variables. The random effect, independent variable is grouped by participant `$ID`, because we have repeated measures for each subject who sees four levels of `$NetworkType`. We have four dependent mitigation-rating variables: `$NetworkRating`, `$Connection-Rating`, `$PasswordRating`, and `$TimerRating`, which correspond to individual ratings of the dimensions: `$NetworkType`, `$Connection`, `$Password`, and `$Timer`, respectively. Mitigation-rating variables are assigned an integer from 1-5.

Knowledge is quantified using a `$Score` variable, which is an independent exploratory variable assigned an integer from 0-10 equal to the number of correct answers provided by the participant to the 10 security screening questions.

The data is analyzed using multi-level modeling [54] to account for our mixed effect experiment design. Tools to conduct the analysis include R [102] and lme4 [13]. As described earlier, each participant rated all four levels of the `$NetworkType` dimension, while only rating one level of the remaining dimensions. Hence, the analysis simultaneously accounts for dependencies in the repeated measures, calculates the coefficients (weights) for each explanatory independent variable, and tests for interactions. The significance of the multi-level models is tested using the standard likelihood ratio test by: fitting the regression model of interest; fitting a null model that excludes the independent variables used in the first model; computing the likelihood ratio; and then, reporting the chi-square, p-value, and degrees of freedom [54]. For fitted models that show statistical significance, the coefficient values from the regression model are reported, which represents the dimension weight for predicting the dependent variable.

To determine sample size, a priori power analysis was conducted using the G*Power [45] tool to test for the required sample size of repeated measures ANOVA. An estimated sample size >96 was needed per threat scenario for the recommended power level of 0.8 and a medium-sized effect [27].

Grounded Analysis The open-ended questions to elicit mitigation requirements were analyzed by first excluding seven non-mitigation responses. Then, open coding was applied [29, 55] to code responses with short phrases (concept labels) and then group the phrases into six emergent categories: *server*, if the requirement is the responsibility of a web server, *client*, if the requirement is the responsibility of an application on the user's computer (e.g., a browser); *encryption*, if the requirement primarily concerns encrypting data or communications; *private network*, if the requirement suggests switching to a non-public network; *attack detection and prevention*, if the requirements is aimed at preventing and/or addressing certain attacks; and *identity and authentication*, if the requirement concerns verifying the identity of the user or their device.

After first cycle coding and categorization, a second-cycle coding [107] was conducted, wherein the categories were linked to vignette dimensions and a direction as follows: a *refinement*, if the requirement refines the dimension by extending its functionality; a *reinforcement*, if the requirement adds auxiliary security not directly related to the dimension; a *generalization*, if the requirement is more general than the dimension, but includes the dimension's mitigation; and a *replacement*, if the requirement replaces the dimension. For example, two requirements, multi-factor authentication and password expiry policy, are coded by the *password* dimension, yet the former is a replacement, because it replaces *password* with new functionality, and the latter is a refinement, because it extends *password* with expiration.

5.2.2 Results of the Requirements Composition Study

I will now present the quantitative and qualitative results of the *Requirements Composition* study.

Descriptive Statistics

A total of 174 participants responded to the Man-in-the-Middle threat survey, of which, 116 returned to respond to the Packet-Sniffing survey. These sample sizes exceed what we estimated prior to conducting the study. The sample consists of 26% females and 73% males (1% unreported gender). The age groups sorted by dominance in the sample are 18-24 (63%), 25-34 (33%), and 35+ (3%). Within the sample there are 101 graduate students, 42 undergraduate students and 2 university professors.

The average number of participants per vignette is 22 for the Man-In the-Middle threat, and 15 for the Packet-Sniffing threat. The number of participants is close but not equal across vignettes due to randomization. Tables 5.2 and 5.3 present descriptive statistics of participant ratings.

Table 5.2: Descriptive statistics of the `$OverallRating` variable

	Man-in-the-Middle			Packet_Sniffing		
	Adequacy Scale			*Adequacy Scale*		
	1	0	-1	1	0	-1
`$OverallRating`	5 %	53%	42%	7%	92%	0%

Table 5.3: Descriptive statistics of the variables for requirements ratings

	Man-in-the-Middle					Packet_Sniffing				
	Adequacy Scale					*Adequacy Scale*				
	5	**4**	**3**	**2**	**1**	**5**	**4**	**3**	**2**	**1**
`$NetworkRating`	1%	9%	37%	21%	32%	2%	7%	36%	22%	33%
`$ConnectionRating`	2%	12%	68%	17%	1%	0%	11%	71%	15%	3%
`$PasswordRating`	7%	17%	43%	21%	12%	8%	13%	39%	26%	14%
`$TimerRating`	2%	11%	29%	17%	41%	4%	12%	27%	21%	36%

*Percentages are calculated with respect to each threat study sample;
adequacy scale 5=Excessive, 3=Adequate, 1=Inadequate

The Overall Rating

The `$OverallRating` variable is the major outcome dependent variable of interest, because this variable represents the experts security rating of the scenario based on the composition of the requirements. Equation 5.1 is our main additive regression model with a random intercept grouped by participant ID. The additive model is a formula that defines the `$OverallRating` in terms of the intercept (α) and a series of components. Each component is multiplied by a coefficient (β) that represents the weight of that variable in the formula. This formula in Equation 5.1 is simplified as it excludes the dummy (0/1) variable coding for the reader's convenience.

$$\begin{aligned} \$OverallRating = \alpha + \beta_N \$NetworkType + \beta_T ran \$Transaction + \\ \beta_P \$Password + \beta_T ime \$Timer + \epsilon \end{aligned} \tag{5.1}$$

For the model above, we will refer to the predictor explanatory variables: `$NetworkType`, `$Transaction`, `$Password`, and `$Timer` as the four predictors. The β parameters in Equation 5.1 represent the weight of each dimension in explaining the data. We tested the significance of the main effects in the additive model (Equation 5.1); and then the interaction terms, which are the added terms generated by multiplication of the explanatory variables terms in the additive model. The indicator variables are dummy coded (0/1) to represent the dimension levels (see Table 5.1) To compare the `$OverallRating` across vignettes, we establish a base level for each variable that fixes the variables. The intercept (α) is the sample's mean outcome in the base case, which includes the following base levels:

- Employer network for the `$NetworkType`,
- Email for `$Transaction`,
- Strong password for `$Password` and,

- No timer for $Timer.

For the Man-in-the-Middle threat, we found a significant contribution of the four predictors for predicting the $OverallRating ($\chi^2(6) = 142.2, p < 0.001$) but failed to find a significant contribution from the interaction terms ($\chi^2(11) = 4.8, p = 0.94$). For the Packet-Sniffing threat, the $OverallRating is also affected by the same four predictor variables with a significant value over the null model ($\chi^2(6) = 20.4, p = 0.002$). We also did not see any significance for the interaction model ($\chi^2(11) = 6.6, p = 0.83$). These results suggest that the four dimensions $ NetworkType, $Transaction, $Password, and $Timer are good predictors that explain change in the experts overall rating. However, it is important to note here that the dataset from the Packet-Sniffing threat is less predictive in explaining the $OverallRating variable due to the violation of the normality assumptions. This is due to the unforeseen effect of no participant choosing the inadequate rating in vignettes with this threat type (see Table 5.2), which reduced the response levels from three to two.

Table 5.4 shows the assigned coefficient weights (labeled by β in the table) along with standard errors and significance levels for the two threat datasets. These weights represent the amount of change in rating caused by the corresponding change in predictor variable level. From the table, we conjecture that the Man-in-the-Middle threat has a significant intercept of -0.24, which indicates that at the base case (employer's network, email transaction, strong password, and no timer), the mean of the $OverallRating is lower than adequate (adequate= 0). Since, $ NetworkType is the only dimension showing significance in Table 5.4, we further interpret the intercept to indicate the mean adequacy level in the case of the employer's network. Interestingly, the public Wi-Fi network and the VPN over unencrypted network significantly decreased the overall rating from the base level employers network. Another interesting observation in Table 5.4 is that the VPN over encrypted network significantly increase the overall rating in the Packet-Sniffing threat scenario, while this has no effect in the Man-in-the-Middle threat. This result is expected from security experts who understand the difference among the two threats: encryption is a reasonable protection against Packet-Sniffing as attackers would not benefit from sniffing encrypted packets, but encryption alone is not enough to mitigate Man-in-the-Middle wherein attackers intercept and decrypt encrypted communication.

Table 5.4: Regression results for the $OverallRating variable

Variable-level	**Man-in-the-Middle**	**Packet Sniffing**
	β ***(Std. Error)***	β ***(Std. Error)***
$Intercept	$-0.24(0.07)^{***}$	$0.10(0.06)$
$Network-PublicWIFI	$-0.50(0.05)^{***}$	$-0.03(0.05)$
$Network-VPNEncrypted	$0.03(0.05)$	$0.10(0.03)^{**}$
$Network-VPNUnencrypted	$-0.24(0.05)^{***}$	$-0.04(0.04)$
$Transaction-Financial	$-0.03(0.06)$	$-0.02(0.04)$
$Password-weak	$0.06(0.07)$	$-0.05(0.05)$
$Timer-yes	$0.14(0.08)$	$0.03(0.06)$

$^{*}p \leq .05,^{**} p \leq .01,^{***} p \leq .001$ with standard errors in parentheses

The Security Requirements Effect

We further examine the effect of each requirement in the security scenario by analyzing participants 5-point Likert-scale ratings of the specific mitigations. To do this analysis, we use the same regression formula in Equation 5.1, but replace the `$OverallRating` outcome variable with `$NetworkRating`, `$ConnectionRating`, `$PasswordRating`, or `$TimerRating`.

The Network Effect The `$NetworkRating` is a measure of the participants adequacy rating of the network in the scenario to get more insight into how experts formed their `$OverallRating` of the scenario. We found a significant contribution of the four predictors for predicting the `$NetworkRating`. This significant result applies to both threat scenarios: Man-in-the-Middle ($\chi^2(6) = 322.1, p < 0.001$), and Packet-Sniffing ($\chi^2(6) = 209, p < 0.001$). As with the variable `$OverallRating`, we did not find any added significance from the interaction terms for both threats: Man-in-the-Middle ($\chi^2(11) = 6.4, p = 0.84$), and Packet-Sniffing ($\chi^2(11) = 6.3, p = 0.85$).

Table 5.5 shows the detailed results of the regression model for the `$NetworkRating` outcome variable. From the intercept value, we conjecture that participants rated the base case (employer's network, email transaction, strong password, and no timer) slightly lower than adequate (adequate = 3). The table also shows how the network type has a significant effect on the `$NetworkRating` variable. In both threat scenarios, changing from the employer's network to the public Wi-Fi network decreased the rating by more than one point. On the other hand, the VPN over encrypted Wi-Fi significantly increased the `$NetworkRating` adequacy level over the employer's network. For the Packet-Sniffing threat, the VPN over unencrypted network did not have an effect on the network rating for that threat. This means that participants view the VPN over unencrypted Wi-Fi and the employer's network to be at the same security adequacy level.

Table 5.5: Regression results for the `$NetworkRating` variable

Variable-level	Man-in-the-Middle	Packet Sniffing
	β ***(Std. Error)***	β ***(Std. Error)***
`$Intercept`	$2.70(0.10)^{***}$	$2.43(0.14)^{***}$
`$Network-PublicWIFI`	$1.28(0.08)^{***}$	$1.13(0.10)^{***}$
`$Network-VPNEncrypted`	$0.35(0.08)^{***}$	$0.47(0.10)^{***}$
`$Network-VPNUnencrypted`	$-0.35(0.08)^{***}$	$0.18(0.10)$
`$Transaction-Financial`	$0.14(0.08)$	$0.08(0.10)$
`$Password-weak`	$0.07(0.09)$	$0.06(0.13)$
`$Timer-yes`	$0.06(0.11)$	$0.05(0.14)$

$^{*}p \leq .05,^{**} p \leq .01,^{***} p \leq .001$ with standard errors in parentheses

Another observation from Table 5.5 is the absence of effect for the other requirements on the `$NetworkRating` adequacy. There are two possible explanations for this result: 1) when participants are rating the network, they isolate it from all other requirements and they only focus on looking at the network type, and/or 2) participants are assigning a higher priority to the `$NetworkType` so it acts as the deciding factor and it supersedes other requirements in the scenario.

The SSL Connection Effect We found slight statistically significant contribution of the four predictors predicting the `$ConnectionRating` adequacy level for the Man-in-the-Middle threat ($\chi^2(6) = 15.1, p = 0.02$), but no significant contribution in the Packet-Sniffing dataset ($\chi^2(6) = 5.8, p = 0.5$). When we further examined the regression model of the Man-in-the-Middle dataset, we found significance only for the intercept ($\alpha = 2.9, SE = 0.10, p < 0.001$) and the public Wi-Fi network ($\beta = 0.10, SE = 0.05, p = 0.03$). This means that the mean for the `$ConnectionRating` in the base case is around adequate, while it slightly drops when the network changes from employer's network to a public Wi-Fi. One possible interpretation of these results could be that the presence of SSL in the scenario is crucial and that is why the mean is around adequate, but the adequacy rating does not significantly change with the change of other requirements except if the change is to an extremely low level of security such as Public Wi-Fi.

The Password Strength Effect The `$PasswordRating` is a measure of the participants adequacy rating of the password strength in the scenario. The four-predictor model significantly increases model fit of `$PasswordRating` over the null model. This is present in both threat scenarios: Man-in-the-Middle ($\chi^2(6) = 37.6, p < 0.001$), and Packet-Sniffing ($\chi^2(6) = 38.6, p < 0.001$). Similar to the above outcome rating variables, the interaction terms do not significantly increase the model fit for the Man-in-the-Middle threat ($\chi^2(11) = 11.7, p = 0.38$). Although the Packet-Sniffing threat showed a significant effect ($\chi^2(11) = 22.5, p = 0.02$), the coefficients did not show significant p-values for the interaction terms, which may indicate that the added significance was distributed across the terms.

Table 5.6 shows the details of the regression model for the `$PasswordRating` variable. In both scenarios, the intercept at the base case where the password is strong shows significant adequate ratings, that drops significantly when the network changes from employer's network (base case) to public Wi-Fi. Changing the password strength from strong to weak also drops the password adequacy rating in both threat scenarios.

Table 5.6: Regression results for the `$PasswordRating` variable

Variable-level	**Man-in-the-Middle**	**Packet Sniffing**
	β ***(Std. Error)***	β ***(Std. Error)***
`$Intercept`	$3.33(0.16)^{***}$	$3.16(0.22)^{***}$
`$Network-PublicWIFI`	$0.15(0.05)^{***}$	$0.18(0.05)^{***}$
`$Network-VPNEncrypted`	$0.01(0.05)$	$0.06(0.05)$
`$Network-VPNUnencrypted`	$0.05(0.04)$	$0.06(0.05)$
`$Transaction-Financial`	$0.05(0.14)$	$0.14(0.19)$
`$Password-weak`	$0.76(0.17)^{***}$	$0.68(0.23)^{**}$
`$Timer-yes`	$0.10(0.20)$	$0.15(0.26)$

${}^{*}p \leq .05, {}^{**}p \leq .01, {}^{***}p \leq .001$ with standard errors in parentheses

The Auto-logoff Timer Effect The `$TimerRating` is a measure of the participants adequacy rating of the auto-logoff timer in the scenario. The four-predictor model significantly increases model fit of `$TimerRating` over the null model. This is present in both threat scenarios: Man-in-the-Middle ($\chi^2(6) = 54.9, p < 0.001$), and Packet-Sniffing ($\chi^2(6) = 49.2, p < 0.001$). Similar to the above outcome rating variables, the interaction terms do not significantly increase the model fit for the Man-in-the-Middle threat ($\chi^2(11) = 17.4, p = 0.09$), or the Packet-Sniffing threat ($\chi^2(11) = 12.9, p = 0.30$).

Table 5.7 shows the details of the regression model for the `$TimerRating` variable. Note that the intercept shows a low mean that is close to inadequate (recall from Section III: inadequate = 1) which is expected since the base level has no auto log-off timer. In the presence of the Man-in-the-Middle threat, the `$NetworkType`, `$Password`, and `$Timer` dimensions have a significant impact on participants `$TimerRating`. The public Wi-Fi, VPN over unencrypted Wi-Fi, decreased the adequacy level of the `$TimerRating` variable, while turning the auto logoff timer on had significantly increased the adequacy level of the `$TimerRating`. In the case of the Packet-Sniffing threat, the network type did not have a significant impact on predicting the `$TimerRating`, but the presence of the timer in the scenario shows a significant increase in the `$TimerRating` compared to the base case where no timer is involved.

Table 5.7: Regression results for the `$TimerRating` variable

Variable-level	**Man-in-the-Middle**	**Packet Sniffing**
	β ***(Std. Error)***	β ***(Std. Error)***
`$Intercept`	$1.79(0.17)^{***}$	$1.51(0.22)^{***}$
`$Network-PublicWIFI`	$0.18(0.05)^{***}$	$0.08(0.05)$
`$Network-VPNEncrypted`	$0.0(0.05)$	$0.06(0.05)$
`$Network-VPNUnencrypted`	$0.12(0.05)^{**}$	$0.07(0.05)$
`$Transaction-Financial`	$0.25(0.15)$	$0.22(0.18)$
`$Password-weak`	$0.60(0.18)^{***}$	$0.82(0.22)^{***}$
`$Timer-yes`	$1.18(0.21)^{***}$	$1.60(0.25)^{***}$

$^{*}p \leq .05,^{**} p \leq .01,^{***} p \leq .001$ with standard errors in parentheses

It is strange and unexpected that the weak password is showing a significant increase in the timer adequacy rating in both scenarios. It is possible that this is a Type I error (i.e., the password did not actually play a role in the decision and this effect is only random) or is due to an interaction effect between password and the other predictor variables. When we examined the coefficients of the interaction model, we observed that the weak password significantly interacts with other variables such as public Wi-Fi and VPN over unencrypted Wi-Fi, which makes us lean more towards the interaction explanation although the data does not show evidence of interaction.

The Knowledge Effect The `$Score` variable is our indicator variable for experience, as it represents participants score (out of 10) on the security test. Scored responses to our knowledge test presented a minimum score of 1 and a maximum of 10, with a mean 5.2 and a median of 5. We added the experience predictor variable (`$Score`) to Equation 5.1 and compared the new model to the four-predictor model in 5.1 for both threat types. The new model with the experience indicator (`$Score`) did not significantly improve the prediction of the overall variable compared to the original model with the four predictors alone. We repeated the same comparison for all the four mitigation-rating variables: $NetworkRating, $ConnectionRating, $PasswordRating, and $TimerRating. Except for the $PasswordRating and the $TimerRating in the Man-in-the-Middle threat, the ($Score) variable did not significantly improve the prediction of the ratings variables.

