How do practitioners reason about security requirements? An interview study

Luciana Provenzano
Mälardalen University, Västerås, Sweden
luciana.provenzano@mdu.se

Robbert Jongeling
Mälardalen University, Västerås, Sweden
robbert.jongeling@mdu.se

***Abstract*—In the development of modern software-intensive systems, security aspects are increasingly emphasized, with new laws and regulations putting more demands on manufacturers. Requirements elicitation must therefore carefully consider security aspects. The literature contains various frameworks that have been proposed to aid in the elicitation of these types of requirements. We are interested to understand how, in industrial practice, persons responsible for cybersecurity reason about so-called "security requirements". To find out, we perform eight semi-structured interviews with experts having leading roles in cybersecurity in large companies. We identify the concepts that they leverage when reasoning about security requirements, what other aspects they look at when identifying security requirements, how they differ between security requirements and other requirements, and what their definition of a security requirement is. In this paper, we report on this interview study and our analysis of it. We highlight the commonalities and crucial differences between experts' reasoning, and a surprising spread of conclusions regarding the identification of example requirements as being security requirements or not. Our analysis opens a new perspective on how to deal with security requirements, we hypothesize the benefits of using multiple approaches for elicitation and a single approach for requirements specification.**

***Index Terms*—Security requirements, Cybersecurity, Information security**

I. Introduction

An increased focus on the security of software-intensive systems also incurs an increased focus on so-called security requirements. A huge number of methods of Security Requirements Engineering (SRE) (e.g., [22], [8], [11], [15], [28], [30], [25], [17]) have been proposed in the academic literature to deal with the elicitation and specification of security requirements. This rich literature is accompanied by many surveys (e.g., [13], [16], [29], [7], [1], [18]) that classify the existing SRE methods according to some criteria. The surveys underline how difficult it might be for the users to select the SRE method that best fits their needs. Since security experts are the potential end-users of SRE methods, we wonder about what they need from a SRE method to be used in their daily work.

Therefore, in this study, we want to understand what the perspective of industrial practitioners on security requirements is. What do they consider to be security requirements exactly? What, to them, defines a requirement as being a security requirement or just any other type of requirement? In particular: what do engineers consider to be security requirements and what do they consider to be *not* security requirements?

The distinction between these categories is by no means obvious and depends on many factors. But on what does it depend? Where do practitioners draw the line? Which security concepts are leveraged by them to understand security requirements? We aim to get insight into the reasoning of practitioners to be able to answer these questions.

To discover how practitioners understand security requirements and, hence, to answer the overall question "how do practitioners reason about security requirements?", we consider the following research questions:

- **RQ1**: What are the security concepts that are leveraged by practitioners to identify a security requirement?
- **RQ2**: Is there any additional information practitioners use to identify a security requirement?
- **RQ3**: Is there any difference between security requirements and other types of requirements?

When considering security concepts (in RQ1), we refer to concepts such as threat, asset, vulnerability, countermeasure, attack, attacker, risk, as defined in e.g. [3], [20], [4]. Additional information (in RQ2) refers to "everything else than the requirement itself" that practitioners leverage to classify the requirement. It can be, but is not limited to, architectural design, traceability links, security analysis, or previous knowledge.

To find answers to our questions, we have performed eight semi-structured interviews with cybersecurity experts in leading roles at different companies across various branches. In this paper, we report on the commonalities and differences in their reasoning about security requirements. Our analysis shows limitation in this strict division and opportunities for improved means to deal with security requirements.

The remainder of the paper has a typical organization as follows. Section II places our work in the context of the literature. Section III describes our research methodology. Section IV presents the results of the interviews. Section V presents our analysis of the results. Section VI presents our analysis of and reflection on the results. Section VII concludes.

II. Related work

Several surveys (e.g., [1], [18], [13], [12], [27]) have been conducted to investigate to which extent the rich arsenal of Security Requirements Engineering (SRE) approaches (e.g., [22], [8], [11], [15], [28], [30], [25], [17]) proposed in the literature consider security standards and, hence, support threat modelling and risk analysis as well as include security-related

terminology. The literature also contains different methods that leverages security-related concepts such as the architecture, standards, other perspectives, etc. to create a method combining these perspectives to elicit security requirements [11]. These methods and surveys acknowledge the importance of considering the security terminology and standards when dealing with security as a fundamental part of every SRE method to be useful for practitioners.

Recent works are focusing on practitioners' understanding of security from different perspectives, such as to make cybersecurity decisions [26], the factors that influence a proactive security behaviour of software developers [2], and "how to ensure that work on security requirements is taken seriously" [31]. Studies such as [14] investigate how well the existing SRE methods actually work when used by people different from their designers also acknowledge the importance of the users' understanding for the effectiveness of these methods in practice. In our study, we take a different perspective and investigate how practitioners understand security requirements. By understanding which key elements practitioners leverage, we aim to build upon and potentially extend existing work calling for a unification of existing approaches to ensure more inclusive elicitation and specification [32].

III. Methodology and study design

We performed eight semi-structured interviews with cybersecurity experts working in large companies. The steps of our methodology are illustrated in Figure 1. We elaborate them in the remainder of this section.

A. Preparation of the interviews

1) Alpha test: As a first step in our research, we presented a poster at a workshop with industrial collaborators who were interested in being informed and joining ongoing research on cybersecurity. Our poster aimed both to gauge interest in our work and to collect initial feedback on the form of the research. The poster is included in the supplementary material [21]. During this poster session, we played a mini-game to observe how industry practitioners reason about security requirements. The mini-game consisted of answering the question "Is this a security requirement" for three provided requirements on a continuous spectrum between "no", "maybe" and "yes" a security requirement. During the poster session, we have tested open discussions about the security requirements, we got twelve initial responses. We used these initial results as input for refining our interview guide by updating the requirements, structuring the interview questions, and by establishing a more concrete set of answering options, rather than a continuous spectrum. We have not used the responses of this alpha test in the results presented in this paper.

