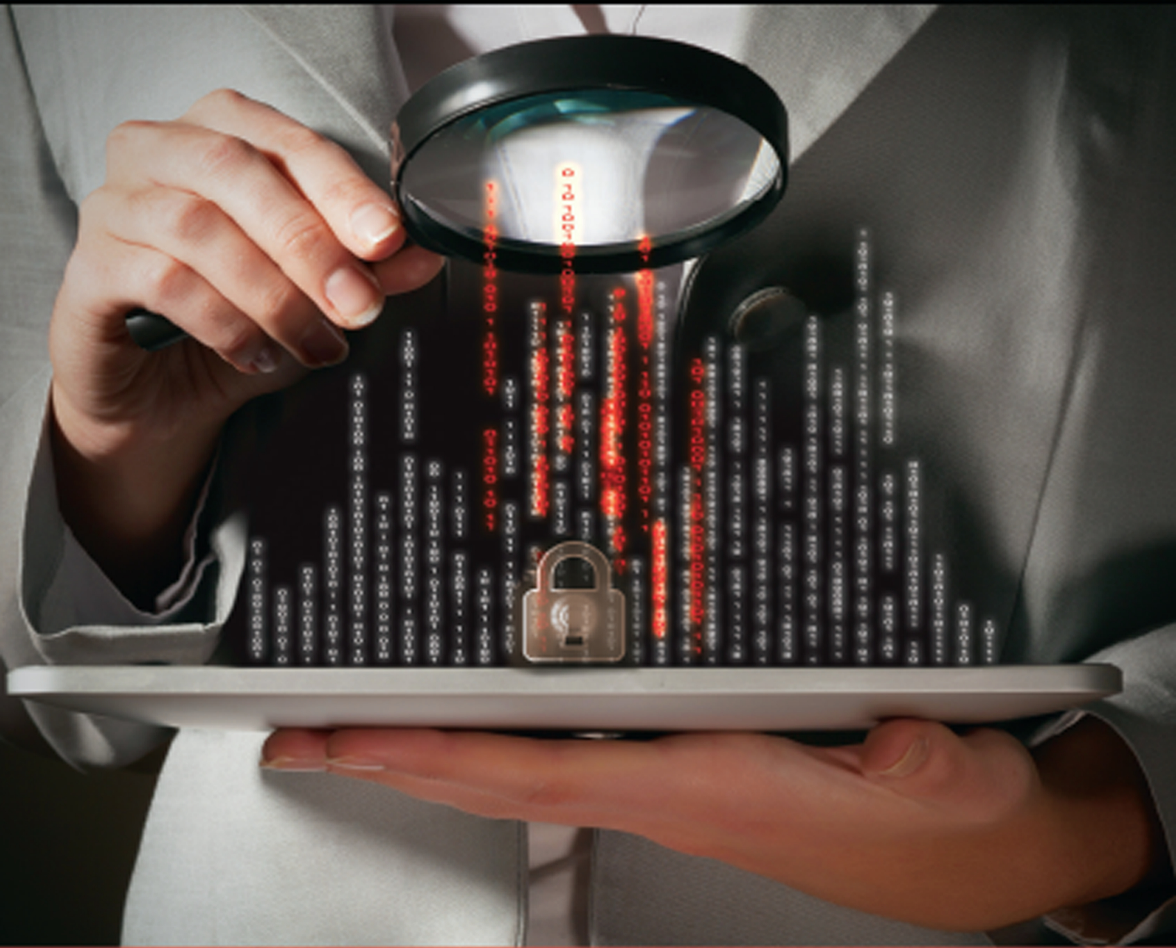
Empirical Research for Software Security

Foundations and Experience



Edited by Lotfi ben Othmane
Martin Gilje Jaatun • Edgar Weippl

CRC Press
Taylor & Francis Group

CRC Series in Security, Privacy and Trust

Empirical Research for Software Security

Foundations and Experience

CRC Series in Security, Privacy and Trust

AIMS AND SCOPE

This book series presents the advancements in research and technology development in the area of security, privacy, and trust in a systematic and comprehensive manner. The series will provide a reference for defining, reasoning and addressing the security and privacy risks and vulnerabilities in all the IT systems and applications, it will mainly include (but not limited to) aspects below:

- Applied Cryptography, Key Management/Recovery, Data and Application Security and Privacy;
- Biometrics, Authentication, Authorization, and Identity Management;
- Cloud Security, Distributed Systems Security, Smart Grid Security, CPS and IoT Security;
- Data Security, Web Security, Network Security, Mobile and Wireless Security;
- Privacy Enhancing Technology, Privacy and Anonymity, Trusted and Trustworthy Computing;
- Risk Evaluation and Security Certification, Critical Infrastructure Protection;
- Security Protocols and Intelligence, Intrusion Detection and Prevention;
- Multimedia Security, Software Security, System Security, Trust Model and Management;
- Security, Privacy, and Trust in Cloud Environments, Mobile Systems, Social Networks, Peer-to-Peer Systems, Pervasive/Ubiquitous Computing, Data Outsourcing, and Crowdsourcing, etc.