In the presence of the Man-in-the-Middle threat, adding the experience indicator ($Score) to the four predictor model improved the prediction of the $PasswordRating ($\chi^2(1) = 1.8, p < 0.001$). The coefficient weight in the model shows that the $PasswordRating significantly decrease by -0.15 as the experience indicator ($Score) increases. Similarly, adding ($Score) to the four predictors model improved the prediction of the $TimerRating ($\chi^2(1) = 8.2, p = 0.004$). The coefficient weight in the model shows that the $TimerRating significantly decrease by âĂŞ0.11 as the experience indicator ($Score) increases. In other words, more knowledgeable participants (with higher $Score) tend to act more conservative when rating the adequacy level of the password and timer mitigations.

Grounded Analysis of Suggested Mitigations

We elicited 905 mitigations from 108 participants: 540 for Man-in-the-Middle (104 participants) and 365 for Packet-Sniffing (64 participants). We organized the mitigations into 6 categories. Figure 5.2 shows all 6 categories with mitigation concepts under each category. We analyzed elicited mitigations in response to the network effect, because our statistical results suggest that the $NetworkType has the most influence on participants judgments. Table 5.8 shows for each $NetworkType, the number of mitigations provided by participants, the number of respondents providing these mitigations, and total mitigations. Table 5.9 shows the number of *refinements*, which are elaborations on an existing security requirement in the vignette (e.g., SSL, VPN); *reinforcements*, which describe auxiliary or new security functionality intended to complement existing requirements; *replacements*, which describe a requirement to supplant an existing requirement (e.g., WPA2 supplants WEP); and *generalizations* (Gen.), which describe more abstract requirements (e.g., secure network v. VPN).

Table 5.8: Number of mitigation requirements by threat and network type

$NetworkType	**Man-in-the-Middle**		**Packet-Sniffing**		**Total**
	Mitigations	Responses	Mitigations	Responses	Mitigations
Employer's Network	129	73	100	51	229
Public Wi-Fi	162	82	110	57	272
VPN over Unencrypted Wi-Fi	135	73	79	47	214
VPN over Encrypted Wi-Fi	114	73	76	42	190

In Table 5.8, the weakest network type Public Wi-Fi has the highest number of mitigations for both threat types. Notably, Table 5.9 includes 155 auto-log off timer mitigations suggested by participants who observed no auto logoff timer in the vignette, and 107 complex-password mitigations suggested by participants who observed a weak password in the vignette. After removing such refinements that we expected to see in the lower security dimension levels, we found 125 refinements remaining. Additional findings are highlighted below.

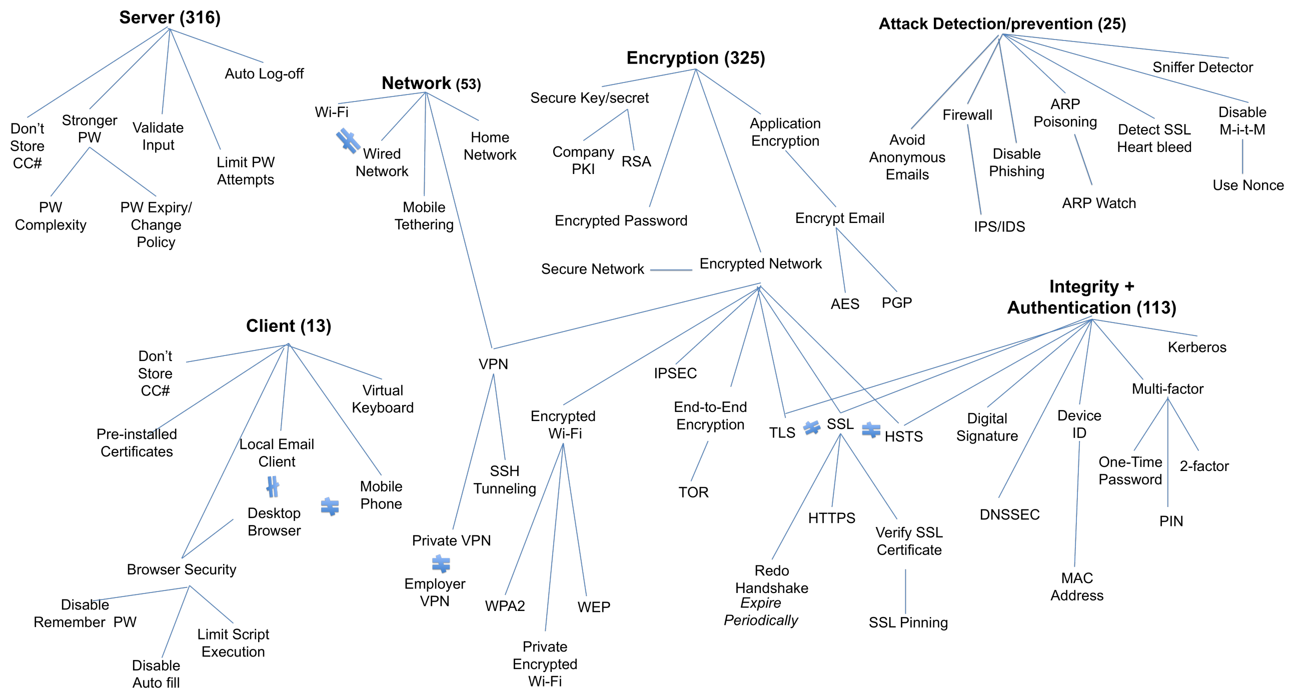


Figure 5.2: The elicited requirements and their categories (numbers in parentheses correspond to number of statements)

Table 5.9: Refinements, reinforcements, replacements, and generalizations requirements by network type

$NetworkType	Refinements	Reinforcements	Replacements	Gen.	Total
Employer's Network	107	41	63	18	229
Public Wi-Fi	88	33	122	29	272
VPN over Unencrypted Wi-Fi	91	23	78	22	214
VPN over Encrypted Wi-Fi	101	23	57	9	190
Total	387	120	320	78	905

Several refinements served to remove ambiguity. For example, we found 51 mitigations that refine SSL, such as requiring updates or patching the heart bleed vulnerability [121]. One participant suggested using WPA2 encrypted Wi-Fi, because the Wi-Fi encryption was unspecified. Two participants stressed that VPN over encrypted network should use a reliably strong encryption.

Among reinforcements, we found 25 mitigations proposing attack detection / prevention techniques (see Figure 5.2), 24 mitigations adding email encryption under the email transaction condition, and 8 requirements to add browser security and pre-installed SSL certificates, among others. Some reinforcements were inspired by the vignette: four mitigations against man-in-the-middle attacks, four against packet sniffing, and two against email phishing attacks.

Replacement mitigations aim to replace a less secure requirement or constraint with a more secure alternative. We found 95 mitigations to replace the password with multi-factor authentication. We also found 21 mitigations to replace SSL with TLS or HSTS, which is a recent security proposal receiving more attention [37, 81].

5.2.3 Discussion of the Security Requirements Composition Study

Results from the multi-level modeling and the grounded analysis suggest that risk perception varies with how requirements are composed. The coefficients obtained from the regression suggest that there are weights and priorities assigned to the requirements.

The `$OverallRating` variable is the major outcome dependent variable of interest, because this variable represents the experts security rating of the scenario based on the composition of the requirements. The multi-level regression results indicate that the `$NetworkType` is the only dimension that had an effect on experts `$OverallRating` of the security scenario. This does not mean the other dimensions had no effect on expert judgment. These estimates imply that the network type had the most influence (weight) on judgments of overall rating and the importance of each network type depends on the type of `$Threat`.

A composition is observed across the participants `$PasswordRating`, `$TimerRating`, and `$ConnectionRating` and from the grounded analysis results. When participants rated the password level adequacy, the `$PasswordRating` was lowered by the Public Wi-Fi network level, even when the password level was strong. Similarly, the `$TimerRating` was lowered by the use of Public Wi-Fi or VPN over unencrypted Wi-Fi. When the `$NetworkType` changes to Public Wi-Fi, respondents rate the strong password and auto-logoff timer as less than adequate, because participants likely view these two requirements as reinforcements that raise the general level of security, but do not mitigate the threat. From grounded analysis, we can further observe that participants were focusing their attention on providing requirements to replace the weak network. One participant stated that the timer, password, and SSL are no longer effective, if the communication is happening over a vulnerable network like Public Wi-Fi. Another participant explained that, despite the use of employer's VPN, a public unencrypted Wi-Fi could still be vulnerable. In addition, our multi-level modeling results for the `$ConnectionRating` show that for the Man-in-the-Middle threat, participants generally rated SSL near adequate, but the ratings dropped in the presence of Public Wi-Fi. Moreover, we saw participants providing requirements refinements for SSL regardless of change in dimensions levels. For example, five participants suggested to update the SSL version, and five participants suggested to verify SSL certificates and they replicated these modifications for all four-network types. This observation might be considered evidence for a ceiling effect with adequacy, meaning analysts did not perceive a limit to how much security they could afford.

The suggested refinements for SSL levels indicate that our proposed vignettes are incomplete, and that we should broaden the scope of our composition to include new dimensions/levels than what we proposed. Our grounded analysis also confirms that there are more dimensions to consider, such as browser security configurations. Secure communication relies on the browser's configuration, as we found 17 browser security reinforcements that 11 participants proposed as mitigations to increase the overall security level. Among these, seven browser security reinforcements were suggested in the presence of the employer's network and/or VPN over Encrypted Wi-Fi. After examining all the mitigations provided by these participants, we found that when

`$NetworkType` is weak, because participants marked it as inadequate or propose to replace it. When the risk is lowered by using a more secure `$NetworkType`, participants propose requirements that target other dimensions to increase the overall security level.

The grounded analysis in this study also shows how experts identified ambiguous requirements proposed to reinforce, replace, and/or refine these requirements. The vignette dimensions were observed to affect participants risk perception leading them to list mitigations based on the dimensions and their levels. For example, participants focus attention on replacing weaker requirements with stronger levels (e.g. replacing Public Wi-Fi), and that explains the high number of replacement mitigations provided for public Wi-Fi (see Table 5.9). In addition, out of the total 907 mitigations, only 78 (9%) were unrelated to our dimensions in the study as they are member of categories such as browser security and device identifiers (see Figure 5.2 for categories). Regarding ambiguity, we note that participants might assume that the public Wi-Fi is unencrypted, because vignette description omits mention of encryption. Similarly, the vignette does not provide details about the SSL dimension and participants made their own assumptions that made them list mitigations of refinements (e.g. version update), reinforcement, (e.g. certificate verification), and even replacement (e.g. TLS). This observation suggests two things with regards to ambiguity resolution: 1) when participants make assumptions to resolve ambiguity, they might lean towards assuming lower security (e.g. unencrypted Wi-Fi, insecure SSL versions); and 2) adding and removing requirements in a composition can have interactions by increasing or decreasing levels linked to the refined requirement (e.g. SSL).

5.2.4 Conclusions from the Security Requirements Composition Study

The purpose of this security requirements *Composition Study* is to empirically examine hypotheses generated earlier by the SA study. We summarize the findings below:

Security requirements exist in composition Our study showed some evidence that assessment of requirements relies on how they are composed together along with other requirements. Participants did not judge security requirements independent of other existing requirements in the scenario. For example, the network type affected the ratings of other requirements involved in the scenario (e.g. password, timer) as participants were evaluating each factor involved in the scenario to make their judgment.

Certain security requirements have more weight In the composition study, we have seen that until the security of some requirements are increased; other requirements may not be introduced or considered in depth. The evidence of this finding comes from our quantitative and qualitative results. For example, we have demonstrated how the public Wi-Fi had an impact on decreasing the ratings and participants would not consider other factors (e.g. connection, password) unless the network type requirements security level improves.

The methodology introduced in this work allowed us to assess security composition, however, additional work is needed to evaluate the effect of these elicited mitigations on the overall and dimension-specific risk perceptions. Next, I will explain how I extended this work and structured the process to introduce the Multi-factor Quality Measurement Method, that can be applied to investigate any quality of interest, not necessarily security.

5.3 Using the Multifactor Quality Measurement Method to Assess Security Requirements Composition

The factorial vignette-based approach introduced earlier in this chapter, uses scenarios to describe an environment that mimics reality to the security analyst to discover dependencies among requirements and elicit previously unforeseen requirements that mitigate threats. Choosing how to write natural language scenarios is challenging, because stakeholders may over-generalize their descriptions or overlook or be unaware of alternate scenarios. In security, this can result in weak security constraints that are too general, or missing constraints. Another challenge is that analysts are unclear on where to stop generating new scenarios. Hence, I will introduce the Multifactor Quality Method (MQM) [66] that aims to help requirements analysts to empirically collect system constraints in scenarios based on elicited expert preferences. This method combines quantitative statistical analysis to measure system quality with qualitative coding to extract new requirements. The method is bootstrapped with minimal analyst expertise in the domain affected by the quality area, and then guides an analyst toward selecting expert-recommended requirements to monotonically increase system quality. We report the results of applying the method to security. This include 550 requirements elicited from 69 security experts during a bootstrapping stage, and subsequent evaluation of these results in a verification stage with 45 security experts to measure the overall improvement of the new requirements. Security experts in our studies have an average of 10 years of experience. The results that we discuss at the end of this chapter, show that using our method, we can detect an increase in the security quality ratings collected in the verification stage. In this chapter I will also discuss how the proposed MQM method can help researchers and analysts to improve security requirements elicitation, analysis, and measurement.

5.3.1 The Multifactor Quality Measurement

I will now describe the Multifactor Quality Measurement (MQM) method for eliciting system constraints that affect an overall quality such as security. Earlier in this chapter, I presented an empirical evaluation of using factorial vignettes for collecting security and found it to be effective. I will show here, the technique is integrated into a framework, the MQM, that can be extended and reused outside of security. Figure 5.3 shows the different stages of the MQM. In addition, the limitations of prior work is addressed in the following way:

1. The MQM is evaluated across four security domains: networking, operating systems, databases and web applications. In the prior security requirements composition study, only one domain was evaluated (computer user surfing the web).
2. Participants are put in an expert role in the scenario (e.g. network administrator)
3. We recruit security experts from industry and government.

I will now describe each phase of the MQM using a walk-through example from one of the security domains.

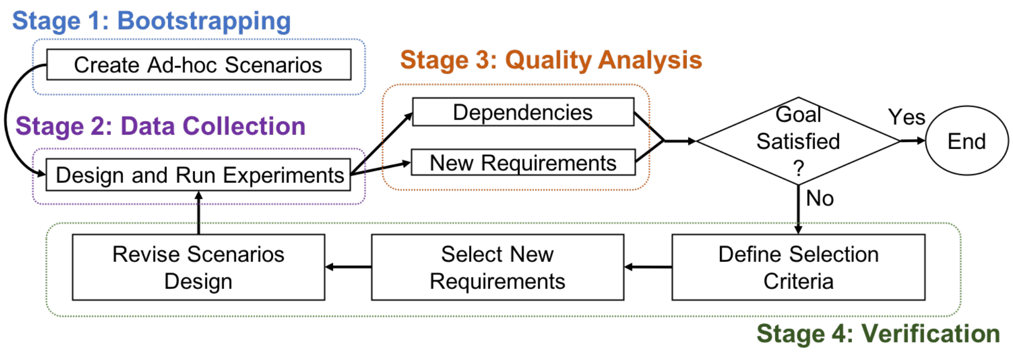


Figure 5.3: The Multifactor Quality Measurement (MQM) Method

Stage 1: Bootstrapping

During bootstrapping, an analyst first chooses the quality to evaluate (which is security in this study), and then the analyst chooses an initial scenario that describes a cohesive system viewpoint [94]. The ad hoc scenario is selected by the analyst who might have limited knowledge, because the MQM will collect empirically measured improvements in this stage. This scenario is a text-based system description that includes the ways people interact with the system. An example scenario template is shown below.

```
You are a website administrator responsible for security a web app against
 cyber attacks. Currently, you are evaluating the following settings:
- The web app performs WebAuth − ThewebappwillStoredUserData in a database for
 display to other users

The Cross-Site Request Forgery attack is a serious security concern. Please
 answer the following questions with regards to mitigating this threat.
```

The template above is from the web applications security domain that consists of variables preceded by the ($) sign. A variable in the scenario is a security requirement category. The variables are replaced by different values that correspond to constraints on the system. The manipulation of variables and their values allows the analyst to generate different instantiations of the template, called vignettes, which will increase the number of scenarios that can be evaluated at one time. The `$WebAuth` variable represents the type of authentication used in the web application and it can take one of many values. To illustrate, we consider two extremely different values: “basic authentication,” which is a weak form of web-based authentication, or “form-based authentication using encrypted credentials stored in a database,” which is stronger. Similarly, the The `$StoredUserData` variable represents how the user input is being collected, and could take the values: “collect user-supplied content from GET request,” or “require CSRF tokens and escape and validate user-supplied content from POST requests before storing;” and again, the latter value is stronger than the former.

Study participants are asked to rate the adequacy of the overall security of the scenario on a 5-point scale where point 1 is labeled "inadequate", point 3 is labeled "adequate" and point 5 is labeled "excessive." This generates the `$Overall` dependent variable. Similarly, users are asked to provide ratings for the individual security requirements in the scenario, which generates a dependent variable for each rated requirement. For example, the web applications study has the `$WebAuthRating`, and the `$StoredUserDataRating`, which are the dependent variables representing experts ratings of the `$WebAuth`, and `$StoredUserData`, respectively.

After creating the initial ad-hoc scenario, the analyst decides the number of factors and factor levels in the scenario:

1. Factors per domain: a domain could have its own subset of factors, with the possibility of having factors that are shared among different domains. The factors often correspond to categories of system constraint e.g., passwords, authentication type, etc.. In addition, factors may, but do not necessarily have to, cross multiple domains, e.g., passwords affect databases, networks, and systems.
2. Levels per factor: how many levels will be manipulated. The levels, which correspond to technically specific interpretations of the factor, can be chosen as high or low levels. The goal is to choose levels that experts can distinguish to measure an effect or interaction among different levels. For example, if password complexity has high and low levels, we can measure whether password complexity affects overall security adequacy in conjunction with other security constraints.

Deciding on the number of factors depends on the quality of interest, the cost of running the surveys, and the estimated number of experts available to rate the scenarios against the quality of interest. An analyst would need to conduct a priori statistical power analysis to decide on the right number of factor/level combinations. Initial pilot studies and focus groups can also help with the design decisions in the bootstrapping phase as it would help eliminate unrealistic factor and level combinations [68].

In addition to the web application template shown above, we describe in the following sections of this chapter how to generate more templates and integrate factors and levels for three more security domains.