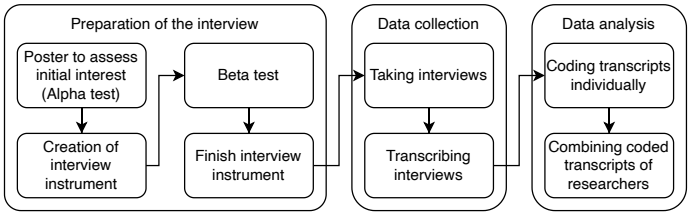

Fig. 1. Followed research process

2) Creation of the interview instrument: After the successful alpha test, in terms of initial results and initial interest from industrial partners, we have developed our initial poster into an interview study. The interview guide is available in the supplementary materials [21]. We performed semi-structured interviews [23], with a mix of closed and open questions. The closed questions are about classifying requirements as security requirements or not, and we follow-up with open questions about the interviewees' rationale for their classifications. By this mix, we aim to obtain answers that show how interviewees both quantitatively and qualitatively experience security requirements. Moreover, by asking the same predefined questions in each interview, we aim to minimize any possible "steering" bias of the interviewer or observer towards the answers of the interviewee.

a) Target population and sampling: The general target audience are cybersecurity experts working for companies developing software-intensive systems. Moreover, we aim for the interviewees to work in some way with requirements. We have contacted people meeting these criteria from our network and invited them to take part in the interview study. During this selection we have aimed for diversity in gender, age, and relevant working experience. We have initially sent out twelve invitations and received eight positive replies, indicating significant industry interest in the topic.

b) Selecting requirements used in the interviews: During the interviews, we ask interviewees to classify ten requirements on a four-point scale between "not at all a security requirement" and "absolutely a security requirement". The requirements intentionally concern both security and non-security requirements to observe which elements are leveraged by the practitioners when understanding the type of requirement. Specifically, including requirements that are not explicitly security-related allows us to identify the concepts or other elements of information, if any, that make the interviewees classify the requirement specifically as a security requirement. An overview of included requirements along with their source is reported in Table I. To arrive at this selection, we defined the following process.

The requirements that the authors intentionally consider as security requirements (R1, R3, R5, R6, R7, R8) are taken from research papers that deal with SRE approaches. The selected papers mostly concern templates-based SRE approaches since these papers provide a specific syntax of the security requirements. This is done to ensure that the security requirements are valid examples (correct). Moreover, using examples of security requirements from research in a study that involves practitioners allows a comparison between research and practice.

The security requirements are formulated to include at least one of the core security concepts according to templates. These

TABLE I
REQUIREMENTS, WITH THEIR SOURCE AND RATIONALE. "AUTHOR" REFERS TO THE ANONYMOUS FIRST AUTHOR OF THE PAPER.

ID	Requirement	Source	Rationale
R1	The system shall deny any user to successfully use DoS (Denial-of-Service) attacks to reduce availability of the system.	[19]	This requirement is a system security requirement, mitigating a threat of attackers using DoS attacks.
R2	The door control system shall ensure that all train passengers' doors are closed and locked when the train is running.	Author	Based on requirements from a real industrial setting. Intentionally not security, but a system functional safety requirement.
R3	The system manager shall be able to define and activate the logging procedure for the critical information system.	[11]	This is a stakeholder security requirement concerning the asset log data that are vulnerable to violation by attackers.
R4	When the train arrives at the station, the train control system shall open all passengers' doors on the train driver's request.	Author	Based on requirements from a real industrial setting. Intentionally not security, but a system functional safety requirement.
R5	The system shall log every time a user checks medications against a list of drugs noted to be ineffective for the patient in the past.	[22]	This is a security requirement based on the template for accountability, which is considered a security objective.
R6	The system shall provide reliable information to the users who have legitimate access to the website.	[15]	This is a system functional security requirement because it has been elicited by applying JAD, which is a structured technique for the elicitation of functional requirements of a system.
R7	The system shall register real persons as customers.	[28]	This is a system functional security requirement. Obtained from reasoning using the "Misuse case" paradigm.
R8	The customer shall register to the system using a unique username and password in order to proceed to book a ticket.	[10]	This security requirement is based on a template in which username and password are considered as security mechanism.
R9	Each of the train passengers' doors shall have a push button installed on the inner side to allow the passengers to open the door.	Author	Based on requirements from a real industrial setting. This is not intended to be a security requirement, but rather a design contraint for the system.
R10	The monitoring system shall refresh the passengers' doors status (open/closed/locked) each second.	Author	Based on requirements from a real industrial setting. This is not intended to be a security requirement, but rather a system functional and non-functional requirement.

templates are known by the authors but not explicitly included for the interviewees. Specifically, the security requirements are chosen by considering that "the specification of a security requirement typically requires to identify problem domain concepts related to (i) the **assets** that it protects, (ii) the **threats** that it is driven by, (iii) the **vulnerability** that it prevents, and (iv) the **countermeasure** that it suggests" [11]. All the proposed requirements are complete in terms of being self-contained, i.e., they can be understood without requiring specific domain knowledge. The requirements that are not security-related are based on real applications (such as the ones for the train doors), utilizing the prior experience of the first author of this paper. Lastly, the number of requirements has been limited to ten to allow the interviewers to "remaining silent and allowing the participants to think aloud" (non-verbal probing [9]), which is fundamental to collect the interviewee's rationale. This also allows to keep the time of the interview within one hour.

c) Interview form: Interview studies have a common threat to their validity due to potential misunderstandings or misinterpretations of the answers of interviewees. A recommendation is to return the transcripts to the interviewees for validation of the raw data [23]. To ensure direct correct recording of the data, we instead created a form that was used by the observer to summarize the answers to each question by the interviewee. During the interviews, this form was made visible to the interviewees by screen sharing and the interviewees were asked to remark any deviations they spotted in the interpretations of their answers. Through these means, we avoid a long feedback cycle where interviewees, much later than their interview, would have to think back on their reasoning and ascertain if our interpretation of their answers is correct. The interview guide is available in the supplementary materials [21].

3) Beta test: Before running the interviews with our interviewees, we have tested our interview guide on a researcher at our department with knowledge of security. Following the beta test, we have finalized the formulation of the requirements to their form as included in Table I.

B. Data collection

After the completion of the interview instrument, we have scheduled individual online meetings with the interviewees. Each interview was scheduled for one hour and consisted of an interviewer (the first author of this paper), observer (the second author of this paper), and a single interviewee. Each interview started with an explanation of the research and obtaining consent for the recording and anonymized processing of the collected information. The remainder of the interview consisted of three parts: (i) gathering background information on the interviewees' working experiences, (ii) classifying the requirements, and (iii) reflection on the ways of working with security requirements and on the interview itself. Specifically, during the part (ii) of the interviews, we ask the interviewees to:

1) for each of the ten provided requirements, classify them on a 4-point scale as "not at all", "mostly not", "mostly", or "absolutely" a security requirement;
2) provide a rationale (in conversation with the interviewer) for their classification;
3) explain what additional information they require in order to be able to classify the requirement, if this was not possible.

We have used Microsoft Teams to record and automatically transcribe the interviews. This has greatly reduced the tran-

scription effort, but has not eliminated it, since the automatic transcripts had to be validated and corrected in some places. Moreover, we have removed filler words from the transcripts and only kept the interviewees' line of reasoning.