PUBLISHED TITLES

Empirical Research for Software Security: Foundations and Experience
Lotfi ben Othmane, Martin Gilje Jaatun, Edgar Weippl

Intrusion Detection and Prevention for Mobile Ecosystems
Georgios Kambourakis, Asaf Shabtai, Constantinos Kolias, and Dimitrios Damopoulos

Touchless Fingerprint Biometrics
Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti

Empirical Research for Software Security: Foundations and Experience
Lotfi ben Othmane, Martin Gilje Jaatun, and Edgar Weippl

Real-World Electronic Voting: Design, Analysis and Deployment
Feng Hao and Peter Y. A. Ryan

Protecting Mobile Networks and Devices: Challenges and Solutions
Weizhi Meng, Xiapu Luo, Steven Furnell, and Jianying Zhou

Location Privacy in Wireless Sensor Networks
Ruben Rios, Javier Lopez, and Jorge Cuellar

Empirical Research for Software Security

Foundations and Experience

Edited by Lotfi ben Othmane
Martin Gilje Jaatun • Edgar Weippl

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20171024

International Standard Book Number-13: 978-1-4987-7641-7 (Hardback)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Preface

The software security field has been plagued by “accepted truths” or “self-evident statements” that at their core are based on nothing more than that some “guru” at one point thought it sounded like a good idea. Consequently, these “accepted truths” have often proved to be of varying longevity, as fashion changes and new fads emerge. Empirical research allows to test theories in the real world, and to explore relationships, prove theoretical concepts, evaluate models, assess tools and techniques, and establish quality benchmarks across organizations. The methods for doing such research have been used in several areas, such as social sciences, education, and software engineering. These methods are currently being used to investigate software security challenges and mature the subject.

The purpose of this book is to introduce students, practitioners and researchers to the use of empirical methods in software security research. It explains different methods of both primary and secondary empirical research, ranging from surveys and experiments to systematic literature mapping, and provides practical examples.

Rather than a complete textbook on empirical research, this book is intended as a reference work that both explains research methods and shows how software security researchers use empirical methods in their work. With some chapters structured as step-by-step instructions for empirical research and others presenting results of said research, we hope this book will be interesting to a wide range of readers.

In the first chapter, Koen Yskout et al. offer a primer on empirical research in the area of security and privacy by design, explaining what to expect and what not to expect as researcher or reviewer. They address the frequent lack of empirical research on new methods or techniques in the early stages of security and privacy design. Their experience-led chapter discusses how to design and perform controlled experiments and descriptive studies in this domain. It contrasts the methods typically applied by beginning and more experienced researchers with those frequently expected by reviewers, and strikes a balance between scientific rigor and pragmatism dictated by the realities of research.

The structured approach guides the reader through the phases of study design, from research questions and study design to execution to data analysis and dissemination. In many cases, recommendations for additional reading are provided for the reader seeking to explore a given area more in depth. It not only provides practical advice for researchers on issues such as using students as test subjects but also includes helpful tips for reviewers, explaining what to look for in an empirical study and which allowances to make when reviewing small-scale studies. With this two-fold approach, it will certainly prove helpful both for empirical researchers who are just starting out and for reviewers of empirical studies who may not have performed such studies themselves.

Moving from primary to secondary studies, "Guidelines for systematic mapping studies in security engineering" by Michael Felderer and Jeffrey C. Carver explains how to use the systematic mapping method to provide an overview of a research domain and to determine which topics have already been extensively covered and which are in need of additional research. The authors of this chapter illustrate the usefulness of systematic mapping studies and provide an overview of systematic mapping studies previously published in the security engineering field. They compare different guidelines for such studies in software engineering and then adapt them to the security engineering field. Illustrated by examples from actual studies, they provide extensive methodological support for researchers wishing to conduct such a study, explaining how to search for and select which studies to include, how to assess their quality and how to extract and classify the data.

In the following chapter "An Introduction to Data Analytics for Software Security," ben Othmane et al. share their experience on using data analytics techniques to derive models related to software security at SAP SE, the largest European software vendor. They use data analytics to study raw data with the purpose of drawing conclusion using machine learning methods or statistical learning methods. They describe in the chapter the data analytics process that the authors practiced with and give an overview of a set of machine learning algorithms commonly used in the domain. They also describe how to measure the performance of these algorithms.

"Generating software security knowledge through empirical methods" by René Nóel et al. combines both primary and secondary research. The authors explain how to use experimental methods to generate and validate knowledge about software security. In addition to a general discussion of validity in research and the use of empirical methods, they guide the reader step by step through an experimental study, explaining why the various methods are chosen and what knowledge can be gained from them. In each section, the theory or method is supplemented with the actual data from the study. Budding empirical researchers will surely find the explanations of how to formulate and test a research hypothesis useful. Following the description of the randomized experiment, the authors explain how they supplemented it with a systematic literature mapping study and a case study, again detailing the reasons for and outcomes of each method applied. Another emphasis of this chapter is on the importance of experimental replication, explaining not just why and how replications should be conducted but also detailing different types of replications.