Domain experts may suggest additional unforeseen requirements that would improve the measurements. An analyst could elicit new expert requirements from experts to improve the measurements. For example, security experts could provide more mitigation that would increase the adequacy ratings, so, we ask experts to list additional mitigations that they believe will increase security.

Stage 2: Data Collection

Once the scenarios are ready, the analyst finalizes the design of the overall experiment. This includes deciding which factors are between-subject or within-subject factors. The analyst in this stage decides on how to operationalize the survey: recruitment methods (e.g. in person, online, mailing lists), tools to be used, and whether expertise screening questions are needed (e.g. knowledge tests, demographics). Finally, the analyst deploys the survey and starts data collection.

Stage 3: Quality Analysis

In this stage, the analyst uses regression analysis to discover the weights of the factor levels (e.g., `$WebAuth`, and `$StoredUserData`) and to discover any interactions among the variables. The priorities of requirements are decided based on the weight of the coefficient. The type of regression (e.g. linear, multi-level) depends on the study design (within-subject vs. between-subject effect). Linear regression is used when there is no within-subjects effect in the data, while multilevel modeling is used if there is at least one within-subject factor. Next, the analyst classifies the experts new requirements into broader categories and links these to the factors/levels in the scenario. The collected new expert requirements mitigations are expressed in natural language. The problem with natural language statements is that different experts could describe the same requirement using different words and phrases. As a first step, requirements are coded using short phrases (concept labels), an open coding grounded analysis approach [68, 107]. Then, the analyst categorizes the requirements using a more abstract security concept. For example, mitigations coded as *password salt* and *stronger password*, are grouped under *passwords*; and *input sanitization* and *input validation*are categorized under *SQL injection mitigations*.

After first-cycle coding and categorization, a second-cycle coding is conducted [107], where requirements are linked to the factor levels that they appear in, which would help to filter the requirements that we anticipated to appear vs. new unanticipated requirements. For example, in the network study, there are scenarios with insecure *Dematerialized Zone* (DMZ) configuration and a more secure split-DMZ configuration. Mitigations that suggest better network segmentation are linked to the level of the DMZ level shown in scenarios where the mitigation was elicited. If associated with the weaker DMZ, then this makes the mitigation anticipated, but if associated with the stronger DMZ, then that means there are further segmentation configurations for the network and DMZ that was not anticipated in the scenario.

In addition, each requirement is assigned one of the following codes: *refinement*, if the requirement refines the dimension by extending its functionality; a *reinforcement*, if the requirement adds auxiliary quality not directly related to the dimension; and a *replacement*, if the requirement replaces the dimension.

Upon completion of analysis, the analyst decides to either stop and be satisfied with the data collected, or continue to the next stage: verification. Verification is an expensive step that the analyst could pursue if the results show rich data that needs further verification, and stop once they reach saturation. By saturation, we mean no new requirements are being collected and the analyst continues to see the same statistical results (e.g. same effect, same dependencies among the variables).

Stage 4: Verification

Based on the output of stage three, the analyst defines a selection criteria and heuristics that will guide the requirements selection process. For example, to ensure monotonically increasing quality, an analyst may only select requirements that would increase the quality of interest in the next scenarios.

In our series of security experiments, our goal is to increase security adequacy. Hence we define the following criteria:

- For each domain, select two categories from second cycle coding with the highest number of requirements within the category.
- For each category, select the requirements with highest frequency that appear even in vignettes where the level of the requirement is strong.

In the verification stage, the requirements evaluated in the bootstrapping stage are assigned a fixed level, which is the strong security level. By fixing these levels, the effect of unanticipated requirements becomes the focus of measurement.

Then, the analyst will repeat steps from stages two and three to verify whether the new set of requirements affects the quality measurements as intended. To exit the iterative process of the MQM, the analyst establishes an end goal to be achieved.

5.3.2 Experimental Evaluation of the MQM Method

I will explain below the research approach used to evaluate the MQM on security-specific domains.

Stage 1: Bootstrapping

For this stage, we select the initial security vignette that is needed to design and run a user study to collect from security ratings of security requirements from experts. We selected four different security domains and we ran four user studies one for each domain. A text template used for all four studies is shown below.

```
      A popular online retailer offers a wide variety of products for
purchase. User information in the company’s databases includes consumers’
credit card information for purchasing products in the future.
      You are a $Domain administrator for the retailer who is responsible
for securing the $Domain against cyber attacks. Currently, you are
evaluating the following settings:
      - $Factor1
      - $Factor2
      - $Factor3 ...
      The $Threat attack is a serious security concern. Please answer the
following questions with regards to mitigating this threat.
```

The values for the variables shown in the template are changed depending on the user study domain. Table 5.10 lists the variables used for security requirements in the four domains and their levels. For example, the `$Domain` is replaced with either network, systems, database, or web applications. The factors (`$Factor1`, `$Factor2`) are replaced with different sets of security requirements factors for each domain. Within a domain, the factors are manipulated with different values (levels) to generate the values for the user study corresponding to that domain.

Stage 2: Data Collection

Each vignette has a different combination of variable levels which generated 12 unique vignettes (study conditions) for each of the network ($2 \times 3 \times 2$), systems ($2 \times 2 \times 3$), and databases studies ($2 \times 3 \times 2$), and 8 vignettes for the web applications study (2×4). We selected one factor in each

Table 5.10: MQM User study security domains and their corresponding requirement variables

Domain	Threat	Factor	Level Code	Level Description
Network	Man-in-the-middle	$**NetworkAccess**	onsite	onsite access using Ethernet
			offsite	offsite External access through a secure VPN
		$NetworkAuth	simp6	a standard 6-digit password
			comp16	a 16-char password that must include an uppercase letter, lowercase letter, a symbol, and a number
			multi8	An 8-character alphanumerical password and a one-time password sent to a mobile phone
		$DMZ	allnosplit	DMZ contains the webserver, app server and the database server.
			split	DMZ contains the front-end webserver and the app server. The DB server is behind the firewall on the internal network. The app server communicates with the DB over a VPN.
Systems	Malware	$**SocialMedia**	permit	workstations permit access to social media sites
			prohibit	workstations prohibit access to social media sites
		$AdminPriviledges	noauth	prior to installing new software, employees who are local system administrators, are not required to re-authenticate
			auth	prior to installing new software, employees are required to re-authenticate
		$VirusScanner	files	workstations has programs to scan files against known malware signatures
			filesmem	workstations has programs to scan memory and files against known malware signatures
			filesmempro	workstations has programs to scan memory, files and processes against known malware signatures
Database	Privilege escalation	$**DBAccess**	extserver	user accounts and access control are handled by SQL table authentication
			sqlauth	user accounts and access control are handled by Windows Active Directory
		$DBMonitor	available	database activities are logged
			needed	database activities are logged, and inspected as needed (to examine an incident)
			month	database activities are logged, and inspected each month by a trained auditor
		$Error	user	errors are handled by notifying users who can then report the error message, as needed
			nouser	errors are handled by logging the error message with no external notification to users
Web applications	Cross-site-request forgery	$**WebAuth**	basic	basic authentication
			form	form-based authentication using encrypted credentials stored in a database
		$StoredUserData	get	store user-supplied content from GET requests
			post	store user-supplied content from POST requests
			cpost	require CSRF tokens for user-supplied content from POST requests before storing
			cespost	require CSRF tokens, escape and validate user-supplied content from POST requests before storing

Table 5.11: MQM security domains and their corresponding added requirement variables

Domain	Threat	Factor	Level Code	Level Description
Network	Man-in-the-middle	`$MFA`	`enabled`	there is a one-time password sent to a mobile phone
			`disabled`	there are no further tokens or one-time passwords sent to mobile-phones
		`$DBSegment`	`empseg`	the DB Server is placed on a special admin segment separate from the employee network
			`sepseg`	the DB Server is placed on the same segment with the employee network
Systems	Malware	`$SWInstallation`	`notest`	admins are specific IT professionals who can install any new SW with no further testing
			`test`	new software must be tested and approved prior to installation
		`$MalwareTools`	`enabled`	heuristic-based and behavioral-based malware-detection tools are enabled
			`disabled`	heuristic-based and behavioral-based malware-detection tools are disabled
Database	Privilege escalation	`$SIEM`	`siem`	a trained IT auditor inspects logs with a specialized SIEM (Security information and event management) tool that the company installed for log analysis and management.
			`nosiem`	a trained IT auditor inspects logs without the assistant of costly SIEM tool
		`$Notification`	`enabled`	admins are automatically notified when errors occur
			`disabled`	no notification sent to admins
Web applications	Cross-site-request forgery	`$InputValidation`	`client`	on the client-side
			`server`	on the client-side, followed by input sanitization on the server-side
		`$SOP`	`verify`	in addition to the CSRF token, HTTP standard headers are examined for same origin
			`noverify`	the CSRF tokens are robust. No need to verify Same Origin on the server side

study to have a within-subjects effect. This approach increases power at smaller sample sizes (security experts are scarce [70]). A participant will evaluate 4 vignettes: two domains with two vignettes in each domain. Within a domain, the two vignettes will vary by the within-subjects 2-level factor. The variables shown in bold in Table 5.10 are within-subject variables: each participant has seen all the levels of that variable; the remaining variables are between-subject variables where each participant was exposed to one level only of that variable. This yields a mixed-effect design.

Upon completion of the security ratings, participants are asked to take a security knowledge test (14 questions); and answer demographics questions (e.g. gender, age, experience, etc.). It is recommended to place background and demographics questions at the end of surveys to avoid potential bias and to increase participants response rate [105].

We targeted security experts who attended the SANSFIRE 2016 conference at Washington, DC. The SANS is a security research and education company that offers security training and certification to government and industry security analysts [108]. Each participant was compensated with a $25 Amazon gift card.

Stage 3: Quality Analysis

As mentioned above, the qualitative data was analyzed using grounded analysis open coding. I will explain below the use of multilevel modeling to analyze the quantitative data collected in stage 2.

Multilevel regression models can better handle the mixed effect in the study design (between-subject and within-subject effects) [54]. Each dependent variable generated from user ratings is analyzed using multi-level regression. For the security knowledge test, we use a `$Score` variable, which is an independent exploratory variable assigned an integer value equal to the percentage of correctly answered security questions. The tools used include: R [102] with the lme4 [13] and SJPlot [80] statistical packages, and the G*Power tool [45] for the power analysis.

Stage 4: Verification

Based on the selection criteria defined above, we select two new requirements from the reinforcement category for each security domain. The newly generated scenarios will retain the bootstrapping requirements, and include new variables for the new reinforcement requirements. Since the goal is to increase security ratings, the levels for the bootstrapping requirements were fixed at the strongest level. For the new requirements, a weak and a stronger level were used to test their effect in improving security ratings. Hence, each new study domain had a 2×2 factorial design (2 new variables with 2 new levels each). Table 5.11 shows all the added requirements and their levels. After deciding on the new requirements and the redesign on the new vignettes, we ran the user experiments using the same protocol from the bootstrapping stage, but with the following changes:

- Recruitment: we re-invited security analysts that we previously recruited for the bootstrapping stage and for other security-related studies by using the emails they provided to opt-in for future studies. We sent each participant a unique one-time code to be used to access the online survey.

- Experiment set-up: we set up the user experiment such that each participant sees one vignette from each domain, so the experiment has a between-subject design (no-mixed effects).
- Statistical analysis: since the new design is between-subject with no mixed-effect, we use linear regression for analysis

5.3.3 Results of the MQM Study

I will now present the quantitative and qualitative results from the bootstrapping and verification stages of the MQM study.

Descriptive Statistics from the Bootstrapping Stage

The bootstrapping stage aims to collect ratings and new requirements for an ad hoc vignette. In this stage, we recruited 69 security participants. Table 5.12 summarizes our sample demographics, and the participants performance on the security knowledge test. Participants have an average of 10 years of experience. The number of responses for each domain is: 39, 30, 49, and 21 for networking, operating systems, databases, and web applications, respectively (each participant was randomly assigned to two vignettes from two domains).

Dependency Analysis from the Bootstrapping Stage

The `$OverallRating` represents the experts security rating of the scenario based on the composition of the requirements. We show an example of the regression equation for the web applications domain. Equation 5.2 is our additive regression model with a random intercept ϵ grouped by participant ID.

$$\$OverallRating_webapp = \alpha + \beta_w \$WebAuth + \beta_s \$StoredUserData + \epsilon \qquad (5.2)$$

The additive model is a formula that defines the `$OverallRating` in terms of the intercept α and a series of components. Each component is multiplied by a coefficient (β) that represents the weight of that variable in the formula. The formula in Equation 5.2 is simplified as it excludes the dummy (0/1) variable coding for the reader's convenience. We use the same formula for each domain, but we replace the independent variables corresponding to the factors in that domain. We follow a similar model for the individual requirements ratings. For example, Equation 5.3 below is the additive regression model for `$WebAuthRatings variable`.

$$\$WebAuthRating_webapp = \alpha + \beta_w \$WebAuth + \beta_s \$StoredUserData + \epsilon \qquad (5.3)$$

We report the significant results of our bootstrapping stage data in Table 5.13 . We use the variable and level codes shown in Table 5.10. For each security domain, we establish a baseline level for factors in that domain. The intercept (α) is the value of the dependent variable when the independent variables are at their baseline values. The baseline levels for each domain are

Table 5.12: Bootstrapping study: demographics

Description		**Participants**	
		Number	Percentage
Gender*	Male	59	86%
	Female	7	10%
Years of Experience* (Mean=10)	Less than 2	9	13%
	2 - 5 years	15	22%
	6 - 10 years	15	22%
	11 - 15 years	9	13%
	16 - 20 years	13	19%
	more than 20 years	5	7%
Job Sector*	Industry: non-research	24	35%
	Government: non-research	22	32%
	Industry: research	5	7%
	Academia	5	7%
	Other	9	13%
Took academic classes in security		39	57%
Took job training in security		54	78%
Self-taught security knowledge		54	78%
Job Roles	Security analyst	46	67%
	Other - IT security related	6	9%
	Other - IT related	13	19%
	Other - Non IT	4	6%
Highest Degree Completed	High school or equivalent	8	12%
	Some college, no degree	7	10%
	Associate degree	5	7%
	Bachelor's degree	31	45%
	Masters graduate degree	17	25%
	PhD degree	1	<1%
Security Knowledge Score	Scored above 60%	18	26%
	Scored between 40% and 60%	40	58%
	Scored below 40%	11	16%

* A few participants did not answer this question

shown in Table 5.13. Table 5.13 also shows the coefficient estimates (*Coeff. Est.*), which show by how much the security requirement level increased or decreased the mean rating of adequacy.

For the networking domain study, we found a significant contribution of the three network factors `$NetworkAccess`, `$NetworkAuth`, and `$DMZ` for predicting the `$OverllRating_Network` ($\chi^2(7) = 11.3, p = 0.022$), over the null model (without the factors). Table 5.13 shows a significant effect from multifactor authentication for the network authentication requirement (coded `multi8`, see table 5.10), increasing the ratings over the intercept (1.83) by approximately one point (0.96) on the adequacy scale (almost adequate). Among all networking scenario requirements, only `$NetworkAuthRating` shows a significant effect ($\chi^2(4) = 18.3, p = 0.001$) (see table 5.13).

Table 5.13: Significant multilevel regression results for the boostrapping data

Dependent Variable (DV)	Independent Variable (IV) - level	Coeff. Est.	Std. Error
Networking	IVs: $NetworkAccess +$NetworkAuth +$DMZ		
baseline	*offsite + comp16 + allnosplit*		
OverallRating	*Intercept (baseline)*	1.83^{***}	0.28
	NetworkAuth-multi8	0.96^{**}	0.34
NetworkAuthRating	*Intercept (baseline)*	2.28^{***}	0.30
	NetworkAuth-multi8	0.75^{*}	0.36
	NetworkAuth-stand6	-0.72^{*}	0.36
Systems	IVs: $SocialMedia +$AdminPriviliges +$VirusScan		
baseline	*permit + auth + files*		
OverallRating	*Intercept (baseline)*	2.2^{***}	0.39
	AdminPrivileges-noauth	-0.95^{*}	0.37
SocialMediaRating	*Intercept (baseline)*	2.06^{***}	0.40
	SocialMedia-prohibit	1.13^{***}	0.19
AdminPriviligesRating	*Intercept (baseline)*	2.31^{***}	0.43
	AdminPrivileges-noauth	-1.33^{***}	0.41
VirusScanRating	*Intercept (baseline)*	2.61^{***}	0.35
	VirusScan-filesmemoryprocesses	0.89^{*}	0.37
Database	IVs: $DBAccess +$DBMonitor +$Error		
baseline	*extserver + available + nouser*		
OverallRating	*Intercept (baseline)*	2.89^{*}	0.33
*interaction term*s	Error-user	-1.35^{**}	0.45
	DBAccess-sqlauth $*$ DBMonitor-month	-0.60^{**}	0.29
	DBAccess-sqlauth $*$ DBMonitor-needed	-0.57^{*}	0.28
	DBMonitor-month $*$ Error-user	1.33^{**}	0.60
ErrorRating	*Intercept (baseline)*	2.80^{***}	0.28
	Error-user	-0.98^{***}	0.27
Web Applications	IVs: $WebAuth +$StoredUserData		
baseline	*basic + cescpost*		
OverallRating	*Intercept (baseline)*	2.36^{***}	0.21
	StoredUserData-get	-0.73^{***}	0.25
	StoredUserData-post	-1.32^{***}	0.29
	StoredUserData-cpost	-0.70^{***}	0.29
NetworkAuthRating	*Intercept (baseline)*	2.04^{***}	0.26
	WebAuth-form	0.76^{***}	0.21

$^{*}p \leq .05, ^{**}p \leq .01, ^{***}p \leq .001$

In the database domain, we see an effect for the interaction terms of the regression model for the overall security rating ($\chi^2(9) = 20.7, p = 0.01$). Reporting errors to users (Error âĂŞ user) decreased the security rating by more than a point, but when the reporting errors to users are combined with a more frequent logging mechanism (DBMonitor - month) the rating increases over the baseline.

New Requirements from the Bootstrapping Stage

Participants provided a total 550 mitigations. After text cleanup and preparation, we classified 547 mitigations into 55 categories and 187 sub-categories. Table 5.14 shows the top five categories for each domain based on number of occurrences (Freq.). The table shows how some categories appear in multiple domains (e.g. accounts/access control), while other categories were unique to a security domain (e.g. SQL injection mitigations).