C. Data analysis

We examined the collected data through thematic analysis, as suggested in [24]. Two authors coded the transcribed interviews independently, using a mixed form of coding. We started by using in vivo (inductive) coding to catch the interviewees' reasoning on security requirements, which is the main aim of this research, while keeping the interviewees' perspectives in the coding itself [24]. We also used versus and descriptive coding to catch possible duality (for example, function/non-functional, especially useful to answer RQ3) and the main idea brought by the data respectively. In the second round of coding, we applied descriptive coding, for which we used pattern coding to identify emergent categories that were later grouped into categories under the main themes following from the research questions. The initial codes of both authors were merged into the final form after a meeting to discuss them. To agree upon the initial codes, we compared the codes we found when coding independently the rationale of each requirement, and we searched for similarities in the codes by looking at words (such as asset, non-functional), terms (such as system design, asset to be protected), and the overall meaning of the code within its context (the rationale) by having in mind the research questions (the main theme). Author's annotations to the coding were also used to support discussions and decisions.

The agreed codes were then organized into categories, and the categories grouped under three main themes, namely *security concepts*, *additional information*, *differences*, corresponding to the three research questions introduced in Section I.

IV. Results of the Interviews

In this section, we report on the results of the eight interviews, organized by the three parts of the interview: demographics, requirements classification, and reflections. Each interview was scheduled for an hour, but lasted between 32 and 55 minutes, on average 41.5 minutes. Table II shows the date of the interviews and the length of the recordings rounded to the nearest minute.

TABLE II
Interviews

ID	Company	Date	Length
I1	C1	2023-11-22	32 min.
I2	C2	2023-11-28	55 min.
I3	C3	2023-12-05	48 min.
I4	C4	2023-12-07	41 min.
I5	C5	2023-12-07	42 min.
I6	C6	2023-12-15	33 min.
I7	C7	2023-12-15	39 min.
I8	C8	2023-12-15	42 min.

A. Demographics

Table III lists the anonymized demographic information about the interviewees. We have interviewed experts with senior roles in industry, their average (mean) experience in security is 14.25 years. Within this set, we have aimed to recruit diverse interviewees, with respect to their age, gender, security-specific working experience, and working domains. We interviewed two women and six men. The interviewees all have responsibilities in IT/OT cybersecurity, with a majority having roles regarding OT cybersecurity, i.e., security regarding the software-intensive systems developed at their companies.

Each interviewee works at a different company, we provide an overview of the companies in Table IV. We include an order of magnitude of the company size in terms of total number of employees and the size of the team that the interviewee works with directly, to get a better idea of their work context.

B. Requirements classifications

The classifications by the interviewees on the ten requirements listed in Table I are shown in Table V. Out of these ten requirements, nine had a majority agreement towards either "no" or "yes", when we group "not at all" and "mostly not" as "no", and group "mostly" and "absolutely" as "yes". Considering this binary division, we see 1x(8-0), 7x(7-1) and 1x(6-2). R7 is the outlier with very mixed responses and the only requirement for which every possible answer was marked by at least one interviewee.

In the remainder of this section, we show the identified categories and sub-categories as a result of the thematic analysis of the interview transcripts. We distinguish three main themes following the research questions: the common security concepts leveraged, additional information leveraged, and the main differences between security and non-security requirements as thought of by the interviewees. In Section V, we further elaborate on the codes supporting each category and include quotes from the interview to further illustrate the reasoning of interviewees.

1) Security concepts (RQ1): Within the theme of leveraged security concepts to identify requirements as security requirements, we identify the sub-categories in Table VI. To illustrate the similarities of some of the sub-categories, we have grouped them further into categories. We see that well-established security concepts are utilized, such as the CIA triad (Confidentiality, Integrity, Availability) and AAA (Authentication, Authorization, Accounting). Moreover, interviewees have identified threats to assets in the form of attacks from attackers.

2) Additional information (RQ2): Table VII shows the additional information interviewees leveraged to identify the requirements as security requirements. Alternatively, these were types of information the interviewees were looking for when initially not being able to classify the requirements. The most common type of additional information is more context about the requirement. Several further sub-categories are listed, most commonly occurring were context about the system's design/architecture and context about the underlying

TABLE III
INTERVIEWEES' ROLES, RESPONSIBILITIES, AND SECURITY-SPECIFIC EXPERIENCE

ID	Role	Responsibilities	Experience
I1	System architect and lead of development security team	In charge of security of development environment, and closely involved with product security.	30 years
I2	Software solution manager	Leads team addressing software security issues and co-leads security-related certification.	10 years
I3	Security coach	To help development teams at the company create more secure software.	30 years
I4	Senior technical specialist and security architect	Consulting on technical decisions for customers to identify their security requirements and advise on how they can be fulfilled.	25 years
I5	Embedded system architect and technology specialist in cybersecurity architecture	Driving the initiative on meeting cybersecurity regulations by design within the company.	5 years
I6	Business information security officer	Ensuring that processes are in place to understand and fulfill information- and cybersecurity requirements.	7 years
I7	Program cybersecurity manager	Enforcing requirements on products and programs. Ensuring full process from requirements to ensuring that they are fulfilled.	2 years
I8	Cybersecurity architect	Guiding the different teams in how to implement security requirements. And teaching them threat modeling, security analysis.	5 years

TABLE IV
COMPANY INFORMATION

ID	Domain	Company size (#employees)	Team size (#persons)
C1	Industrial electronics	100-200	6
C2	Industrial electronics	100-200	16
C3	Industrial electronics	3000-5000	6
C4	Software	100,000+	11
C5	Automotive	15,000-20,000	15
C6	Telecommunications	15,000-20,000	16
C7	Railway	5000-10,000	5
C8	Industrial electronics	100,000+	80

TABLE V
NUMBER OF ANSWERS PER REQUIREMENT TO THE QUESTION: IS THIS IS A SECURITY REQUIREMENT?

	Not at all	Mostly not	Mostly	Absolutely
R1	0	1	2	5
R2	3	4	0	1
R3	0	1	4	3
R4	5	3	0	0
R5	1	5	2	0
R6	0	1	2	5
R7	2	1	3	2
R8	0	1	0	7
R9	7	0	0	1
R10	4	3	0	1

TABLE VI
CODING WITHIN THE THEME: SECURITY CONCEPTS LEVERAGED

Category	Sub-category
Properties to keep	Confidentiality, Integrity, and Availability (CIA)
	Privacy
	No information leaks
Functionality to ensure	Auditability
	Authentication, Authorization, and Accounting (AAA)
	Monitoring/Access control
What to protect (against)	(Security function of) Assets
	Attacker
	Attacks/Threats
Reason for requirement	Countermeasures
	Obligations from the standard

reason for the requirement. Interviewees also commonly speculated about the design of the system when thinking about possible scenarios in which a non-security requirement could become relevant for security anyway. When searching for the underlying reason for the presence of a particular requirement, interviewees, for example, remarked that the underlying reason for requiring logging (and the type of information that is logged) influences if that requirement is related to security or not.