Table 5.14: Top five mitigations categories

Networking		**Operating Systems**	
Category	*Frequency*	*Category*	*Frequency*
Passwords	29	Accounts/Access Control	59
Segmentation	20	Software Installation	21
Authentication	17	Social Media	17
Firewalls	6	Malware Detection	13
Certificates	6	White/Blacklisting	12
Databases		**Web Applications**	
Category	*Frequency*	*Category*	*Frequency*
Logs	74	Authentication	14
Accounts/Access Control	68	SQL Injection Mitigations	9
Error Handling	31	Web Applications Protections	9
Monitoring	10	Accounts/Access Control	4
Authentication	8	Testing	4

Descriptive Statistics from the Verification Stage

The verification stage aims to evaluate to what extent the new requirements increase security. We sent 100 email invitations, and received 45 expert responses (45% response rate). Survey Gizmo, a large online surveying platform reports that internal employee surveys receive a 30-40% response rate on average and external surveys receive an average of 10-15% [48]. Compared to the bootstrapping stage, respondents to the verification stage scored higher on the security knowledge test ($Mean_{Bootsrapping} = 52\%, Mean_{Verification} = 60\%$). Demographics of the sample from the verification stage are shown in Table 5.15

Statistical Analysis from the Verification Stage

We now review the linear regression results from the verification stage, before comparing the security ratings obtained from bootstrapping and verification.

Recall from Section 5.3.2, the MQM uses linear regression to analyze the results of the vignette surveys responses. The independent variables in the regression formula are the requirements variables shown in table 5.11 to verify the effect of the new requirements on the security ratings. We now report the regression results for each security domain.

a. **Networking**: the regression model shows that different levels of the new requirements variables `$MFA`, and `$DBSegment` do not significantly predict the overall security rating, because

Table 5.15: Verification study: demographics

Description		**Participants**	
		Number	Percentage
Gender*	Male	43	96%
	Female	1	2%
Years of Experience* (Mean=10)	Less than 2	1	2%
	2 - 5 years	14	31%
	6 - 10 years	16	36%
	11 - 15 years	8	18%
	16 - 20 years	4	9%
	more than 20 years	2	4%
Job Sector*	Industry: non-research	14	31%
	Government: non-research	12	27%
	Industry: research	2	4%
	Academia	6	13%
	Other	7	16%
Took academic classes in security		34	76%
Took job training in security		40	89%
Self-taught security knowledge		374	82%
Job Roles	Security analyst	30	67%
	Other - IT security related	4	9%
	Other - IT related	4	9%
	Other - Non IT	4	9%
Highest Degree Completed	Bachelor's degree	12	27%
	Masters graduate degree	24	53%
	High school or equivalent	2	4%
	Some college, no degree	4	9%
	Associate degree	1	2%
	PhD degree	1	2%
Security Knowledge Score	Scored above 60%	20	44%
	Scored between 40% and 60%	21	47%
	Scored below 40%	4	9%

* A few participants did not answer this question

the regression model of `$OverallRating` as a function of the `$MFA`, and `$DBSegment` did not show any significance over the intercept-only model ($F(2,39) = 1.595, p = 0.2$). Hence, the `$OverallRating` mean, which is the intercept-only model is a better predictor of the overall security ratings for the networking study. The result is similar for the regression models constructed for the `$NetworkAccessRating`, `$NetworkAuthRating`, `$DMZRating` with: ($F(2,42) = 1.2, p = 0.3$), ($F(2,42) = 0.04, p = 0.9$) and ($F(2,42) = 0.5, p = 0.6$), respectively. The `$MFA` variable that represent multifactor authentication is shown to be a good predictor of the experts `$MFARating` ($F(2,42) = 5.3, p < 0.01$). Scenarios that include multifactor authentication show an increase of 0.85 ± 0.27 (standard error) on the scale of `$MFARating` ($p < 0.001$). Similarly, the scenarios in which the database is in a separate segment (`$DBSegment`) shows a significant increase ($p < 0.001$) in the `$DBSegemtnRating` by 1.4 ± 0.30 ($F(2, 41) = 11.4, p < 0.001$).

b. **Systems**: The regression model for `$OverallRating` as a function of the `$SWInstallation` and `$MalwareTools` show significance ($F(2, 41) = 4.57, p = 0.02$) over the intercept-only model. When inspecting the coefficients, only the intercept and `$MalwareTools` show significant effects. Enabling heuristic-based and behavioral-based malware-detection tools show a significant increase ($p = 0.02$) in the `$OverallRating` by 0.53 ± 0.22 and also show a significant increase ($p < 0.001$) in the `$MalwareRating` by 1.68 ± 0.16; and thus, `$MalwareTools` is a good predictor of the `$MalwareRating` ($F(2, 42) = 61.26, p < 0.001$). The variable `$SWInstallation` is found to be good predictor of the `$SWInstallationRating` ($F(2, 42) = 35.25, p < 0.001$). Scenarios that include testing new software prior to installation (`$SWInstallation`) show a significant increase of 1.5 ± 0.18 in the participants ratings of the software installation requirement (`$SWInstallationRating`). For the regression models constructed for the `$SocialMediaRating`, `$AdminPriviligesRating`, `$VirusScanRating` variables, we found no significant effect with: ($F(2, 42) = 1.33, p = 0.3$), ($F(2, 42) = 1.63, p = 0.2$) and ($F(2, 42) = 1.45, p = 0.2$), respectively. We also found no significant effect for the interaction terms.

c. **Databases**: The regression for `$OverallRating` as a function of `$SIEM`, and `$Notification` show no significance ($F(2, 38) = 1.06, p = 0.35$) over the intercept-only model. Except for `$DBMonitorRating` and `$NotificationRating`, no significant effects are found for the requirements ratings in the database scenarios. Database scenarios that include using a specialized SIEM (security information and event management) tool, show a significant ($p = 0.009$) increase of 0.54 ± 0.20 on the `$DBMonitorRating` The `$SIEM` shows significance in predicting the `$DBMonitorRating` ($F(2, 42) = 3.8, p = 0.03$). Similarly, `$Notification` is a good predictor of the `$NotificationRating` ($F(2, 42) = 24.29, p < 0.001$). Scenarios that include notifying admins about errors show a significant ($p < 0.001$) increase of 1.48 ± 0.22 on the `$DBMonitorRating`.

d. **Web Applications**: Except for the regression model constructed for `$SOPRating`, which rates the same origin policy, no significant effects are found for the `$OverallRating` nor for all other requirements in this scenario. For the `$SOPRating`, it was not the `$SOP` variable that significantly affected this rating, but the `$InputValidation`. Scenarios that include validating the client's input on the server-side, show a significant ($p = 0.007$) increase of 0.9 ± 0.32 on the `$SOPRating`. The `$InputValidation` show significance in predicting the `$SOPRating` ($F(2, 39) = 4.03, p = 0.03$).

The major takeaway is that the intercept-only model is sufficient to explain the outcome dependent variable. The significance of the intercept-only model means that we can rely on using the means of the dependent variables to explain the observations in the data. For the security analyst, this means that varying levels of new factors did not show significance, but we cannot remove the factors from the model. We will explain this further as we show the mean values below.

Comparing the Security Ratings Between Bootstrapping and Verification Stages

Recall from above that the purpose of the verification stage is to evaluate to what extent the new requirements increase security. In Table 5.16, the mean ratings in the verification stage are higher than the bootstrapping stage, except for the overall rating for database, which has a slightly lower average than the bootstrapping stage. Table 5.16 also shows that some variables increased more than others, for example, the `$OverallRating` for Networking only increased by 0.20, while the `$NetworkAuthRating` increased 4.5 times by 0.90. Despite the ratings increase, the values in Table 5.16 also indicates that all averages are close to adequate (3.0 on the 5-point scale). The standard deviation of all the ratings ≤ 1.

Table 5.16: Comparison of experts security ratings

Rating Variable Name	**Bootstrapping Stage**	**Verification Stage**
	Mean Rating	*Mean Rating*
Networking		
OverallRating	2.37	2.57
NetworkAccessRating	2.70	3.09
NetworkAuthRating	2.32	3.22
DMZRating	2.53	2.82
Operating Systems		
OverallRating	2.10	2.70
SocialMediaRating	2.60	3.13
AdminPriviligesRating	1.74	3.07
VirusScanRating	2.73	2.80
Databases		
OverallRating	2.51	2.34
DBAccessRating	2.62	2.71
DBMonitorRating	2.56	3.00
ErrorRating	2.25	2.60
Web Applications		
OverallRating	1.80	2.62
WebAuthRating	2.05	2.69
StoredUserDataRating	1.86	3.07

5.3.4 Discussion and Conclusions from the MQM Study

The results from the MQM study show that the mean overall security ratings increased in the verification stage over the bootstrapping stage. This means that experts view the refined scenarios in the verification stage to have higher security adequacy than the original scenarios used in the bootstrapping stage. The results in Table 5.16 also indicate that the average ratings are approximately 3 ± 1 (STD) (adequate=3, see above sections). One possible explanation could be that security experts are more conservative when rating security and cannot envision excessive security. We found in earlier work that security experts do prefer more conservative security ratings [70].

The regression analysis of the verification stage also shows that the new requirements matters to the analysis, but the individual levels do not vary significantly. While in the verification stage experts report a ratings increase over the bootstrapping stage, the increase cannot be attributed to the new requirements levels. This finding yields two key insights: *security saturation*, wherein it is sufficient to accept new, elicited requirements and a verification stage may not be necessary; and *label bias*, in which the excessive label is unreachable and thus reduces the ability to measure significant differences. Below, I will further discuss these two insights.

Reaching saturation is an important point in empirical research, where analysts receive little new information and thus they can stop iterating through a process. Saturation is also important in practice, because security analysts would prefer a *wish-list* of all possible security mitigations, but it is the financial cost that forces analysts to revise and only choose what is necessary. Our results from the verification stage indicate that increasing the requirements from the bootstrapping stage to a stronger level is what is necessary to reach security adequacy. When we increased the requirements to a stronger level, the overall security increased to a point that the two new added requirements with their levels did not necessarily standout in the regression model. This is an effect caused by the combination of strong security requirements in the scenarios tested.

The security ratings that do not rise above adequate raise a question about the adequacy scale. The adequacy scale was evaluated in separate studies [65, 70] to select the appropriate language labels that explain adequacy. The evaluation examined synonyms for inadequate, adequate and excessive in four scenarios where adequacy perception is skewed by the object being evaluated (see Chapter 4 for details). Haley et al. proposed a framework to "determine adequate security requirements for a system" [61]. What has not been discovered, yet, however is whether security experts view any security requirements as excessive, or whether the nature of security unknowns inhibits experts from reaching this conclusion.

The MQM employs vignette surveys to link requirements as factors to a system quality, and to elicit expert judgements about quality levels achieved by those requirements. This is different from prior work in scenario-based requirements elicitation that employs interviews [101, 117, 122]. Although interviews provide detailed scenario descriptions, our approach allows analysts to attribute a quality level to specific requirements and their interactions. The MQM does not measure coverage, but it offers increased coverage of scenarios as it allows the manipulation of descriptions, and the measurement effects of certain requirements on the outcome as well as the dependencies between the requirements. In addition, the use of surveys make it more convenient to recruit more stakeholders, which increases the number of viewpoints of the scenario, and multiple viewpoints improve inter-personal uncertainty; which means, one expert might point out something that other experts missed while other experts find something different. This uncertainty among experts, which impacts security assessments [68, 69, 70], is due to differences in background or human memory limitations [69].

5.4 Threats to Validity

Factorial vignettes have been shown to improve both internal and external validity [1]. In this section, I will discuss threats to validity in our preliminary study and the MQM study.

Construct validity is the degree to which a measurement corresponds to the construct of interest [113]. In each scenario, we present one-sentence definitions for the security level terms inadequate, adequate, and excessive, to encourage participants to interpret the label levels, similarly. The label name choice was evaluated in separate prior studies that were explained earlier in Chapter 4.

External validity concerns how well results generalize to the population [113]. Our target population is security experts. In the preliminary study we recruited participants from senior and graduate level security classes and mailing lists. Furthermore, we conducted a security knowledge test to measure their expertise. One possible sample bias is that our sample was drawn from two U.S. Universities.

In the MQM study, we decided on diversifying the sample by including industry experts. We recruit security professionals who attend security conferences. To assess security expertise, we measured years of experience (mean=10.0 years) and we conducted a security knowledge test that included technical questions about how to configure file permissions, network firewalls, etc.

Internal validity is the degree to which a causal relationship can be inferred between the independent predictor variables and the outcome dependent variables [113]. In the preliminary study, we randomized the assignment to conditions and the order of the four vignettes shown to each participant. We also randomize the order of the 3 adequacy ratings in the overall security-rating question, and we mask the numerical values for these ratings from participants. To address the threats of learning and fatigue effects, we estimate a 20 minutes average time for each threat survey, and we maintain a time space of a week minimum between threat conditions. We did not randomize the threat scenario order, but we mitigated the effect of this decision by treating the two threats as separate datasets during analysis.

Similarly, in the MQM study, we randomize the assignment of participants to conditions, and we randomize the presentation order of scenarios. Based on our pilot results, we limited the number of vignettes shown to four vignettes per participant to reduce fatigue. We ran the verification study seven months after the bootstrapping stage to reduce learning effects.

Increasing power in user experiments reduces Type II errors (false negatives). In the preliminary study, we instrumented a mixed-models design that combines within-subjects and between-subjects effects. We also analyze our data with multi-level regression modeling which limits the biased covariance estimates by assigning a random intercept for each subject [54]. In the MQM study, we increase our power in the bootstrapping stage by using repeated measures within-subject effect, and analyzing the data with multi-level modeling, which assigns a random intercept for each subject and hence, limits the biased covariance estimates [54]. For a power of 80% or above, we estimate a sample size of 30 participants for the networking, operating systems, and database scenarios and 24 participants for the web applications scenario. We achieved higher sample sizes than these minimum estimates. For the verification phase, we estimate 30 participants per domain to achieve at least 80% power, and our actual sample size is 45 participants per domain.

5.5 Conclusions

In this chapter, I present results from a series of experiments [66, 68] conducted on security experts. I also describe how I used a mix of quantitative and qualitative analysis methods to understand how a change in security requirements can affect security experts decision-making. Our results indicate that the composition of requirements present in the scenarios affect the experts security ratings. The results also show that experts assign different priorities to different requirements; for example, the type of network has a higher priority than the password settings in the preliminary study. Because of the diverse background of experts and the uncertainty in security decisions, our qualitative data analysis shows a variety of added security mitigations suggested by these experts to increase security ratings. In the next chapter, I will explain how we used results and insights from one of these studies to inform the design of a security assessment system.

In addition to studying security expertise and decision-making, this chapter introduced the MQM method, which combines different research and analysis methods adapted from fields like social science and statistics to provide a defined framework for researchers and analysts that can be used to study any quality of interest. For example, if an analyst can use the MQM to design scenarios in the health care domain to analyze how changing certain settings can affect patients privacy-related decisions; and results of the analysis can be used to improve the organization's privacy practices.

Chapter 6

Data-driven Approach for Modeling Expert Knowledge

In this chapter, I will report results from developing a security requirements rule base that characterizes uncertainty in human expert reasoning to enable new decision-support systems. I will show how to use relevant information collected from cybersecurity experts to enable the generation of: (1) interval type-2 fuzzy sets that capture intra- and inter-expert uncertainty around vulnerability levels; and (2) fuzzy logic rules underpinning the decision-making process within the requirements analysis. The proposed method relies on comparative ratings of security requirements in the context of concrete vignettes, providing a novel, interdisciplinary approach to knowledge generation for fuzzy logic systems. The proposed approach is tested by evaluating 52 scenarios with 13 experts to compare their assessments to those of the fuzzy logic decision support system. The initial results show that the system provides reliable assessments to the security analysts, in particular, generating more conservative assessments in 19% of the test scenarios compared to the experts' ratings.

The full results of this work is discussed further in our paper published at the 2016 IEEE Symposium on Computational Intelligence in Cyber Security (CICS 2016) [70].

6.1 Motivation and Background

As already mentioned in previous chapters, the number of cybersecurity experts is scarce. The scarcity of experts and the need for cybersecurity, makes the demand for intelligent decision support and semi-automated solutions a necessity. The following summarizes the general challenges to human-based security assessments:

- Context: experts risk assessment of a system must consider the system context in which the requirements apply [68, 69].

Excerpts from this work were previously published as: H. Hibshi, T. D. Breaux, and C. Wagner, *Improving security requirements adequacy: an interval type 2 fuzzy logic security assessment system*, in 2016 IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1-8.

- Priorities: some requirements have higher priorities than others, depending on their strength in mitigating threats [68].
- Uncertainty: security risk assessment and decision-making includes a level of uncertainty [69, 104].
- Stove-piping: security expertise crosses different domains of knowledge such as hardware, software, cryptography, and operating systems [69].

The goal is not to remove the above challenges through increased decision-support. Instead, we account for these challenges by modeling human decision making with uncertainty, in a security assessment support tool based on collected data from various security experts.

6.1.1 Uncertainty in Requirements Engineering

Uncertainty is increasingly a focal point for researchers in requirements and software engineering. In architecture, Garlan argues that the human-in-the-loop, mobility, rapid evolution, and cyber physical systems are possible sources of uncertainty [53]. Esfahani and Malek identify sources of uncertainty in self-adaptive systems and they include the human-in-the-loop as a source of uncertainty [44]. In requirements engineering, Yang et. al [130] used machine learning to capture language uncertainties in speculative requirements. The approach succeeds at identifying speculative sentences, but performs weaker at identifying the scope of uncertainty when identifying specific parts of speech such as adjectives, adverbs, and nouns. The FLAGS is a goal modelling language introduced to model uncertainty in self-adaptive systems [11, 97].

Cailliau and van Lamsweerde introduce a method to encode knowledge uncertainties in probabilistic goals [18]. This method characterizes uncertainty as the probability of goal satisfaction using estimates of likelihood collected from experts. Although the authors method is sound, the reliability depends heavily on a third-party method to record expert estimates [18]. In this chapter, we contribute a novel method to elicit estimates and incorporate estimates into an IT2FLS.

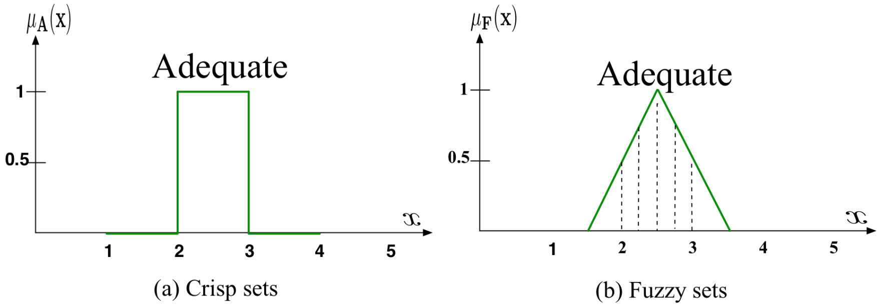
6.1.2 Uncertainty and Type2 Fuzzy Logic

Zadeh introduced Fuzzy Logic (FL) in 1965 as a mathematical tool, wherein the calculations use a degree of truth rather than simple propositions: true or false [132].

To illustrate, security experts have been shown to use the linguistic adjectives inadequate, adequate, and excessive on a 5-point semantic scale to evaluate the security of the scenarios [68]. Let X be our universe of discourse on a continuous, real-valued, inclusive scale X=[1,5] and set $A \in X$ to represent "adequate" Assume that an interval between [2,3] is adequate, as shown in Figure 6.1(a). The function $_A(x)$ is the membership function (MF) to describe A, where 1 is true and 0 is false:

$$A \Rightarrow M_A(x) = \begin{cases} 1 & 2 \leq n \leq 3 \\ 0 & oherwise \end{cases}$$

Based on the definition above, the value 1.9 for example is not adequate, because 2 is the inclusive threshold value for the adequate set, but 1.9 is very close to adequate or is adequate with a lesser degree than 1, but greater than 0. To address this concern, fuzzy set theory allows one to express to what degree a value x belongs to a (fuzzy) set A [88, 132]. Figure 6.1(b) shows how a fuzzy set F, captures *adequate*.



(a) Crisp sets

(b) Fuzzy sets

Figure 6.1: The definition of adequate in crisp and fuzzy sets

A fuzzy set F of values in X may be represented as a set of ordered pairs of the value x and its membership grade [88].

$$F = \{(x, M_F(x)|x \in X)\} \tag{6.1}$$

Type-1 MFs summarize the results of experts ratings into a single MF, suppressing the uncertainty in the data. Alternatively, Type-2 MFs model the uncertainty by providing a *footprint of uncertainty* (FOU) [88, 90]. Figure 6.2 below highlights a prototypical Type-2 MF; it is as if we blur the Type-1 MFs with the uncertainties. With the general Type-2 approach we construct MFs from

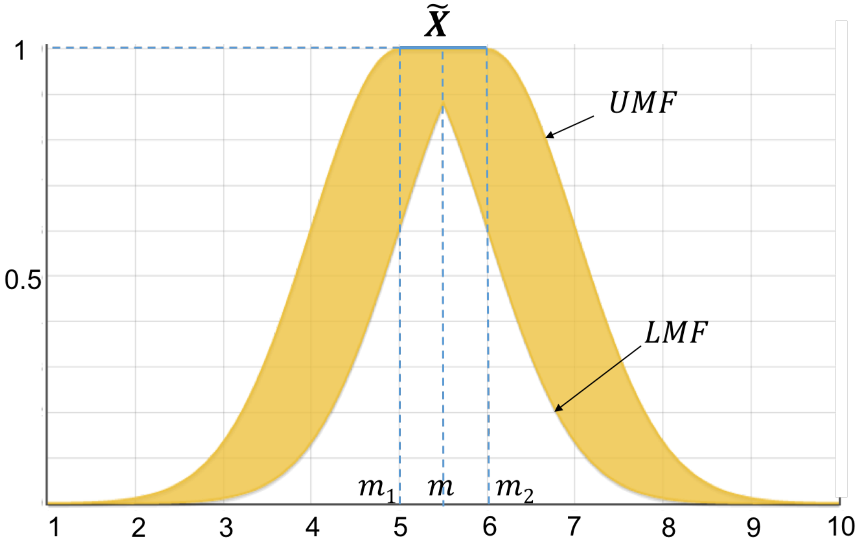


Figure 6.2: Type-2 FOU constructed by blurring a Type-1 MF

each expert's response and as if we sketch all the Type-1 MFs together to form the FOU. For each value x, there is N possible grades or MFs associated with it: $MF_1(x), MF_2(x), ...MF_N(x)$. For each of these MFs, we think of the *possibility* of this MF value for a value x by assigning a weight that represents the possibility. This is called the secondary MF (in 3-D) where at each point x, the collection of MFs and their weights is represented as: $\{(MF_I(x), w_xi), i = 1, , N\}$ [88, 89].

To make computation more feasible to the demand of applications and systems, the general Type-2 approach can be simplified by assuming uniform weights, which would result in a uniform FOU. This type of fuzzy set (FS) is called the interval Type-2 fuzzy set (IT2FS) [88, 89]. Figure 6.2 shows an example of IT2FS.

6.1.3 Interval Type 2 Fuzzy Logic Systems

Type-2 fuzzy sets are used in rule-based intelligent systems. The rule base is expressed as a collection of if-then statements and they can be collected by surveying experts in the field [89]. In the remainder of this chapter, I will show how we build a security system using an IT2FL approach. Figure 6.3 shows the main components of the proposed system. The components shown in Figure 6.3 represent what is typically found in IT2FLS [89, 90]. The components in an IT2FLS are similar to a Type-1 FLS, but with the addition of a type reducer. The type reducer reduces the inference engine's IT2FS output to an interval Type-1 fuzzy set that the defuzzifier can use to produce the final crisp output number.

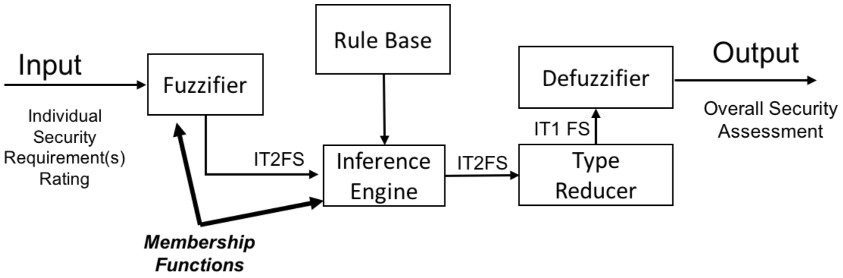


Figure 6.3: IT2FLS for Security Assessment

6.2 Overall Approach

This section explains the overall research method to build a security assessment system using IT2FLS. The contribution is two-fold:

- A comprehensive approach for developing the linguistic labels and associated membership functions for an FLS.
- An innovative approach to designing the rule base from surveys of domain experts.

Now, I will describe these two contributions.

6.2.1 Developing Linguistic Labels and Associated Fuzzy Sets

For the FSs used in the security assessment system, a decision had to be made on the appropriate linguistic labels, which are the vocabulary used in the system. The choice of labels relies on background knowledge and expertise in the field, and user surveys that support the choices made [88, 90]. Recall from Chapter 4 how three labels to describe security adequacy were developed: inadequate, adequate, and excessive. The labels were empirically evaluated because they are considered a new scale to measure our *construct of security adequacy*, and there are no existing, empirically valid scales to measure this construct (see Chapter 4 for details).

6.2.2 Eliciting the Membership Functions for the Fuzzy Sets

The membership function definition depends upon a scale assignment along an interval (e.g., from one to ten) for each word selected from our word ranking study (see Chapter 4 for details). The approach commonly accepted by the fuzzy logic research community was adopted in which experts are asked to assign the interval start and end points on one scale for each word [88, 90]. Participants were asked to specify the intervals of the 17 words from our previous ranking survey plus the word adequate (total 18 words) using the text template we show below, replacing Adequate with each of the other 17 words. We include a security scenario to add context to each word as follows:

> A security expert was asked to rate a security scenario with regards to mitigating the Man-in-the-Middle threat.
>
> The expert would give an overall security rating using a linguistic term.
>
> In the next sections of this survey, we will present 18 linguistic terms describing the overall security of a scenario. We would like you to mark an interval between 1-10 that represents each term.
>
> **Note**: Intervals for different terms can overlap.

For each word (e.g., “adequate”), participants were asked:

> Imagine "**Adequate**" represented by an interval on a range from 1-10. Where would you indicate the start and end of an "AdequateâĂİ security rating?

The word order in the survey was randomized, and we recruited participants by sending out email invitation to security mailing lists at Carnegie Mellon University. Similar to our prior work [66, 68], a test was used to assess the security knowledge of the survey participants.

Intervals were collected from 38 security experts that consists of 74% males, 18% females, and 8% unreported. The average score on the security knowledge test is 6 out of 10 possible points (sd=1.75). For each word, we calculated the average for the interval end points that we collected from participants. The results show that the three words: inadequate, adequate, and excessive are sufficient to be used as fuzzy sets covering an interval from 1-10. Figure 6.4[1]shows all the labels and the selected fuzzy sets and their coverage of the 1-10 interval. The solid region represents the interval between the mean values of the start and end points collected from the experts. The shaded region on each side of the solid region represents the standard deviation for that point, which represents the uncertainty surrounding the mean value. It is only possible to cover the entire region from 1-10 using only three labels (Inadequate, Adequate, and Excessive) because of the uncertainty that yield overlapping intervals for the three words. Mendel explains how this approach improves performance as it reduces the size of the rule base [88].

[1] we have shown this figure earlier in Chapter 4 and we repeat it here for the reader’s convenience

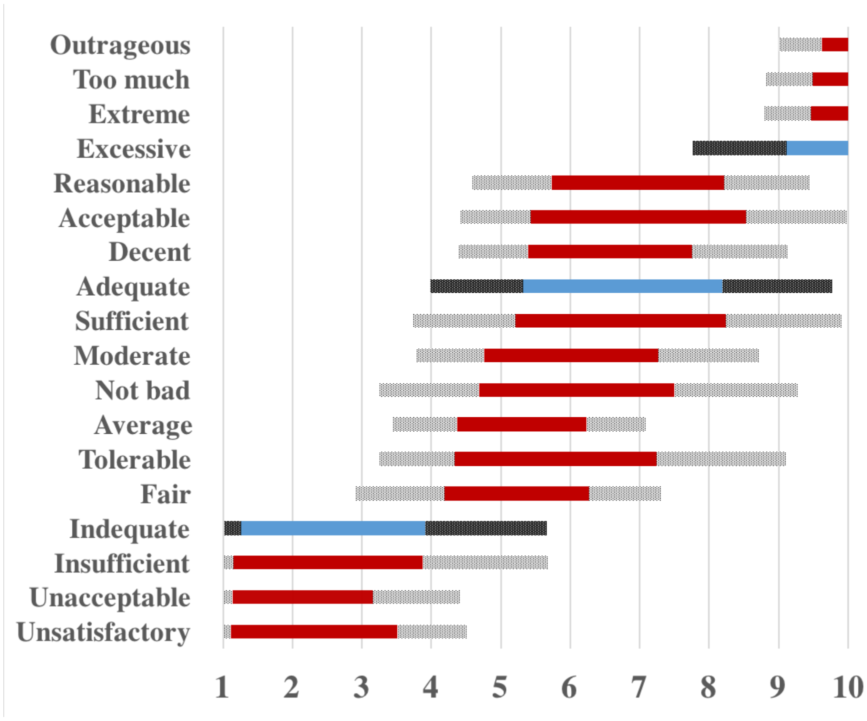


Figure 6.4: The fuzzy sets with the start and end means and standard deviation

After choosing the labels for the fuzzy sets, we now explain how to derive the MFs. We create the Type-1 MF and then blur its mean by adding a degree of uncertainty and creating the shaded region that represents the FOU. We calculate the mean for the Gaussian Type-1 MF by averaging the two end points for the interval representing each word: $Mean_{interval} = Mean_{start} + Mean_{end}/2$. Then, we average the standard deviation : $\sigma_{interval} = \sqrt{(\sigma^2_{start} + \sigma^2_{end}/2)}$ .To represent the uncertainty level surrounding the Gaussian Type-1 MF: let α represent the uncertainty level, and then calculate two means: $m1$ and $m2$ and use these for the upper and lower membership calculations: $m1 = Mean_{interval} - \alpha, andm2 = Mean_{interval} + \alpha$. We assume that we have 50% uncertainty present in our data, which makes: $\alpha = 0.5$. Table 6.1 shows the final means and standard deviations for each word label for fuzzy sets. Figure 6.5 illustrates the membership functions. We use the same MFs for the output and all the inputs: network, SSL, password, and timer.

Table 6.1: Summary statistics of the three fuzzy sets

Word	$Mean_{Interval}$	$\sigma_{Interval}$	m_1	m_2
Inadequate	2.58	1.26	2.08	3.08
Adequate	6.75	1.75	6.25	7.25
Excessive	9.50	1.35	10.00	9.00

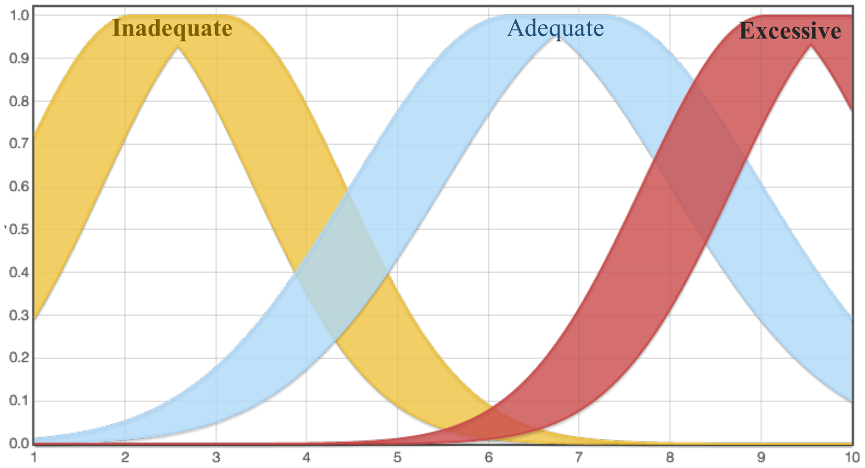


Figure 6.5: The MFs of the input/output variable(s)

6.2.3 Designing the Fuzzy Logic Assessment system

A number of researchers have built software packages and tools for IT2FLSs [19, 96, 128, 129]. Packages and tools were designed for the mathematics modeling and simulation software MATLAB, and are based on the .m files originally written by Mendel and Wu. We chose to use the Juzzy and JuzzyOnline Java-based toolkit to obtain our results, because these are open-source and actively maintained by a team of fuzzy logic researchers [124, 125]. Based on prior IT2FLSs research [129], we made the following design choices:

Input and output MF shapes: The choice of MFs is dependent upon the context of the problem and other factors, such as continuity, and computational cost. We chose to use a Gaussian shape for our MFs for it's added advantage of simplicity and faster computation time [129]. As explained in above, three membership functions were selected for each input domain: inadequate, adequate, and excessive.

Input Fuzzification: An important step in a fuzzy system is to fuzzify the input by mapping an input vector $X = (x'_1, \ldots, x'_p)$ into p fuzzy sets $X_i, i = 1, 2, \ldots, p$ [88, 129]. We choose to use the singleton fuzzifier, where: $M_{Xi}(x_i) = 1 \text{at} x_i = x'_i and : M_{X_i}(x_i) = 0$ otherwise. Singleton fuzzifiers are more practical due to their simplicity [88, 129]. The input to the system would be a number between 1-10 representing the level of the security requirements adequacy to mitigate a threat.

Rules: we construct the rules following the Mamdani style, because of its better human interpretability [88, 129]. We also chose the minimum t-norm, because we want our security assessment system to give conservative security ratings.

6.3 Ruleset Discovered from Security Experts

In this section I will explain how the user survey results from prior work [68] are translated to a rule base for the assessment system. In the user study, participants rated the overall security of scenarios using a 3-point scale: 1=excessive, 0=adequate, and -1=inadequate. Participants also rated four individual requirements-related factors in scenarios: network type, using SSL, password strength, and presence of a timer using a five-point semantic scale with: 5=excessive, 3=adequate, and 1=inadequate with the midpoints 2,4 between inadequate-adequate and adequate-excessive, respectively. Experts rated four security scenarios with four network types: employer's network, public Wi-Fi, unencrypted VPN, and encrypted VPN. Each scenario included a password, timer, and SSL requirements. The password and timer had two conditions each (either strong or weak) [68].

Based on our prior results [68], the rules were built as follows:

- The regression results for the overall security ratings indicate network type has the major significant effect, it takes priority over other requirements.
- The network rating suggests that network type can drop the overall ratings significantly with no significant effect for the other factors, hence it is safe to remove the other factors from the rule antecedents only when the network type drops to inadequate.
- When network type increases to adequate, other requirements are included as antecedents, because the statistical results show that the model with all the four factors exhibits an effect over the null model.

Next, I will show how we applied the above heuristics.

6.3.1 The Inadequate Network

The *public Wi-Fi* and the *VPN over unencrypted Wi-Fi* networks significantly dropped the overall security ratings towards inadequate (Public Wi-Fi: $Mean = -0.7$, VPN-unencrypted $Mean = -0.4$). The public Wi-Fi ratings are closer to inadequate ($Mean = 1.3$) while VPN-unencrypted ratings are in between adequate and inadequate ($Mean = 2.2$). From the above, we can infer that when the network type is definitely inadequate, then the network type has more priority in the scenario that the security levels of other requirement(s) would not matter in deciding the adequacy of the overall security of the system. Hence, we construct the following rule:

$$R^1 : IF \quad NetworkType \quad \text{is} \quad Inadequate$$

$$THEN \quad OverallRating \quad \text{is} \quad Inadequate$$

Reduction of rules in rulesets used in intelligent systems simplifies the reasoning for the human analysts interacting with the system [88]. Without the results from our user study [68], we would have 26 more rules with 26 more input combinations for the adequate network alone. To explain, we would have a four antecedent rule, wherein each input antecedent has three MFs: inadequate, adequate, and excessive. For the inadequate network alone, input combinations of the remaining three inputs (SSL, password and timer) will result in 27 rules and, if we follow a canonical approach, we would need to survey experts to obtain the consequents of all 27 rules.

However, our approach derives rules from the statistical analysis of the empirical results in which the non-significant factor levels are dropped.

6.3.2 The Adequate and Excessive Network Types

When the network adequacy level increases, then the rules for factors would change as well. The remaining network types in the study: “employer’s network” and “encrypted VPN” were rated close to adequate, but never close to excessive ($Mean = 2.6$, and $Mean = 2.9$ respectively). The overall security of the scenario was rated below adequate: ($Mean = -0.19$, and $Mean = -0.16$ respectively). This data is not sufficient to infer a rule similar to R1; i.e. we cannot use network adequacy alone in a single antecedent rule. However, as discussed previously in section 5.2, our results from the requirements composition study does show that when the network adequacy level improves, participants begin paying attention to the other factors in the scenario and their decisions become based on the composition of these other factors. The regression model for the overall security rating shows that all the factors in the scenario are predictors of the model [68]. Hence, we decide to include more input variables in our rule set antecedents. Table 6.2 below shows the antecedents and consequent combinations for the remaining rules that we constructed from our scenarios. The column *R#* is the rule number, *Antecedents* are the requirements that serve as input antecedents in the if-then rules, and *Consequence* is the consequence output that is the rating of the overall security.