TABLE VII
CODING WITHIN THE THEME: ADDITIONAL INFORMATION LEVERAGED

Category	Sub-category
Leveraged context areas	Usage of the system
	Implementation process
	System design and implementation
	Underlying reason for the requirement
	General
Needed information from	Security function/goal
	Security/risk analysis
Requirement shortcomings	Requirement is too vague
	Requirement should be split up
	Requirement includes unnecessary detail

3) Differences (RQ3): Table VIII shows three categories of concerns raised by the interviewees that we can fit under the theme "differences between security requirements and any other type of requirements." The interviewees commonly identified other types of requirements, such as safety requirements, business requirements, or system requirements. Moreover, interviewees commonly distinguish between functional and non-functional requirements. Functional requirements were typically seen as not security requirements, but they are not necessarily so.

C. Definition of security requirements

Table IX lists the answers given by the participants to the question "what is a security requirement to you?". Commonly, interviewees mention protection of assets against threats or prevention of attacks. This is expected and aligns well with the well established security concepts from e.g. Kim and Lee [11].

TABLE VIII
CODING WITHIN THE THEME: DIFFERENCES BETWEEN SECURITY REQUIREMENTS AND ANY OTHER TYPE OF REQUIREMENTS

Category
Other types of requirements identified
Considerations about functional versus non-functional requirements
Multiple possible "labels" for a requirement

V. ANALYSIS OF THE RESULTS

A. Pattern of reasoning (RQ1)

We observed that each interviewee has their own *pattern of reasoning* that they apply systematically to categorize the example requirements as security requirements or not. This pattern is the same whether the requirement is judged security or not, and it is consistent with the interviewee's definition of security requirement, as reported in Table IX. The only observed exception is I1, who reasons in terms of protecting assets, but defines security requirements in terms of achieving the CIA triad. We also observed that the pattern of reasoning of the interviewees is mostly based on the internal process and security standards they use in their daily work, as well as, the knowledge acquired from their experience within security. Two of the interviewees do not know or use any of the security requirements methods proposed in the literature. Others use a mix of processes defined internally at their companies, and other knowledge. Specifically, two interviewees mention threat modelling, three mention standards including explicitly ISO/IEC 62443 and ISO/IEC 27001, and two interviewees mention STRIDE [25]. We have identified 5 distinct patterns of reasoning, summarized in Table XI. In the remainder of this subsection, we discuss each of these 5 patterns and how they emerge from the interviewees' answers.

1) Protecting assets: Interviewee I1 mostly reasons in terms of `assets` and `protection of the security function`. When classifying R1, I1 says "*to be security related it should either compromise the security functionality of the product [..] which this requirement does not [..] the integrity of the data we protect is still in place. [..] The data is the valuable asset.*" and to classify R2 "*I do not see the asset (to be protected)*". I1 thinks that R1 is mostly not a security requirement and R2 is not at all a security requirement. The interviewee I1 applies the same reasoning to, for example, decide that R3 is absolutely a security requirement, namely "*critical information sounds as an important asset* ".

Interviewee I8 thinks that security requirements are about "*protecting the system or protecting the data or protecting the users*" (refer to Table IX). Indeed, I8 leverages in the first place `assets` and `countermeasures` in terms of `access control` "*which is clearly a cybersecurity requirement*" (R1) to categorize the example requirements. For example, to justify that R9 is not at all a security requirement, I8 thinks about something to be protected, when they say: "*it's about what we are protecting. If there is something to protect, what is the asset that we're handling in this requirement*". While when scoring R3 as mostly a security requirement, I8 says: "*logging mechanisms are quite often also part of the security audit logging. It is really one of those security mechanisms that are important*". Thinking in terms of `countermeasures` is also supported by the rationale of R1 (scored as absolutely a security requirement), when I8 tries to figure out what could be put in place to manage/protect the access control, that is "*hardening task [..] is also definitely something related to cybersecurity*".

Interviewee I7 follows a clear sequence of steps when dealing with security requirements, starting by looking for `attacks/threats` that should be protected against, e.g., "*cybersecurity content with respect to the denial of service, with respect to attack, and user access control also*" (R1). In the next step, I7 considers the why and how of the requirement, which correspond to their definition of security requirement in IX, as well as the recommendations from the security standards that apply to their domain, e.g. "*according to the standard, you need to be able to log the events that you identify as cybersecurity events. So there is a long list of events that we are logging and it doesn't have to do anything with any person*" (R3). So, when it comes to understand how a goal, e.g. "*I could see it as a goal*" (R1), can be implemented, interviewee I7 would need the system architecture and the context, e.g. "*if you would put in the context, the communication and protection of the signals sent to the doors, then we could discuss cybersecurity*" (R2), and "*This is the system-specific. [..] If you think about the web system [..]*" (R7).

2) Identifying attacks: Interviewee I3 leverages the security concepts `attacks/threats` and `attacker`. Specifically, interviewee I3 justifies the usage of the security concepts attackers and threats by explaining their process to work with security requirements: "*we always start with the threat first. The threat is the why, the threat describes what the attacker does, and the security requirement is the countermeasure to protect the threat from happening*". Indeed, interviewee I3 applies this process consistently to classify all the example requirements, as suggested by these excerpts taken from their rationales: "*This text gives me context that there is some sort of attacker and they do bad to the system that reduces availability*" (R1) and "*I cannot see the attacker*" (R2).

3) Achieving the CIA triad: Interviewees I2 and I4 search for the `CIA` triad in first place. Interviewee I4 states that they have "*in mind always [..] non-functional and function and the CIA triad* " as also supported, e.g., by these statements: "*monitoring is not part of the CIA triad*" (R3), "*It doesn't even connect to the CIA triad*" (R5), "*Reliable information would point to the integrity part of the CIA triad*" (R6). Interviewee I2 also thinks that in their application domain "*availability is the top concern if you look at security* " and uses `availability` to decide, for example, that R1 and R5 are absolutely a security requirement and mostly not a security requirement respectively. This is supported by "*I'd*

TABLE IX
DEFINITION OF A SECURITY REQUIREMENT.

ID	Definition
I1	Security requirements are usually non-functional requirements that aim to increase some or more parts of the CIA triad (Confidentiality, Integrity, Availability.)
I2	To counter threats and not specifically related to the functions of the system
I3	A security requirement is a thing to prevent threats
I4	A security requirement is most often a non-functional requirement that I could connect to the CIA triad
I5	Requirements that help to achieve a security goal would be deemed as security requirements
I6	The security requirement is basically a statement expressing what the organization needs to do to protect its employees, assets, strategies, and so forth towards malicious actions
I7	The statement that defines first of all what am I protecting? what is the goal? and then which means am I using?
I8	The requirement which is there for the sake of protecting the system or protecting the data or protecting users. You should talk about the confidentiality, integrity, availability. Then, of course, there is also auditability and things like that

say it is most towards robustness and availability of the product" (R1) and "*the log [..] is part of this 3A (`AAA` in Table VI) where the user has the privilege to access the system, the right user can access the system*" (R5).