Table 6.2: Rules for security assessment system

R#	Antecedents (IF)				Consequence (THEN)
	Network	*SSL*	*Password*	*Timer*	*Overall*
R1	I				I
R2	A	I			I
R3	A		I		I
R4	A			I	I
R5	A	A	A	A	A
R6	E	E	E	E	E

6.4 System Evaluation

I will explain in the section the qualitative approach for system evaluation. Qualitative methods are better suited to our system evaluation as we are looking for the participants description of the process, the rationale, and their reasoning.

6.4.1 Evaluation Process

The IT2FLS was evaluated using a two stage process: first, we survey 13 experts and have each evaluate 4 scenarios each (52 total test scenarios); and second, we conduct follow-up interviews to discuss participants’ decisions and their rationale.

The survey used in the first stage is similar to the survey in the original factorial vignettes study, which examined the interaction of a public/private network with SSL-encrypted connections, varying password strengths and an automatic, timed logout feature [68]. We modified the original design to limit the network levels to two levels that highly contrast each other: public unencrypted Wi-Fi and encrypted VPN. Furthermore, we use two levels for the password (weak/strong), and two levels for the logout timer (no timer, 15 min timer). We did not use a number of different SSL levels as this would present an obvious focal point for the participant to become concerned about security [68]. Hence, we reworded the scenario to describe SSL as follows: "The browser is already using the latest (and patched) version of SSL/TLS for the session." Participants were randomly assigned to different conditions, and we randomized the order in which they see different vignettes. Each participant rates four vignettes in total with combinations that show the two levels of each variable: network, timer, and password.

For each of the four scenarios, participants provide their overall security judgement of the scenario. Participants choose either: inadequate, adequate and excessive to evaluate the overall security adequacy without the use of any scales or numbers.

The four inputs to the IT2FLS are the adequacy ratings for the network, the SSL, the password, and the timer. We use the participants' provided ratings as inputs to our system. The output would be the overall security rating represented by a number on an interval from 1-10. After we calculate the output, we interview participants and remind them of their initial ratings including the overall security judgement of the scenario. Before showing them the output of the system, we ask them to describe the overall security ratings on a scale from 1-10, and why they rated a scenario the way they did. Then, we show the participant the output of our security assessment system in the form of fuzzy sets and we solicit their opinion. Finally, we ask participants to state what they would change in the scenario to improve the adequacy ratings, and in contrast, what would they imagine to be the worst possible change to drop the adequacy ratings further.

6.4.2 Evaluation Results

The participants' median score on the knowledge test was 7 out of 10. Three out of 13 participants work in cybersecurity at Federally Funded Research and Development Centers, one participant has 10 years of experience as a security consultant, and the remaining 9 are graduate students from Carnegie Mellon University who completed security courses and who are involved in security research.

Table 6.3 shows the participant agreement for all eight scenario combinations: the network type, the password, the logout timer, the total number of participants per scenario, and the percent agreement, which is the total number of overall ratings that match the ratings produced by the security assessment system. In Table 6.3, we see participants disagreed with the system's overall security rating predictions. We conducted follow-up interviews with nine participants. Six participants agreed with the security assessment system's overall ratings for 4/4 scenarios: however, they explained that their assessment was borderline between two rating levels. Two participants agreed with 3/4 scenarios in the system's result: in the one disagreeable scenario, both participants provided an excessive rating while the system rated the scenario as adequate. Finally, the last participant P5, who scored 7 on the security knowledge test, disagreed with the system for 4/4 scenarios, because they mistakenly believed that SSL was an adequate mitigation against man-in-the-middle attack in all scenarios, even when the network is public Wi-Fi. The participant explains: "for the purpose of man-in-the-middle, SSL is all what we need; if we worry about sniffing while in a public place, then passwords and timers are important." The participant acknowledged why the overall security could be inadequate: "If we are worried that users may not understand insecure certificates, then the VPN over an encrypted connection might provide an extra layer of security."

Table 6.3: Participant agreement with overall security

Sceanrio			**Total Participants**	**Agreement Ratio**
Network (Wi-Fi)	*Password*	*Timer*		
Public unencrypted	Weak	None	5	4/5 (80%)
Public unencrypted	Weak	15-min	8	6/8(75%)
Public unencrypted	Strong	None	8	6/8(75%)
Public unencrypted	Strong	15-min	5	3/5(60%)
VPN over encrypted	Weak	None	8	6/8(75%)
VPN over encrypted	Weak	15-min	5	2/5(40%)
VPN over encrypted	Strong	None	5	2/5(40%)
VPN over encrypted	Strong	15-min	8	4/8(50%)

The follow up interviews helped us verify the participants inputs, check for mistakes, and identify false positives. By false positives, we mean that participants could provide assessments that match the results of the system, but their reasons and priorities for security requirements did not match what the rule base had encoded. We found one false positive, participant P8, who scored 8 on the security knowledge test. Unlike P5 who disagreed with the system, P8 agreed but using a rationale similar to P5. Participant P8 mistakenly believed that SSL made the other factors less relevant, because they believe that SSL alone is sufficient to defeat man-in-the-middle attacks. The participant did not rate SSL as adequate, because they were concerned about checking the certificates and about whether or not users would trust untrusted certificates.

We asked participants: "what is the most important change in the scenario that, if it occurs, will cause you to drop your ratings?" All eight participants identified SSL, which only had one level; participants did not see stronger or weaker SSL variants, despite the existence of such variants. Participants identified requirements when weaker settings were presented: e.g., if they saw no timer, they would suggest adding a timer. This behavior was expected, because participants saw combinations where they reviewed both weak and strong settings for network, timer, and password.

We asked participants to identify requirements changes that would cause them to improve their adequacy ratings. Participant P8 indicated they would improve SSL by ensuring the server certificates are checked. The remaining six participants all suggested avoiding public unencrypted Wi-Fi and replacing it with a VPN over encrypted Wi-Fi or even better, as two participants suggested, using their own private home network. The six participants also suggested using a timer for automatic logout instead of no-timer, and using a stronger password setting instead of a plain 8-character password with no enforced character requirements. One participant suggested adding two-factor authentication to the scenario.

Our survey results reveal when participants provide different ratings to the same requirement level in two different scenarios. Eight of 13 participants provided different network ratings for the same network, but in two different scenarios. Four of eight participants clarified their choice during a follow-up interview. Three of four participants reported not remembering their previous choice, which suggests within-subject variance. The remaining participant reported providing different network ratings, because they believe that their decisions were impacted by other requirements settings, such as the timer and password.

6.5 Discussion and Conclusions from the Security Assessment System

I will now discuss the results in the presence of inter- and intra-personal uncertainties in analysts' security decisions, and explain the reliability of the security assessment system.

Interpersonal uncertainty is the uncertainty that exists between multiple analysts [88]. Security analysts, in particular, demonstrate this uncertainty by disagreeing on the same scenario [68] or artifact [69]. Our method does not rely on a single analyst's assessment: if the analyst experiences uncertainty, then judgments from other analysts would reduce the uncertainty, unless all analysts are uncertain. As shown in the previous section above, two participants P5 and P8 stated that a good SSL/TLS protocol is sufficient to defeat a man-in-the-middle attack, even if the network is public Wi-Fi. While these analysts believe that SSL/TLS is sufficient, others argue that this is insufficient over public Wi-Fi and they recommend using a secure VPN. This is an example of interpersonal uncertainty. To illustrate, if a user is connected over public Wi-Fi, and they are visiting a non-SSL website before being redirected to an SSL-enabled website, then it is easy for a malicious adversary to hijack the session and redirect the user to a website with a forged certificate. Furthermore, the attacker can use certificates signed with trusted certificates, which can cause the SSL connection to appear safe in the browser [36, 73]. Rare events and recent advances in technology illustrate the need for decision-support tools that can address limitations of human memory, such as the over- or under-estimation of risk. Cognitive psychologists argue that human memory can fail to recall relevant facts, which can be used to inform decision support models, theories and frameworks to yield intelligent systems [28]. Even the "best" expert could make mistakes and needs support with their evaluation.

Intrapersonal uncertainty is the uncertainty that one analyst experiences about a judgment [88]. In our follow-up interviews, we observed how three experts provided different ratings of the same factors, because they forgot their ratings in the prior scenario. This inability to recall allowed these participants to demonstrate uncertainty within their own ratings. Other factors that affect intrapersonal uncertainty include how representative a scenario appears, or how available the analyst's knowledge of recent events are when passing judgment [119]. In a prior study [68], the SSL Heartbleed vulnerability that affects OpenSSL had recently been announced and this event affected participants' responses about adequacy ratings for SSL [68]. Thus, surveys to collect adequacy ratings may need to be repeated to react to the evolving influences of certain events.

We choose IT2FL to build our assessment system because it handles interpersonal and intrapersonal uncertainties. As shown in our results, we interviewed nine participants in order to verify 36 test scenarios. In only six scenarios (19%), participants disagreed with the security assessment system. In all six disagreed test cases, the security assessment system was more conservative compared to the participants' ratings, i.e., the system provides inadequate for a situation that the participant believes is adequate, or adequate for a situation that the participant believes is excessive. Participant P9 commented, "in security, I prefer a conservative system's rating like that."

Rule reduction improves readability by human analysts. Earlier in this chapter we have shown how the rule base is derived from expert-ratings in factorial vignette surveys and we present heuristics to omit unnecessary inputs in the rule antecedents. However, this method has a limitation in that it does not model situations that are absent from the dataset. For example, in the scenarios that we studied, we cannot model requirements combinations that are excessive or adequate overall, because these were not present in survey data. However, this limitation can be addressed by improving the survey design using expert focus groups aimed at discovering scenarios wherein security is deemed excessive.

Fuzzy logic has been applied in multiple domains [90], including security [47, 79, 115]. Fuzzy data mining techniques using Type-1 Fuzzy Logic have been introduced in intrusion detection systems and have shown an improved outcome [47, 79, 115]. De Ru and Eloff proposed modeling risk analysis using Type-1 Fuzzy Logic and explain that modelling risk analysis with fuzzy logic produces system recommendations that are very close to real situations. They argue that without such systems, organizations run the risk of over- or under-estimating security risks [30]. In this work, we have shown how sometimes analysts underestimate the risk as our assessment systems provided more conservative ratings in 19% of the test scenarios. De Ru and Eloff's use of Type-1 Fuzzy Logic addresses vagueness, but it does not account for uncertainty and their method did not elicit security knowledge from multiple experts.

In this chapter, I have shown a new approach to build an automated security assessment system based on an Interval Type 2 Fuzzy Logic system (IT2FLS). Survey data collected from 174 security experts were used to derive the IT2FL rules, and we built membership functions based on this data. Finally, we evaluated the system by running 52 test scenarios on 13 participants. Results indicate that the system succeeds in providing a reliable assessment to analysts, although, it was more conservative in 19% of the 52 scenarios by assessing the security to be lower than our human evaluators.

Chapter 7

Conclusion and Future Research

In this thesis, I studied security expert decision-making in the presence of uncertainty and modeled this knowledge to create a human-centric intelligent solution that conforms to expert reasoning in the real-world. Throughout this research, I investigated a number of challenges that impact security decision-making and risk analysis: 1) security experts' stove-piped knowledge; 2) security requirements composition, 3) presence of uncertainty, and 4) scarcity of experts in security, which limits the volume of data collected.

I show in previous chapters how I applied mixed quantitative and qualitative methods from multiple disciplines to collect and analyze data, and how I model data collected from experts that includes interpersonal and intrapersonal uncertainties using type-2 fuzzy logic. I provide below insights and future research opportunities derived from the diversity of subtopics and research methods used in this thesis.

7.1 Improving Security Decision-Making for Novices

In Chapter 3, I discuss how novices and experts exhibit different patterns of situation awareness (SA) when analyzing security artifacts. Based on the results of this work, one can envision an adaptive security analysis system that adapts to the training needs of a security trainee based on their perception and comprehension of cues. If a trainee fails to identify a cue, then the system could provide deeper training with further cues in order to help the trainee perceive vulnerabilities, comprehend its risk, project the impact, and decide on the proper mitigation. The SA application described earlier in Chapter 3, helps to surface the cues that likely need to be supported in such a system. While experts may have little difficulties reaching projection and decision, novices may need additional information to help them reach these higher levels.

In Chapter 6, I present a proof-of-concept security assessment system that aims to model expertise using expert data collected in user studies. The idea introduced is a building block to construct more intelligent solutions for decision-support, in which rules are derived from real expert data. Advancing this idea would be to build a system that provides recommendations for novice security analysts, wherein the system educates the analyst about better design choices or security configurations that improve the security ratings of their systems.

7.2 Measuring the Effect and Priorities of Composed Requirements

I present in Chapter 5 the multi-factor quality measurement method to help study the effect of the composition of security requirements and capture requirements weights. Researchers can apply the MQM method to domains other than security where composable requirements exist. The MQM provides a defined framework for researchers and requirements engineers in academia, government and industry. A researcher who aims to study a quality of interest can create scenarios and follow the steps defined in the MQM framework. By using MQM, one can examine the dependencies among requirements and collect additional missing requirements. For example, the MQM can be applied to study the privacy requirements, and help understand the weights and priorities affecting privacy risk assessment performed by an analyst or an engineer.

The MQM process highly relies on vignette generation that would benefit from using tools for automation, wherein any analyst creates vignettes by selecting the domain of interest and factors in that domain. Systematic scenario generation is itself an ongoing research topic in requirements engineering, research in this area had focused on scenarios shown to stakeholders using a formal representation that is closer to a model [117]. Such formal representation of scenarios, has reduced readability compared to natural language scenarios [117]. Presenting scenarios using natural language improves readability, but systematically generating the scenarios is faced with challenges that include, but not limited too: 1) *consistency*, which means ensuring that each scenario has sufficient context; 2) *completeness*, which means ensuring that each scenario covers some number of factors; and 3) precision, which means ensuring that the words used are technically specific. Such challenges would be interesting to explore in future research that aims to automate the generation of vignettes.

7.3 Examining Scalability in Scenario-Based Approaches

Similar to other scenario-based approaches [117], one limitation of the MQM method is scalability. Although the MQM offers more coverage when compared to other scenario-based approaches, scalability becomes an issue as the number of scenarios, factors, and levels increase. The MQM would benefit from future research that improves scalability for scenario-based approaches. One future direction includes the use of classes of requirements in a taxonomy, in which an analyst defines criteria to select factors from different classes instead of selecting individual factors as we have done in our research. Such an approach requires the creation of a taxonomy that can define and distinguish the classes of requirements and their relationships.

7.4 Developing New Metrics

In this thesis, I explain how to develop and empirically evaluate new scales when existing metrics are unavailable or insufficient. I discuss in Chapter 4 how a new scale was developed for security adequacy and then applied in user experiments. In Chapter 5, observations from the MQM user studies indicate that security experts do not use *excessive* to describe requirements, so future studies will benefit from avoiding the use of *excessive* as an anchor point on a semantic-scale.

Rethinking and examining new metrics can improve the construct validity and the internal validity of empirical research conducted in computer science and software engineering. A beneficial research direction is to explore techniques and methodologies used in the field of psychometrics, and apply these research methods to examine new metrics. In cybersecurity specifically, it would be interesting to explore possible security metrics other than adequacy. Examples of other security metrics include, but not limited to, likelihood of an attack on a system, and a systems sensitivity to threats.

7.5 Modeling Human Experts Knowledge

In application areas where there is high dependency on human reasoning, modeling human knowledge is a necessity and a challenge. The scarcity of experts limits the amount of data collected, which introduces a challenge because machine learning, and deep learning in particular, requires large datasets. In Chapter 6, I present an application of Interval type-2 fuzzy sets to model linguistic security measures, and built a rule-base derived from quantitative and qualitative analysis of survey data. For future research, it is beneficial to study how to scale my approach and formalize the processes into a broader algorithm that can be applied to other application areas.

This thesis is limited to using fuzzy logic. Different formal modeling approaches such as, Markov Decision Processes (MDPs), and description logic fuzzy or probabilistic extensions could be examined in future research. In addition, other sources for data, such as network traffic could be explored to help build intelligent systems. In this thesis, I have surveyed experts to obtain security ratings. Other possible sources to obtain security assessment data could be through the setup of a honey-pot environment or by collecting network data that contain information about attacker and user behavior. With security experts being scarce, adding other data sources could create an opportunity to use machine learning and/or data mining methods to model human expertise.

7.6 Final Remarks

This work aims to study expert reasoning to help build and create smarter tools that help human analysts achieve their goals. As explained in earlier chapters, there is high demand for human analysts to perform the crucial task of security risk assessments, while lacking an intelligent decision-support tool that can aid the analyst in the process. This thesis highlights a number of technical challenges and explains possible approaches and research methodologies that if adopted, could become a building block towards more advanced decision-support systems that closely model real-world human reasoning.

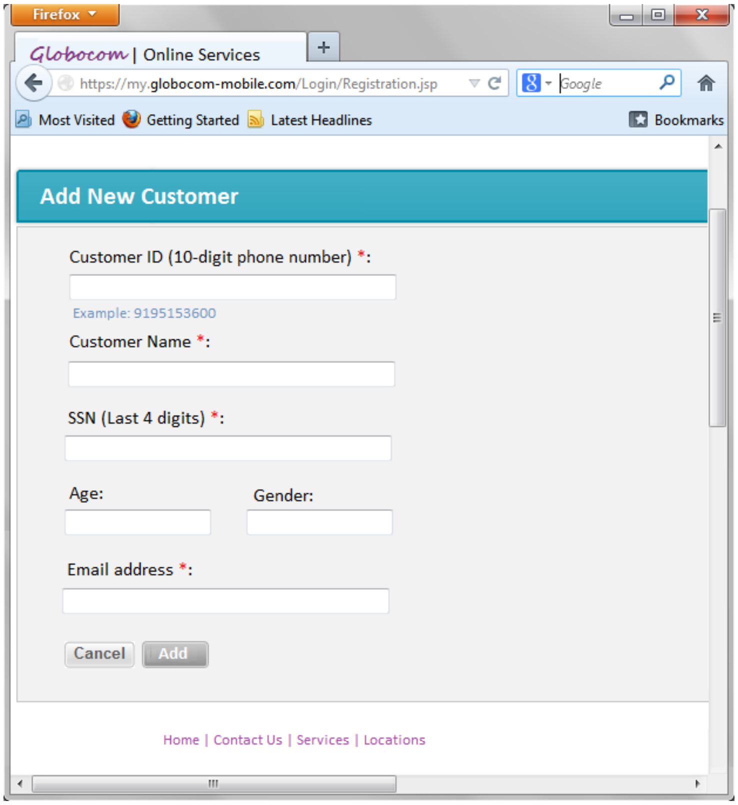
Appendix A

Study Materials for the Situation Awareness Study

A.1 Artifacts in the Situation Awareness Study

In the next pages, I will present the artifacts that were used in participants interviews in the exploratory situation awareness exploratory study mentioned in Chapter 3

Consider a global telecommunication company 'Globocom'. Web interface to add new customers is shown below.



Firefox
Globocom | Online Services
https://my.globocom-mobile.com/Login/Registration.jsp
Google
Most Visited
Getting Started
Latest Headlines
Bookmarks
Cancel
Add
Home | Contact Us | Services | Locations

Add New Customer

Customer ID (10-digit phone number) *:

Example: 9195153600

Customer Name *:

SSN (Last 4 digits) *:

Age:

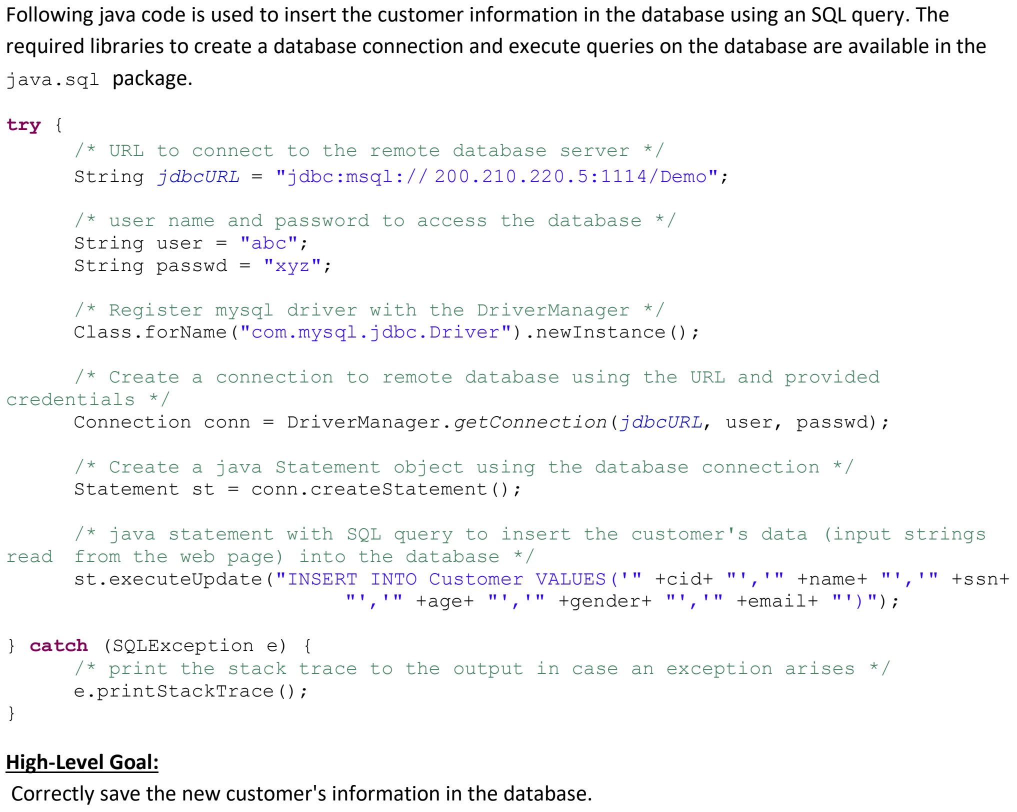
Gender:

Email address *:

Following code snippet shows how the information of the newly added customer is read in a .jsp file:

```
/* read the input strings entered in the web page */
String cid = request.getParameter("customerid");
String name = request.getParameter("name");
String ssn = request.getParameter("ssn");
String age = request.getParameter("age");
String gender = request.getParameter("gender");
String email = request.getParameter("emailaddress");
```

Figure A.1: The Source Code Artifact - Web Interface



Following java code is used to insert the customer information in the database using an SQL query. The required libraries to create a database connection and execute queries on the database are available in the `java.sql` package.

```
try {
      /* URL to connect to the remote database server */
      String jdbcURL = "jdbc:msql:// 200.210.220.5:1114/Demo";

      /* user name and password to access the database */
      String user = "abc";
      String passwd = "xyz";

      /* Register mysql driver with the DriverManager */
      Class.forName("com.mysql.jdbc.Driver").newInstance();

      /* Create a connection to remote database using the URL and provided
credentials */
      Connection conn = DriverManager.getConnection(jdbcURL, user, passwd);

      /* Create a java Statement object using the database connection */
      Statement st = conn.createStatement();

      /* java statement with SQL query to insert the customer's data (input strings
read  from the web page) into the database */
      st.executeUpdate("INSERT INTO Customer VALUES('" +cid+ "','" +name+ "','" +ssn+
                             "','" +age+ "','" +gender+ "','" +email+ "')");

} catch (SQLException e) {
      /* print the stack trace to the output in case an exception arises */
      e.printStackTrace();
}
```

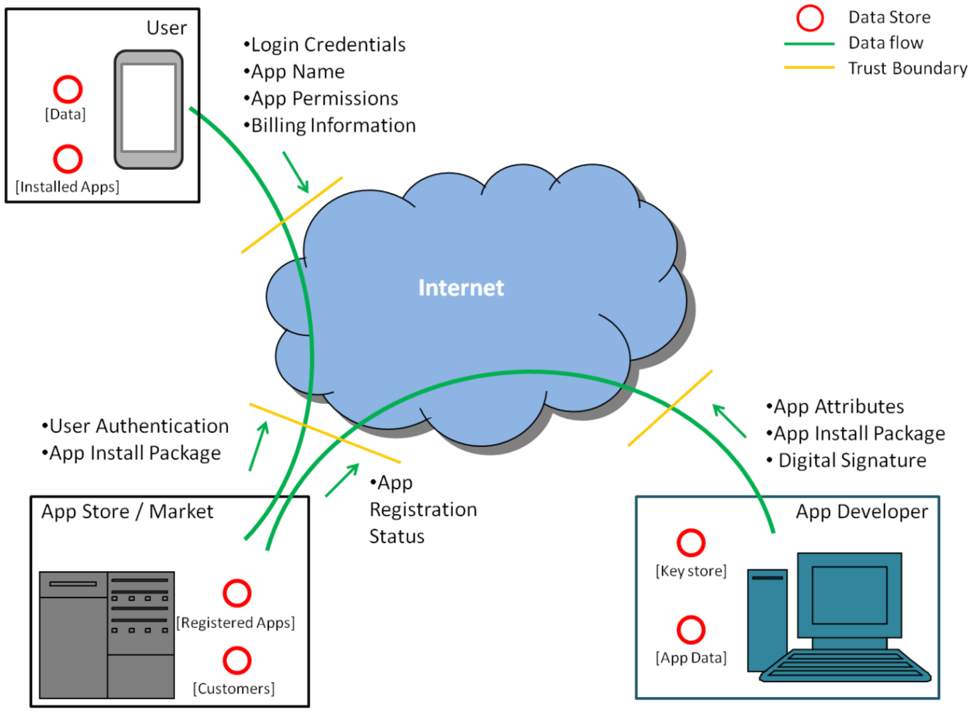
<u>High-Level Goal:</u>

Correctly save the new customer's information in the database.

Figure A.2: The Source Code Artifact - code

Data Flow Diagram

Consider the following data flow diagram (DFD) for installing an application on a mobile platform.



High-Level Goal:

Ensure secure information flows across trust boundaries.

Figure A.3: The Data Flow Diagram Artifact

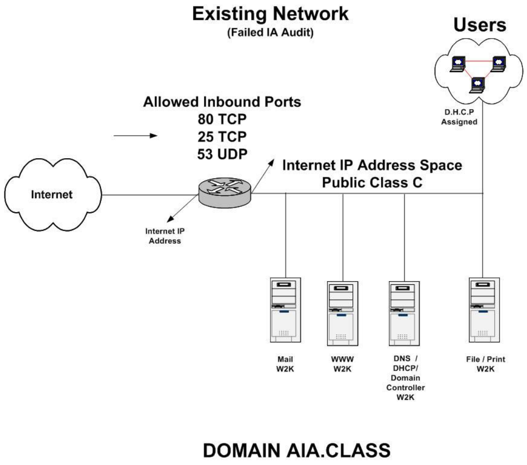


Figure 1: Existing Network

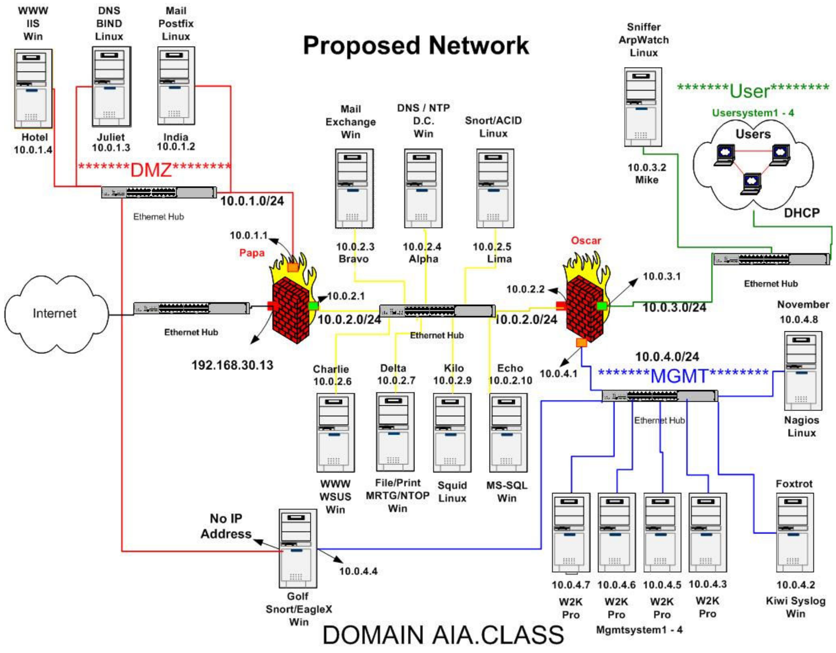


Figure 2: Proposed Network

2

Figure A.4: The Network Diagram Artifact

A.2 List of Requirements Used in Artifact ND2

R1. Company X's network, with the exception of the publicly available services which will reside in a demilitarized zone (DMZ), will be unavailable for connections initiated from the Internet to Company XâĂŹs network

R2. The employees of Company X will be required to use a web proxy server for connections to the World Wide Web.

R3. Company X will harden and secure the services and operating systems of critical systems

R4. Company X will implement web content filtering and shall block inappropriate (pornographic) web sites

R5. Company X will implement a Windows domain, and will manage server and user system configurations through group policy centrally on the network

R6. Company X will implement a electronic mail relay, relaying mail from the Internet through a mail filter, which will filter spam and malware as mail enters Company XâĂŹs network.

R7. Company X will require strong passwords (8 characters with complexity) for all user accounts.

R8. Company X will implement multiple networks (management, user, data center), and will implement strict access controls between each network.

R9. Company X will deploy system logging capabilities at all critical systems and will gather the logs centrally for review and response

R10. Company X will implement system time synchronization on the network for logging and auditing capabilities.

R11. Company X will implement multiple Intrusion Detection Systems (IDS) in multiple places on the network and shall audit regularly

a. File System Integrity IDS sensors shall be implemented

b. Network packet pattern matching IDS sensors shall be implemented.

R12. Company X shall implement split Domain Name System (DNS) services.

R13. Company X will monitor network traffic with packet sniffers.

R14. Company X will implement centralized system/service availability monitoring.

R15. Company X will administer all systems either interactively from the console or remotely from an isolated management network.

Appendix B

Examples of Security Knowledge Questions Used in Expert Surveys

I will provide below some example questions used in our security knowledge tests that we conducted in a number of studies [66, 68, 70]. I do not list the full test as we continue to reuse some of the questions in our ongoing research. The full list of questions, however, is available to share with other interested researchers upon request.

- Which of the following is considered a good encryption algorithm for encrypting files on your hard disk:
  - SSL
  - PGP
  - SHA256
  - MD5
  - TLS
  - AES
  - DES
- From the following list, choose the most secure algorithm for hashing:
  - SSL
  - PGP
  - SHA256
  - MD5
  - TLS
  - AES
  - DES

Bibliography

[1] Herman Aguinis and Kyle J. Bradley. Best practice recommendations for designing and implementing experimental vignette methodology studies. *Organizational Research Methods*, 17(4):351–371, 2014. 5.1, 5.1.1, 5.4

[2] Mohammad Salim Ahmed, Ehab Al-Shaer, and Latifur Khan. A novel quantitative approach for measuring network security. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 1957–1965, April 2008. 2.2

[3] Cheryl S. Alexander and Henry Jay Becker. The use of vignettes in survey research. *Public opinion quarterly*, 42(1):93–104, 1978. 5.1

[4] Christopher Alexander. *A pattern language: Towns, buildings, construction*. Oxford university press, 1977. 2.1

[5] Abdulrhman Alkhanifer and Stephanie Ludi. Towards a situation awareness design to improve visually impaired orientation in unfamiliar buildings: Requirements elicitation study. In *2014 IEEE 22nd International Requirements Engineering Conference, (RE '14)*, pages 23–32, Aug 2014. 3.2.1

[6] James A. Anderson. Cognitive styles and multicultural populations. *Journal of Teacher Education*, 39(1):2–9, 1988. 3.2, 3.8

[7] John Robert Anderson. *Learning and memory*, volume 86. John Wiley & Sons Inc., 2000. 3.2

[8] Arvind Arasu, Surajit Chaudhuri, Kris Ganjam, and Raghav Kaushik. Incorporating string transformations in record matching. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pages 1231–1234. ACM, 2008. 3.3.3

[9] Scott Atran, Douglas L. Medin, and Norbert O. Ross. The cultural mind: Environmental decision making and cultural modeling within and across populations. *Psychological review*, 112(4):744, 2005. 3.4

[10] Katrin Auspurg and Thomas Hinz. *Factorial survey experiments*, volume 175. SAGE Publications, 2014. 5.1.1

[11] Luciano Baresi, Liliana Pasquale, and Paola Spoletini. Fuzzy goals for requirements-driven adaptation. In *2010 18th IEEE International Requirements Engineering Conference (RE' 10)*, pages 125–134, Sept 2010. 6.1.1

[12] Frederic C Bartlett and Cyril Burt. Remembering: A study in experimental and social

psychology. *British Journal of Educational Psychology*, 3(2):187–192, 1932. 3.2

[13] Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, and Bin Dai. lme4: Linear mixed-effects models using Eigen and S4, July 2014. `http://cran.r-project.org/web/packages/lme4/index.html`. 5.2.1, 5.3.2

[14] Noam Ben-Asher and Cleotilde Gonzalez. Effects of cyber security knowledge on attack detection. *Computers in Human Behavior*, 48:51–61, July 2015. 2.3.1, 2.3.2

[15] Jaspreet Bhatia, Travis D. Breaux, Liora Friedberg, Hanan Hibshi, and Daniel Smullen. Privacy risk in cybersecurity data sharing. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security (WISCS '16)*, pages 57–64. ACM, 2016. 5.1.1

[16] Jaspreet Bhatia, Travis. D. Breaux, Joel R. Reidenberg, and Thomas B. Norton. A theory of vagueness and privacy risk perception. In *2016 IEEE 24th International Requirements Engineering Conference, (RE '16)*, pages 26–35, Sept 2016. 5.1.1

[17] Shawn A Butler and Paul Fischbeck. Multi-attribute risk assessment. In *Symposium on Requirements Engineering for Information Security*, 2002. 2.2

[18] Antoine Cailliau and Axel van Lamsweerde. Handling knowledge uncertainty in risk-based requirements engineering. In *2015 IEEE 23rd International Requirements Engineering Conference, (RE' 15)*, pages 106–115. IEEE, 2015. 6.1.1

[19] Oscar Castillo, Patricia Melin, and Juan R. Castro. Computational intelligence software for interval type-2 fuzzy logic. *Computer Applications in Engineering Education*, 21(4): 737–747, 2013. 6.2.3

[20] P. C. Chen, P. Liu, J. Yen, and T. Mullen. Experience-based cyber situation recognition using relaxable logic patterns. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 243–250, March 2012. 3.2.1, 3.2.1

[21] Thomas M. Chen and Saeed Abu-Nimeh. Lessons from stuxnet. *Computer*, 44(4):91–93, 2011. 2.3.2

[22] Michelene TH Chi. Two approaches to the study of experts' characteristics. *The Cambridge Handbook of Expertise and Expert Performance*, pages 21–30, 2006. 2.3.1

[23] Incheol Choi, Richard E. Nisbett, and Ara Norenzayan. Causal attribution across cultures: Variation and universality. *Psychological bulletin*, 125(1):47–63, 1999. 3.8

[24] Niteesh K Choudhry, Robert H Fletcher, and Stephen B Soumerai. Systematic review: The relationship between clinical experience and quality of health care. *Annals of Internal medicine*, 142(4):260–273, 2005. 2.3.1

[25] Lawrence Chung. Dealing with security requirements during the development of information systems. In *Advanced Information Systems Engineering*, pages 234–251. Springer, 1993. 2.2

[26] Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968. 3.4.2

[27] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. L. Erlbaum Associates, 1988. 5.2.1

[28] Alan F. Collins. *Theories of memory*. Psychology Press, 1993. 6.5

[29] Juliet Corbin and Anselm Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. SAGE Publications, 2007. 3, 3.3, 3.3.3, 3.3.3, 5.2.1

[30] Willem G. De Ru and Jan HP Eloff. Risk analysis modelling with the use of fuzzy logic. *Computers & Security*, 15(3):239–248, 1996. 6.5

[31] Peter DeGrace and Leslie Hulet Stahl. *Wicked problems, righteous solutions*. Yourdon Press, 1990. 2

[32] Giusj Digioia and Stefano Panzieri. INFUSION: A system for situation and threat assessment in current and foreseen scenarios. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA '12)*, pages 316–323, March 2012. 3.2.1

[33] Allen H. Dutoit, Raymond McCall, Ivan Mistrik, and Barbara Paech. Rationale management in software engineering: Concepts and techniques. In *Rationale management in software engineering*, pages 1–48. Springer, 2006. 2

[34] Varun Dutt, Young-Suk Ahn, and Cleotilde Gonzalez. Cyber situation awareness: Modeling the security analyst in a cyber-attack scenario through instance-based learning. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 280–292. Springer, 2011. 3.2.1, 3.2.1

[35] Ward Edwards and Amos Tversky. *Decision making: Selected readings*, volume 8. Penguin Books, 1967. 2.3.1

[36] Wassim El-Hajj. The most recent SSL security attacks: Origins, implementation, evaluation, and suggested countermeasures. *Security and Communication Networks*, 5(1): 113–124, 2012. 6.5