4) Implementing security goals: Interestingly, interviewee I5 often expresses the need of the system architecture and context to understand the relationships that exist with the requirement in order to put the requirement in the "*right place*". In addition to the security concepts, interviewee I5 would need the system architecture to figure out how to "*break it (security goal) down to something more concrete*" (R1). Correspondingly, this interviewee thinks that some example requirements are high-level (system) requirements or security goals "*formulated, of course, as the requirement*" and therefore "*you would decompose it into technical requirements how to do that*".

The `system design and implementation` as well as the `context about the usage of the system` are therefore the additional information about the possible technical solution used to classify the requirement. In addition, the interviewee needs information from the `security goals`, for example in R7 "*but it really depends on how they do it, what would be the protocol for deciding if it's a real person or who it is*", and R9 "*I don't see any cybersecurity goal [..]. You are both looking again to the way it can be used, so which is kind of this scenario, and also the asset itself. So again, the architecture of the system, the physical part of the system.*".

5) Searching for malicious intent: The interviewee I6 has a broad perspective on security. They think about security at the organizational level, as stated in their definition of a security requirement: "*[..] what the organization needs to do to protect its employees, assets [..]*" and in terms of responsibilities within the organization, as supported by "*it's definitely the security department that will work with setting up the requirements*" (R1) or "*it's within the security responsibility to maintain a correct view of the status* " (R10). So, they also include personal security (R2, R9) in their judgment. That is most likely the reason why almost all the example requirements are scored as security related requirements. Another interesting aspect is that interviewee I6 reasons in terms of "*malicious*" to identify the security requirements. In the rationale of R1, for example, they state that "*denial of service attacks is, as we see it, malicious*" while for R4 they say "*I can't see the malicious intent that I'm trying to mitigate here*", and for R8 "*it's really easy for a malicious actor to say that he or she is somebody else*". As a result, I6 always leverages security concepts in first place, such as `attacks/threats` or `CIA`, to classify the example requirements.

As an example of the different patterns that interviewees apply to categorize the example requirements, let us consider requirements R1. Five interviewees score R1 as absolutely a security requirement, as shown in Table V. The interviewees, namely I2, I3, I4, I6 and I8, use different concepts (X), that correspond to their own "pattern of reasoning", to reach the same conclusion on the same requirement.

TABLE X
SAME SCORE, DIFFERENT REASONING (R1)

ID	Pattern of reasoning	Sub-category
I3	Identifying attacks	Attacks/Threats, Attacker
I2, I4	Achieving CIA triad	Availability, non-functional
I8	Protecting assets	Access control
I6	Searching for malicious intent	Attacks/Threats, Attacker

It is worth noting that the interviewees also search for additional information to identify the security requirements, such as the `system design and implementation` in Table VII and `other types of requirements` in Table VIII. This happens especially when they cannot leverage any of the security concepts they are used to. Also, some interviewees, such as I1, I2, I4, I5 and I8, reason in terms of functional/non-functional requirements or security/other requirements (such as business, environment, and safety) as a means to identify a security requirement (refer to `considerations about functional versus non-functional requirements` in Table VIII). I4

clearly states that they have "*in mind always [..] non-functional and function and the CIA triad*" when dealing with security requirements, I2 thinks that "*it is not a functional requirement*" to support their judgment about R1 (absolutely a security requirement), and I1 says "*I would not call this requirement a security requirement at all. It is a safety requirement*" (R2).

B. Statement versus context (RQ2)

In addition to the identified patterns of reasoning, we observed that the interviewees search for the context of the requirements to supplement their assessment done based on the requirement's statement, as supported by the arguments provided in the following sub-sections. We refer to the statement of a requirement as the content of the requirement along with its syntax and keywords that are employed to express the requirement [5]. On the other hand, the context of the requirement refers to aspects external to the statement of the requirement, such as the system's architecture/design, the reason for the requirement and the usage of the system, as summarized under `leveraged context areas` in VII.

1) Statement of the requirement: The statement of the requirement plays a central role in the classification of the example requirements. For example, I1 leverages at first the content of R7 to decide that it is mostly a security requirement, as supported by: "*the key is "real". [..] I would write identified individual so to know who is the customer in order to protect the system then. Unidentified people should not gain access to the system*". This example also shows that besides the security concepts from the content that are leveraged to identify the requirements, there are other relevant aspects, such as specific nouns or verbs, that are leveraged too. Moreover, some interviewees, such as I7 and I1, express the need of a specific content to state that the requirement is security related. When justifying that R7 is not at all a security requirement, I7 says: "*the security perspective would be how you store the information, if anyone can log to the system. And then what that user can do. But really checking if it's the real user or not. It doesn't really ring the bell from the security point of view*". I1 says about R2 (classified as "not at all a security requirement") "*the access to the door control system would be the security requirement but the function of the door control system is not a security requirement*".

2) Context of the requirement: Requirement R7 is an example requirement for which three interviewees (I1, I3, I5) mention the context as something needed to support their judgment. Specifically, I1, I3 and I5 think that R7 is a security requirement by analysing its content through the security concepts `threats`, `attacks/threats` and `goal` that form their pattern of reasoning respectively. However, they think that the context of the requirement is missing. I1 says "*it is possibly again a matter of context*", I3 says "*I can say that it is mostly a security requirement because of business logic security reasons, such as fraud, type of attacks on the system, this would be a very obvious countermeasure in security requirements. But it needs context*", and I5 says "*it really depends on the how they do it [..] depending on the context*". Hence, they score R7 as mostly a security requirement.

I4 is not pointing explicitly to the need for more context when they say "*there would be another requirement stating that the system shall verify that the customers' information put in are correct, and you've verified that it is some possibly governmental database or something. But that would be a follow up requirement. [..] That would be a security requirement and if the customer didn't write it, it will be a requirement that we would derive due to this*". However, they keep reasoning about a possible architecture to derive the missing security requirement "*it would be part of the architectural process to actually identify all those functions*" and "*they will start deriving the architecture*", to say that R7 is not at all a security requirement.

Interviewee I7 is also trying to imagine a general context around R7 to justify that R7 is absolutely not a security requirement, when, in their rationale, they think to "*it could be part of the intrusion detection system*" or "*if you think about the web system.. You could be afraid that many of these could affect your, let's say, availability or your response time, but..*" (`general` in Table VII).