[37] Andy Ellis. SSL is dead, long live TLS - The Akamai Blog, October 2014. `https://blogs.akamai.com/2014/10/ssl-is-dead-long-live-tls.html`. 5.2.2

[38] Pardis Emami-Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujo Bauer, Lorrie Faith Cranor, and Norman Sadeh. Privacy expectations and preferences in an IoT world. In *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security (SOUPS' 17)*, pages 399–412, Berkeley, CA, USA, 2017. USENIX Association. 5.1.1

[39] Mica R Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 32, pages 97–101. SAGE Publications, 1988. 3, 3.2

[40] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995. 3.2

[41] Mica R Endsley and Debra G Jones. *Designing for situation awareness: An approach to*

user-centered design. Taylor & Francis US, 2003. 3, 3.2, 3.2.1, 3.3.2, 3.3.3, 3.3.4, 3.6.1, 3.9.3

[42] K. Anders Ericsson and Andreas C. Lehmann. Expert and exceptional performance: Evidence of maximal adaptation to task constraints. *Annual review of psychology*, 47(1): 273–305, 1996. 2.3.1

[43] K. Anders Ericsson, Neil Charness, Paul J. Feltovich, and Robert R. Hoffman. *The Cambridge handbook of expertise and expert performance*. Cambridge University Press, 2006. 2.3.1, 3.6

[44] Naeem Esfahani and Sam Malek. Uncertainty in self-adaptive software systems. In *Software Engineering for Self-Adaptive Systems II*, pages 214–238. Springer, 2013. 6.1.1

[45] Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. G* Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods*, 39(2):175–191, 2007. 5.2.1, 5.3.2

[46] Yu-Hong Feng, Teck-Hou Teng, and Ah-Hwee Tan. Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, 36(1):455–463, January 2009. 3.2.1

[47] German Florez, Susan M Bridges, and Rayford B Vaughn. An improved algorithm for fuzzy data mining for intrusion detection. In *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings (NAFIPS-FLINT 2002)*, pages 457–462, June 2002. 6.5

[48] Andrea Fryrear. What's a good survey response rate?, July 2015 (accessed February 16, 2017). `https://www.surveygizmo.com/survey-blog/survey-response-rates/`. 5.3.3

[49] Mike Furr. *Scale construction and psychometrics for social and personality psychology*. SAGE Publications, 2011. 4

[50] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: Elements of reusable object-oriented software*. Pearson Education, 1994. 2.1

[51] Simson Garfinkel. *Design principles and patterns for computer systems that are simultaneously secure and usable*. PhD thesis, Massachusetts Institute of Technology, 2005. 2.1, 2.2

[52] Simson L. Garfinkel. The cybersecurity risk. *Commun. ACM*, 55(6):29–32, June 2012. 2.2, 2.3.2

[53] David Garlan. Software engineering in an uncertain world. In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, pages 125–128. ACM, 2010. 2.3.3, 6.1.1

[54] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, 2006. 5.2.1, 5.3.2, 5.4

[55] Barney G. Glaser and Anselm L. Strauss. *The discovery of grounded theory: Strategies for qualitative research*. Aldine Pub. Co., 1967. 3, 3.4, 5.2.1

[56] Miguel Alberto Gomez and Eula Bianca Villar. Fear, uncertainty, and dread: Cognitive heuristics and cyber threats. *Politics and Governance*, 6(2):61–72, June 2018. 5.1.1

[57] Cleotilde Gonzalez, Noam Ben-Asher, Alessandro Oltramari, and Christian Lebiere. Cognition and technology. In *Cyber defense and situational awareness*, pages 93–117. Springer, 2014. 3.2.1, 3.2.1

[58] John R. Goodall, Wayne G. Lutters, and Anita Komlodi. I know my network: Collaboration and expertise in intrusion detection. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*, pages 342–345. ACM, 2004. 2.3.1

[59] John R. Goodall, Wayne G. Lutters, and Anita Komlodi. Developing expertise for network intrusion detection. *Information Technology & People*, 22(2):92–108, 2009. 2.3.1

[60] Greg Guest, Arwen Bunce, and Laura Johnson. How many interviews are enough? An experiment with data saturation and variability. *Field methods*, 18(1):59–82, 2006. 3.4

[61] Charles Haley, Robin Laney, Jonathan Moffett, and Bashar Nuseibeh. Security requirements engineering: A framework for representation and analysis. *IEEE Transactions on Software Engineering (TSE)*, 34(1):133–153, 2008. 2.2, 5.3.4

[62] Charles B. Haley, Robin C. Laney, Bashar Nuseibeh, and W. Hall. Validating security requirements using structured toulmin-style argumentation. Technical Report, Department of Computing, The Open University, Milton Keynes, UK, 2005. 2

[63] Charles B. Haley, Robin C. Laney, Jonathan D. Moffett, and Bashar Nuseibeh. Using trust assumptions with security requirements. *Requirements Engineering*, 11(2):138–151, 2006. 3.9.3

[64] Jefferson B. Hardee, Ryan West, and Christopher B. Mayhorn. To download or not to download: An examination of computer security decision making. *Interactions*, 13(3): 32–37, May 2006. 5.1.1

[65] Hanan Hibshi and Travis D. Breaux. Evaluation of Linguistic Labels Used in Applications. Technical Report, Carnegie Mellon University, 2016. 4.3, 5.3.4

[66] Hanan Hibshi and Travis D. Breaux. Reinforcing security requirements with multifactor quality measurement. In *2017 IEEE 25th International Requirements Engineering Conference (RE'17)*, pages 144–153, Sept 2017. 5, 5.3, 5.5, 6.2.2, B

[67] Hanan Hibshi, Travis D. Breaux, Maria Riaz, and Laurie Williams. Towards a framework to measure security expertise in requirements analysis. In *2014 IEEE 1st International Workshop on Evolving Security and Privacy Requirements Engineering (ESPRE)*, pages 13–18, Aug 2014. 1

[68] Hanan Hibshi, Travis D. Breaux, and Stephen. B. Broomell. Assessment of risk perception in security requirements composition. In *2015 IEEE 23rd International Requirements Engineering Conference, (RE'15)*, pages 146–155, Aug 2015. 1, 2.3.2, 5, 5.3.1, 5.3.1, 5.3.4, 5.5, 6.1, 6.1.2, 6.2.2, 6.3, 6.3.1, 6.3.2, 6.4.1, 6.5, B

[69] Hanan Hibshi, Travis D. Breaux, Maria Riaz, and Laurie Williams. A grounded analysis of experts' decision-making during security assessments. *Journal of Cybersecurity*, 2(2): 147–163, 2016. 1, 3, 3.3, 3.10, 5.3.4, 6.1, 6.5

[70] Hanan Hibshi, Travis D. Breaux, and Christian Wagner. Improving security requirements adequacy: An interval type 2 fuzzy logic security assessment system. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, Dec 2016. 5.3.2, 5.3.4, 6, B

[71] Douglas L Hintzman. "Schema Abstraction" in a multiple-trace memory model. *Psychological review*, 93(4):411–428, 1986. 3.2

[72] John Homer, Xinming Ou, and David Schmidt. A sound and practical approach to quantifying security risk in enterprise networks. Technical report, Kansas State University, 2009. 2.2

[73] Lin Shung Huang, Alex Rice, Erling Ellingsen, and Collin Jackson. Analyzing forged SSL certificates in the wild. In *2014 IEEE Symposium on Security and Privacy*, pages 83–97. IEEE, May 2014. 6.5

[74] G. Jakobson. Using federated adaptable multi-agent systems in achieving cyber attack tolerant missions. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pages 96–102, March 2012. 3.2.1, 3.2.1

[75] Guillermina Jasso. Factorial survey methods for studying beliefs and judgments. *Sociological Methods & Research*, 34(3):334–423, 2006. 5.2.1

[76] Allen C Johnston, Merrill Warkentin, Maranda McBride, and Lemuria Carter. Dispositional and situational factors: Influences on information security policy violations. *European Journal of Information Systems*, 25(3):231–251, 2016. 5.1.1

[77] Justin Kruger and David Dunning. Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, 77(6):1121–1134, 1999. 2.3.1, 3.9.3

[78] K. Labunets, F. Massacci, F. Paci, and Le Minh Sang Tran. An Experimental Comparison of Two Risk-Based Security Methods. In *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 163–172, October 2013. 2.2

[79] Jianxiong Luo and Susan M. Bridges. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems*, 15(8):687–703, 2000. 6.5

[80] Daniel LÃijdecke. sjPlot: Data visualization for statistics in social science, 2015. R package. Available at `https://CRAN.R-project.org/package=sjPlot`. 5.3.2

[81] Marshall Honorof. SSL vs. TLS: The future of data encryption, September 2013 (accessed March 8, 2015). `http://www.tomsguide.com/us/ssl-vs-tls,news-17508.html`. 5.2.2

[82] Kirsten Martin. Do privacy notices matter? Comparing the impact of violating formal privacy notices and informal privacy norms on consumer trust online. *The Journal of Legal Studies*, 45(S2):S191–S215, 2016. 5.1.1

[83] Kirsten Martin and Helen Nissenbaum. Measuring privacy: An empirical test using context to expose confounding variables. *Colum. Sci. & Tech. L. Rev.*, 18:176, 2016. 5.1.1

[84] Kirsten E. Martin. Diminished or just different? A factorial vignette study of privacy as a social contract. *Journal of Business Ethics*, 111(4):519–539, December 2012. 5.1.1

[85] Christopher May. Applied information assurance, June 2016 (accessed June 23, 2016). `https://www.andrew.cmu.edu/course/14-761/`. 3.3.1

[86] Maranda McBride, Lemuria Carter, and Merrill Warkentin. Exploring the role of individual employee characteristics and personality on employee compliance with cybersecurity policies. *RTI International-Institute for Homeland Security Solutions*, 2012. 5.1.1

[87] Alexander McKelvie, J Michael Haynie, and Veronica Gustavsson. Unpacking the uncertainty construct: Implications for entrepreneurial action. *Journal of Business Venturing*, 26(3):273–292, 2011. 5.1.1

[88] Jerry M. Mendel. *Uncertain rule-based fuzzy logic systems: Introduction and new directions*. Prentice Hall PTR,, 2001. 2.3.3, 4.1.1, 4.1.2, 6.1.2, 6.1.2, 6.1.2, 6.1.2, 6.2.1, 6.2.2, 6.2.2, 6.2.3, 6.2.3, 6.3.1, 6.5

[89] Jerry M. Mendel. Type-2 fuzzy sets and systems: An overview. *IEEE Computational Intelligence Magazine*, 2(1):20–29, February 2007. 6.1.2, 6.1.3

[90] Jerry M. Mendel and Dongrui Wu. *Perceptual computing: Aiding people in making subjective judgments*, volume 13. John Wiley & Sons, 2010. 4.1.1, 4.1.2, 6.1.2, 6.1.3, 6.2.1, 6.2.2, 6.5

[91] Charles I Mosier. A critical examination of the concepts of face validity. *Educational and Psychological Measurement*, 1947. 4

[92] John Mylopoulos, Lawrence Chung, and Brian Nixon. Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on software engineering (TSE)*, 18(6):483–497, 1992. 2.2

[93] NIST. NIST/ITL Special Publication (800), January 2015. Available at: `http://www.itl.nist.gov/lab/specpubs/sp800.htm`. 1, 2.1, 3.2

[94] Bashar Nuseibeh, Jeff Kramer, and Anthony Finkelstein. A framework for expressing the relationships between multiple views in requirements specification. *IEEE Transactions on Software Engineering (TSE)*, 20(10):760–773, Oct 1994. 5.3.1

[95] OWASP. OWASP Top Ten Project - OWASP, October 2014. Available at: `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`. 1, 2.1

[96] Muzeyyen Bulut Ozek and Zuhtu Hakan Akpolat. A software tool: Type-2 fuzzy logic toolbox. *Computer Applications in Engineering Education*, 16(2):137–146, 2008. 6.2.3

[97] Liliana Pasquale and Paola Spoletini. Monitoring fuzzy temporal requirements for service compositions: Motivations, challenges and experimental results. In *2011 Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems*, pages 63–69, Aug 2011. 6.1.1

[98] Celeste Lyn Paul and Kirsten Whitley. A taxonomy of cyber awareness questions for the user-centered design of cyber situation awareness. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 145–154. Springer, 2013. 3.2.1, 3.2.1

[99] Kaiping Peng and Richard E. Nisbett. Culture, dialectics, and reasoning about contradiction. *American Psychologist*, 54(9):741, 1999. 3.8

[100] Bruce Potter and Gary McGraw. Software security testing. *Security & Privacy, IEEE*, 2 (5):81–85, 2004. 3.6.2

[101] Colin Potts, Kenji Takahashi, and Annie I. Anton. Inquiry-based requirements analysis. *IEEE software*, 11(2):21–32, 1994. 5.3.4

[102] R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, 2013. Available at: http://www.R-project.org/. 5.2.1, 5.3.2

[103] Ashwini Rao, Hanan Hibshi, Travis Breaux, Jean-Michel Lehker, and Jianwei Niu. Less is more?: Investigating the role of examples in security studies using analogical transfer. In *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security*, page 7. ACM, 2014. 3.2

[104] Loren Paul Rees, Jason K. Deane, Terry R. Rakes, and Wade H. Baker. Decision support for Cybersecurity risk planning. *Decision Support Systems*, 51(3):493–505, 2011. 6.1

[105] Michael T Roberson and Eric Sundstrom. Questionnaire design, return rates, and response favorableness in an employee attitude questionnaire. *Journal of Applied Psychology*, 75 (3):354, 1990. 4.1.2, 5.2.1, 5.3.2

[106] Peter Henry Rossi and Steven L. Nock. *Measuring social judgments: The factorial survey approach*. SAGE Publications, April 1982. 5.1, 5.2.1

[107] Johnny Saldaña. *The coding manual for qualitative researchers*. SAGE Publications, 2nd ed. edition, 2013. 3.3, 3.3.3, 5.2.1, 5.2.1, 5.3.1

[108] SANS. SANS institute: About, February 2017 (accessed February 4, 2017). Available at: https://www.sans.org/about/. 5.3.2

[109] Yukiko Sawaya, Mahmood Sharif, Nicolas Christin, Ayumu Kubota, Akihiro Nakarai, and Akira Yamada. Self-confidence trumps knowledge: A cross-cultural study of security behavior. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*, pages 2202–2214. ACM, 2017. 2.3.1

[110] Kristin E. Schaefer, Deborah R. Billings, and Peter A. Hancock. Robots vs. machines: Identifying user perceptions and classifications. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pages 138–141, March 2012. 3.2.1

[111] Frank L. Schmidt and John E. Hunter. Tacit knowledge, practical intelligence, general mental ability, and job knowledge. *Current Directions in Psychological Science*, 2(1):8,9, 1993. 2.3.1

[112] Ariha Setalvad. Demand to fill cybersecurity jobs booming, March 2015 (accessed March 8, 2016). http://peninsulapress.com/2015/03/31/cybersecurity-jobs-growth/. 1

[113] William R Shadish, Thomas D Cook, and Donald T Campbell. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin, 2002. 5.4, 5.4,

5.4

[114] Mary Shaw and David Garlan. *Software architecture: Perspectives on an emerging discipline*, volume 1. Prentice Hall Englewood Cliffs, 1996. 2.1

[115] George Brannon Smith and Susan M. Bridges. Fuzzy spatial data mining. In *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings (NAFIPS-FLINT 2002)*, pages 184–189, June 2002. 6.5

[116] Gary Stoneburner, Alice Y. Goguen, and Alexis Feringa. SP 800-30. Risk Management Guide for Information Technology Systems. Technical report, National Institute of Standards & Technology, 2002. 2.2

[117] Alistair Sutcliffe. Scenario-based requirements analysis. *Requirements engineering*, 3(1): 48–65, 1998. 5.3.4, 7.2, 7.3

[118] Brad S Trinkle, Robert E Crossler, and Merrill Warkentin. I'm game, are you? reducing real-world security threats by managing employee activity in online social networks. *Journal of Information Systems*, 28(2):307–327, 2014. 5.1.1

[119] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases. *science*, 185(4157):1124–1131, 1974. 6.5

[120] U.S. Bureau of Labor Statistics. Information security analysts: Occupational outlook handbook: U.S. Bureau of Labor Statistics, March 2016 (accessed March 8, 2016). Available at: `http://www.bls.gov/ooh/computer-and-information-technology/information-security-analysts.htm`. 1, 2.3.4

[121] US-CERT. OpenSSL Heartbleed vulnerability (CVE-2014-0160) | US-CERT, April 2014 (accessed March 9, 2015). Available at: `https://www.us-cert.gov/ncas/alerts/TA14-098A`. 5.2.2

[122] Axel Van Lamsweerde. Requirements engineering in the year 00: A research perspective. In *Proceedings of the 22nd international conference on Software engineering (ICSE 2000)*, pages 5–19. ACM, 2000. 5.3.4

[123] Axel van Lamsweerde, Simon Brohez, Renaud De Landtsheer, and David Janssens. From system goals to intruder anti-goals: Attack generation and resolution for security requirements engineering. In *Workshop on Requirements for High Assurance Systems (RHAS' 03)*, pages 49–56, 2003. 3.6.2

[124] C. Wagner, Mathieu Pierfitt, and Josie McCulloch. Juzzy online: An online toolkit for the design, implementation, execution and sharing of type-1 and type-2 fuzzy logic systems. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 2321–2328, July 2014. 6.2.3

[125] Christian Wagner. Juzzy - A Java based toolkit for type-2 fuzzy logic. In *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*, pages 45–52, April 2013. 6.2.3

[126] Lisa Wallander. 25 years of factorial surveys in sociology: A review. *Social Science Research*, 38(3):505–520, September 2009. 5.1, 5.1.1, 5.2.1

[127] Rick Wash and Emilee J Rader. Too much knowledge? Security beliefs and protective behaviors among united states internet users. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 309–325, 2015. 2.3.1

[128] Dongrui Wu. A brief Tutorial on Interval type-2 fuzzy sets and systems. *Fuzzy sets and systems*, 2010. 6.2.3

[129] Dongrui Wu and Jerry M. Mendel. Designing practical interval type-2 fuzzy logic systems made simple. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 800–807, July 2014. 6.2.3, 6.2.3, 6.2.3, 6.2.3

[130] Hui Yang, Anne De Roeck, Vincenzo Gervasi, Alistair Willis, and Bashar Nuseibeh. Speculative requirements: Automatic detection of uncertainty in natural language requirements. In *2012 20th IEEE International Requirements Engineering Conference (RE' 12)*, pages 11–20, Sept 2012. 6.1.1

[131] Robert K Yin. *Case study research: Design and methods*, volume 5. SAGE Publications, 2009. 3.8, 3.8, 3.8, 4.3, 4.3, 4.3

[132] Lotfi A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965. 6.1.2