Similarly, the interviewee I2 thinks that more context is needed about the reason for the logging (`underlying reasons for the requirement` in Table VII) to assess if R3 is a security requirement. They state: "*it (the classification) depends on the use case. If we talk like more debugging and stuff like that is something but if we talk about audit logging if something happens,.. it is related to security incidents*". They score the requirement R3 as mostly a security requirement.

The context is definitely needed by many interviewees to figure out if a requirement that at first glance is understood as non-security by its content, could be a security requirement when looking at it more carefully. For example, the interviewees I1 and I2 think that R10 "*is more related to the safety function*" (I2). However, they try to come up with a context in which R10 might be a security requirement. I1 "*can think of situation of this requirement could be a security requirement*", that is "*the actual communication between the sensors and the monitoring station could be a security requirement*" (`system design and implementation` in VII). I2 suggests to "*break it down into more requirements under it then you will probably come into issues such as you may need login of the status, you want to have maybe encryption so you do not inject traffic for the status*" (`implementation process` in Table VII). I3 "*would not say that (R10) is a security requirement because I can imagine the attacker, the business logic, people want to sneak onto the train kind scenario that would trigger this, but there is a lot a weak logic steps from this steps to that*". But, they say that "*this requirement has a little bit thorn of a use case, this is why I start thinking*" (`usage of the system` in Table VII).

It is worth noting that some interviewees do not always

search for the requirement's context even if the requirement's content is "*unclear*". For example, I6 suggests that R7 "*must be written much better. But it's definitely a security requirement, that every user should be uniquely identified*", and score R7 as absolutely a security requirement because "*as customers*" (the unclear part) is interpreted as users.

Some interviewees, on the other hand, search for the context even if the requirement's statement includes words in the security domain, such as `Denial of Service attack` in requirement R1, that should make clear that the requirement is security-related (R1 is indeed intended by the authors as a security requirement, as stated in Table I). For example, the interviewee I8 thinks about a specific task to address the Denial of Service attack, "*I think that this will end up being a hardening task*" (`implementation process` in Table VII), in the rationale to justify that R1 is absolutely a security requirement.

C. Multiple labels (RQ3)

A third observation from our analysis is that many interviewees agree on the challenge of putting a unique label on the requirements. As an example, I2 thinks that requirements R1 and R2 can be stamped with multiple labels. Specifically, to the interviewer's question "*the fact that this requirement (R1) can be recognizable as non-functional requirement makes you think that it is or could be a security requirement?*" I2 answers: "*you have functional, safety, security.. you probably have others as well. But they are not super clear, 100%, it is this one or that one. It can be a mixed. Because if you talk about robustness, typically this leads towards safety and maybe security in the product*". And, about requirement R2, I2 says "*Just to say that safety is something, functional is something, security, environmental,... you can label them (the requirements) as much as you want. It is quite difficult to slap a label on*". Similarly, I5 thinks that R2 "*connects to several domains and then once we want to have this requirement realized and we maybe decompose it further into some technical solutions around that how to actually achieve it. Then you can also get safety relevant requirements and cybersecurity relevant requirements in terms of some mechanism in place for that*".

Specifically, the interviewees set the label that mostly emphasizes the aspect they think be more dominant or relevant in the requirement. This is clear, for example, in the rationale given by I8 about R2 being mostly not a security requirement: "*it sounds very much like a safety requirement. [..] Because clearly if this is not the case, you can cause harm to people by having the doors not closed, for example. So, however, safety and security is interconnected as we all know, and when handling this requirement you would probably end up in looking at possible threats that could make this requirement not being fulfilled, so to speak*".

In general, interviewees are looking for an emphasis or keyword in the requirements to understand which of multiple possible labels are the most appropriate. For example, I1 states about R8: "*The important part is not unique username but the fact that you can identify the user through the registration process.*". About the same requirement, I8 reasons: "*But that (handling passwords) is not the emphasis of this requirement in my perspective (...) Register, that is the thing that triggers that this is a security requirement"*.

The fact that the interviewees try to relate the example requirements to security may depend on the specific task they are requested to perform, namely answer the question "is this a security requirement?". However, it is interesting to see that many interviewees acknowledge the fact that the requirements, especially the ones understood at a first glance as non-security related, can be thought from different perspectives, hence it is difficult to label them in a unique way. And, to this aim, one must take a precise perspective, e.g., security, safety, robustness, and write the requirement's statement that is appropriate for that perspective.

VI. Discussion

A. Insights from the interviewers

From our analysis of the security concepts leveraged to recognize a security requirements, we see that the interviewees start by leveraging the security concepts (or terms/verbs that remind the security concepts) they are familiar with based on their knowledge and experience, to analyse the requirement's statement. They search for additional information, where the context and the type of the requirements are the most used (as in Section IV-B), to eventually support their assessments.

From our analysis of the statement versus the context, and of the definitions of security requirements as created by the interviewees, we see that the security concepts leveraged to classify the security requirements are also used for the definition. This internal consistency across interviewees is of course good, but the differences of reasoning and of the provided definitions by engineers raises an important question on how we can ensure that the required information is in the requirement to facilitate correct classification.

Several interviewees remarked about many requirements that "security" was not the only applicable label. Indeed, we have heard different labels as well, such as safety, system, environment, and business requirements. Such labels can be useful to group requirements concerning the same type, but if the labelling is then too subjective (interviewees also remarked that a subjective assessment of the emphasis of a requirement influenced their classifications) the grouping may lead to miscommunication between stakeholders relying on different reasoning. Moreover, managing multiple labels on a single requirement also means that this requirement must be examined by engineers from different areas such as safety and security. Indeed, labelling a requirement as e.g., safety, rather than security, is necessary for the product assessment. For each quality (safety, security, etc.) different standards dictate the process to be followed to achieve the required level of the quality (safety, security, etc.).

The nature of this study was to look at one specific type of requirements: security requirements. But our analysis about multiple possible labels, and the reasoning of interviewees about them, could be applicable to other types of requirements,

TABLE XI
SUMMARY OF PATTERNS OF REASONING

ID	Security Concepts (RQ1)	Additional Info (RQ2)	Differences (RQ3)	Pattern of reasoning
I1	Asset/Countermeasure	System design	Functional vs. non-functional	Protecting assets
I8	Asset/Countermeasure	Implementation process System Design	Functional vs. non-functional Other types Multiple labels	
I7	Attacks/Threats Standards	System design Security goals General	Other types Multiple labels	
I3	Attacks/Threats Attacker	Reason for requirement Usage of the system	Other types	Identifying attacks
I2	AAA	Reason for requirement Implementation process	Functional vs. non-functional Other types Multiple labels	Achieving CIA triad
I4	CIA Triad	System Design	Functional vs. non-functional Other types	
I5	All concepts leveraged	System Design Usage of the system Security goals	Other types Multiple labels	Implementing security goals
I6	Attacks/Threats CIA triad	Implementation process Security analysis	Other types	Searching for malicious intent

too. Given the observation that requirements may need to be labeled with multiple types, we would expect similar observations in a replications of this study, but rather focused on, e.g., safety requirements. This part of our results may therefore also be applicable beyond the scope of just security requirements and open a perspective on dealing with many types of requirements in the collaborative development of software-intensive systems.

B. On the results

All security concepts leveraged by the interviewees are well-known in the literature. This is a comforting result, indeed it would be worrying if the results were very surprising in this regard. However, we did find interesting differences in the reasoning applied by the various practitioners. Our study provides us insights into the elements that practitioners use to work with security requirements, and therefore the elements that they would need in order to make the correct classifications. Surprisingly, these elements go beyond the security-related concepts like threats, vulnerabilities, risk, assets, up to and including reasoning on other types of requirements.

When reflecting on the interviews, some of the interviewees remarked that going through the classifications was helpful for them to realise their patterns of reasoning about security requirements. I5: "*this was a good exercise*", I2: "*This requirement was difficult. I like this discussion a lot and I'm interested in the results.*" We appreciate these comments and they may imply that increased training about these rationales could benefit practitioners.

Lastly, we reflect on the question from the title: how do practitioners reason about security requirements? We have heard different perspectives in the interviews, but when we merge them and the displayed rationales from the interviewees, we can derive a statement according to the union of all perspectives. According to the interviewees, a security requirement is: A statement that, by means of a *countermeasure*, *counters/prevents/protects* the *system/data/users/assets/etc.* against *threats/attacks/malicious actions* by *attackers* to ensure *confidentiality/integrity/availability/auditability/privacy/etc.*, motivated by *standards/security goals/objectives*.

C. Future directions in research and practice

Our results could be used by researchers to further investigate more comprehensive methods of requirements elicitation and specification that are more suitable to manage the requirements of modern complex systems, especially with security and safety aspects, while assisting practitioners in setting-up the requirements activities that best fit their internal processes and standards. Such an approach can further support bridging the gap from industrial practice to the existing requirements elicitation frameworks from the academic literature.

1) Multi-perspective security requirements elicitation: How to accommodate the existing SRE methods to meet the different ways of working of practitioners? Indeed, based on the observed different patterns of reasoning about security requirements (as described in Section V-A and summarized in Table XI), we hypothesize that there is a need for a more comprehensive way to elicit security requirements that benefits from the different perspectives brought by different people to support structured and creative reasoning about security requirements. Such a method would be especially valuable to improve the completeness of elicited requirements by employees new to security and/or requirements, as we have observed that many of the interviewees strongly rely on their experience when classifying the requirements.

2) Collaboration on requirements: How to make experts from different domains collaborate to create requirements of complex systems? In practice, as we observed in Section V-C, most requirements will not be exclusively labelled as security, safety, or any other label, but rather be identified as a combination of several of those types. This suggests the need to

harmonize the processes and support collaborations between engineers from these different domains in their joint work on the system's requirements to achieve and assess the different system qualities. This is a real need as well as a difficulty that companies face when developing the modern complex systems. There is a danger in considering the differently labeled requirements in isolated silos, instead, a common view of the requirements, potentially labelled with different relevant aspects of the system can enhance collaboration across areas.

3) A unique requirements specification: How to provide a commonly understood format of the different types of requirements? As we observed, multiple different patterns of reasoning can lead to different labeling of the requirements. The additional information leveraged (see Section V-B) as well as the definition of security requirements provided by the interviewees, give indication of the different elements that specifications of security requirements shall contain to ensure their unambiguous and complete specification, even in a setting involving engineers with diverse backgrounds. Such additional information also includes links to the context of the system, in term of e.g., traceability links to security analysis, or to aspects of the system's architecture.

In short, we derive from the interviews a need for combining multiple ways to elicit security requirements, but a single way of specifying requirements.

D. Threats to validity

This paper presents empirical software engineering research. In particular, for the semi-structured interviews we limit the threats to validity by following guidelines for empirical software engineering research, particularly those on qualitative surveys [6].

We have designed our study in multiple phases, starting with the alpha test in the form of a poster session and later performing a trial interview in the beta test. From each phase, we have learned and made improvements to the study design. Other measures were taken too, to limit threats to the validity during the design of this study. For example, we have created a form to record the interviewees' rationale during the interview so that the interviewees could already during the interview confirm or correct our interpretations. Since our study needs expertise in security and/or requirements to answer our questions, we could not conduct random sampling of the interviewees. Instead, we gathered a sample of cybersecurity engineers with expert knowledge and relevant expertise in cybersecurity and requirements from companies in various domains. We believe that we have interviewed a relevant sample of experienced engineers, as detailed in Table III. We have stopped interviewing after the initial eight interviews, due to having reached saturation, i.e., we had largely converging classifications of all the requirements, and seen a pattern in the rationales of interviewees when classifying them. Since each interviewee repeatedly applied the same pattern of reasoning to classify the requirements and the pattern can be recognized after the classification of few requirements independently whether they are judged security or not (refer to V-A), we think that ten requirements suffice. We also observed that a larger number of requirements would have requested an additional cognitive effort from the interviewees that could have compromised the quality of their rationales.

There is a remaining threat related to the bias of the researchers. We aimed to limit this threat by having two authors code the transcripts independently and later merging the transcripts. The coding from both authors were very similar (they contained explicit terms, such as system design, access control, asset to be protected, that unambiguously clarified the concept in the coded rationale), giving us confidence in the limitations of our personal bias. Nevertheless, it is difficult to eliminate this bias entirely, since it is brought by the knowledge and experience of the first author that brings the value to the analysis of the data as well.

VII. Conclusion

In this paper, we have investigated how practitioners understand security requirements. We have performed semi-structured interviews with eight experts in cybersecurity with leading roles in industry. Our analysis of the transcripts has shown that (i) the interviewed experts exhibit five different patterns of reasoning about security requirements but mostly agree on their classification, (ii) experts utilize contextual information about the use of the system and its architecture to classify the requirements, (iii) the difference between security requirements and other types of requirements is vague and in some cases not worth making, since there are many labels possible for single requirements. In this last regard, we note that requirements for complex software-intensive systems will rarely involve just a single perspective and thus shall be labeled and managed in a way that supports collaboration of all relevant aspects (such as both safety and security). We have also observed that practitioners have a well-defined way to work with security requirements which is mostly based on the security standards that apply to the specific application domain than to security methods proposed in the literature.

The logical next step that we are already working on is to provide practitioners with means on how to improve the completeness of their requirements elicitation in practice. To this end, we propose to combine multiple requirements elicitation techniques, thereby allowing engineers to discover requirements that were previously not met. We hypothesize that a single means of requirements specification in combination with this diverse approach to elicitation will provide the best approach to allow engineers with very different rationales to successfully collaborate on the development of complex systems that require the inclusion of multiple perspectives such as safety and security.

Acknowledgment

This work is partly supported by the SERENDIPITY project funded by the Swedish Foundation for Strategic Research (SSF), and Software Center https://www.software-center.se.

References

[1] Javed Ahmad, Chaudhary Wali Mohammad, and Mohd. Sadiq. On software security requirements elicitation and analysis methods. IT in Industry, 9(1):24–36, 2021.

[2] R. Arizon-Peretz, I. Hadar, and G. Luria. The importance of security is in the eye of the beholder: Cultural, organizational, and personal factors affecting the implementation of security by design. IEEE Transactions on Software Engineering, 48(11):4433–4446, 2022.

[3] CENELEC. Information Technology - Security techniques - Information security management systems - Overview and vocabulary. CEN/CENELEC, Brussels, 2020. Ref. No. EN ISO/IEC 27000:2020 E.

[4] Common Criteria. Common criteria for information technology security evaluation - part1 - introduction and general model. Retrieved from https://www.commoncriteriaportal.org/cc/, accessed: 09.03.2023, 2022.

[5] Jeremy Dick, Elizabeth Hull, Ken Jackson, Jeremy Dick, Elizabeth Hull, and Ken Jackson. Requirements engineering in the problem domain. Requirements Engineering, pages 113–134, 2017.

[6] Ralph et al. ACM SIGSOFT empirical standards. CoRR, abs/2010.03525, 2020.

[7] Benjamin Fabian, Seda Gürses, Maritta Heisel, Thomas Santen, and Holger Schmidt. A comparison of security requirements engineering methods. Requirements Engineering, 15(4):7–40, 2009.

[8] S Gürses and T. Santen. Contextualizing security goals: a method for multilateral security requirements elicitation. Sicherheit, 2006.

[9] Hanna Kallio, Anna-Maija Pietilä, Martin Johnson, and Mari Kangasniemi. Systematic methodological review: developing a framework for a qaulitative semi-structured interview guide. Journal of Advanced Nursing, 72(12):2954–2965, 2016.

[10] Massila Kamalrudin, Nuridawati Mustafa, and Safia Sidek. A template for writing security requirements. Communications in Computer and Information Science, 809:73–86, 2018.

[11] Bong-Jae Kim and Seok-Won Lee. Understanding and recommending security requirements from problem domain ontology: A cognitive three-layered approach. Journal of Systems and Software, 169:110695, 2020.

[12] Siwar Kriaa, Ludovic Pietre-Cambacedes, Marc Bouissou, and Yoran Halgand. A survey of approaches combining safety and security for industrial control systems. Reliable Engineering & System Safety, 139:156–178, 2015.

[13] Nadeem Malik, Mohammad Anwar, Mohammed Nazir, and Khurram Mustafa. A systematic review and analytical evaluation of security requirements engineering approaches. Arabian Journal for Science and Engineering, 2018.

[14] Fabio Massacci and Federica Paci. How to select a security requirements method? a comparative study with students and practitioners. In Nordic Conference on Security IT Systems (NordSec), pages 89–104, 2012.

[15] Nancy R. Mead, Eric D. Hough, and Theodore R. Stehney. Security quality requirements engineering (square) methodology. CMU/SEI-2005-TR-009 ESC-TR-2005-009, 2005.

[16] Daniel Mellado, Carlos Blanco, Luis E Sánchez, and Eduardo Fernández-Medina. A systematic review of security requirements engineering. Computer Standards & Interfaces, 32(4):153–165, 2010.

[17] H. Mouratidis and P. Giorgini. Secure tropos: a security-oriented extension of the tropos methodology. International Journal of Software Engineering and Knowledge Engineering, 17(2):285–309, 2007.

[18] Denisse Muñante, Vanea Chiprianov, Laurent Gallon, and Philippe Aniorté. A review of security requirements engineering methods with respect to risk analysis and model-driven engineering. CD-ARES 2014 Lecture Notes in Computer Science, 8708:79–93, 2014.

[19] Suvda Myagmar, Adam J Lee, and William Yurcik. Threat modeling as a basis for security requirements. In Symposium on Requirements Engineering for Information Security (SREIS), 2005.

[20] Committee on National Security Systems. National information assurance (ia) glossary. CNSS Instruction No. 4009, 2003.

[21] Luciana Provenzano and Robbert Jongeling. Supplementary material for the paper "how do practitioners reason about security requirements? an interview study". https://doi.org/10.5281/zenodo.10942725. https://zenodo.org/records/10942725.

[22] Maria Riaz, Jason King, John Slankas, and Laurie Williams. Hidden in plain sight: Automatically identifying security requirements from natural language artifacts. In 2014 IEEE 22nd International Requirements Engineering Conference (RE), Karlskrona, Sweden, pages 183–192. IEEE, 2014.

[23] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering, 14:131–164, 2009.

[24] Johnny Saldana. The coding manual for qualitative researchers. SAGE, 2021.

[25] Adam Shostack. Threat modeling: Designing for security. John Wiley & Sons, 2014.

[26] Benjamin Shreeve, Catarina Gralha, Awais Rashid, João Araújo, and Miguel Goulão. Making sense of the unknown: How managers make cyber security decisions. ACM Transactions on Software Engineering and Methodology, 32(4), 2023.

[27] Paulina Silva, René Noël, Santiago Matalonga, Hérnan Astudillo, Diego Gatica, and Gastón Marquez. Software development initiatives to identify and mitigate security threats - two systematic mapping studies. CLEI Electronic Journal, 19(3):5, 2016.

[28] Guttorn Sindre and Andreas L. Opdahl. Eliciting security requirements with misuse cases. Requirements Engineering, 10:34–44, 2004.

[29] Amina Souag, Camille Salinesi, and Isabelle Comyn-Wattiau. Ontologies for security requirements: A literature survey and classification. In Advanced Information Systems Engineering Workshops: CAiSE 2012 International Workshops, Gdańsk, Poland, June 25-26, 2012. Proceedings 24, pages 61–69. Springer, 2012.

[30] Amina Souag, Camille Salinesi, Raúl Mazo, and Comyn-Wattiaw Isabelle. A security ontology for security requirements elicitation. In International Symposium on Engineering Secure Software and Systems (ESSoS 2015), pages 157–177, 2015.

[31] Evenynke Terpstra, Maya Daneva, and Chong Wang. Agile practitioners' understanding of security requirements: insights from a grounded theory analysis. In 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW), pages 439–442. IEEE, 2017.

[32] Sven Türpe. The trouble with security requirements. In Proceedings of the 25th international Requirements Engineering conference, pages 122–133, 2017.