The chapter "Visual Analytics: Foundations and Experiences in Malware Analysis by Markus Wagner et al. shows how visual analytics, which combines automated with human analysis by providing powerful visual interfaces for analysts to examine, can be used to analyze the enormous data loads of malware analysis.

It explains the basics of visual analytics (data processing, models, different visualization techniques and human interaction with the visualized data, knowledge generation, and how to design and evaluate visual analytics systems) and how its methods can be applied to behavior-based malware analysis. This is illustrated with three projects that used visual analytics for malware analysis. The methods employed in these projects are compared and used as a basis for recommendations for future research.

In "Evaluating Classification Accuracy in Intrusion Detection," Natalia Stakhanova and Alvaro A. Cárdenas offer an excellent example of how a systematic literature review can be used to analyze methods employed by the research community and detect previously unknown ontological issues. They analyze the use of different evaluation methods for intrusion detection systems (IDSs) and investigate which factors contribute to or hamper the adoption of new IDS evaluation methods. They found that the vast majority of researchers use traditional metrics, including methods that have been criticized as insufficient, and are reticent toward adopting new ones. In their analysis, they also found a wide variety of different names for the same metrics, prompting the call for a unified terminology. They also propose guidelines for researchers introducing new evaluation metrics, suggesting that new metrics introduced be explained clearly so that they might be adopted by other researchers as well. In addition to the literature analysis, this paper discusses the benefits and challenges of all metrics, and compares IDSs by classification accuracy, and proposes a framework for the validation of metrics.

Martin Gilje Jaatun explains how the Building Security in Maturity Model (BSIMM) might be used as an academic research tool. Initially developed by software security company Cigital to assess the security maturity level of their clients by quantifying their software security activities, the BSIMM survey in its original form was administered in person by representatives of Cigital. It measures twelve practices in the domains of governance, intelligence, SSDL touchpoints, and deployment. Jaatun describes how it was converted into a questionnaire with a follow-up interview. While this method does not provide a BSIMM score in the traditional sense, the low-threshold approach can yield interesting data for researchers in the security domain.

In "Agile test automation for web applications," Sandra Ringmann and Hanno Langweg address the topic of test automation for security testing. They advocate the integration of (automated) security testing into the other testing processes of the software development life cycle. In this very practice-oriented paper, the authors discuss the main requirements for tools used in agile testing, where testing is performed by all members of the agile development team, many of whom are not security experts: they must be user-friendly and human-readable. In addition to a discussion of different risk rating methodologies and threat models, they provide a thorough and well-structured overview of different testing methods and tools and explain how to choose the right one for the job.

The paper presents a number of vulnerability scanners some of which are partially or completely open source and compares their scan results. It also provides an overview of freely available dynamic analysis tools and presents their use in BDD (behavior-driven development) frameworks, which allow everyone in the development team to participate in or follow the testing process.

In "Benchmark for Empirical Evaluation of Web Application Anomaly Detectors," Robert Bronte et al. argue the need for a common benchmark for detecting application-layer attacks. Their chapter provides an overview of benchmarks in the field previously suggested by other researchers and compares their advantages and disadvantages as well as the attributes they focused on before setting out to define the characteristics a unifying benchmark for application-layer attack detection would have to have. They pay careful attention to the environment required to generate benchmark data and demonstrate how such data could be used to evaluate an intrusion detection system.

Validity is the extent to which the design and conduct of empirical studies are likely to prevent systematic errors or bias. Empirical research studies are associated always with validity threats that limit the use of the results. Cruzes and ben Othmane provide in the chapter "Threats to Validity in Software Security Empirical Research" a taxonomy of validity threats that apply to secure software engineering qualitative and quantitative studies. In addition, they give examples on how these threats have been addressed or discussed in the literature. Rigorous threats to validity helps to advance the common knowledge on secure software engineering.

The back cover picture is provided by Srdjan Pavelic.

We hope that this book provides an interesting introduction into the use of empirical research methods and helps researchers and practitioners alike select the appropriate evaluation method for their project.

List of Figures

List of Tables

Contributors

Hernán Astudillo
Universidad Técnica Federico Santa María
San Joaquín, Santiago, Chile

Robert Bronte
Kennesaw State University
Kennesaw, Georgia, USA

Achim D. Brucker
The University of Sheffield
Sheffield, UK

Alvaro A. Cardenas
University of Texas at Dallas
Richardson, Texas, USA

Jeffrey C. Carver
University of Alabama
Tuscaloosa, Alabama, USA

Daniela S. Cruzes
SINTEF
Trondheim, Norway

Stanislav Dashevskyi
University of Trento
Trento, Italy

Michael Felderer
University of Innsbruck
Innsbruck, Austria

Eduardo B. Fernandez
Florida Atlantic University
Boca Raton, Florida, USA

Hisham Haddad
Kennesaw State University
Kennesaw, Georgia, USA

Martin Gilje Jaatun
SINTEF
Trondheim, Norway

Wouter Joosen
imec-DistriNet, KU Leuven
Leuven, Belgium

Dimitri Van Landuyt
imec-DistriNet, KU Leuven
Leuven, Belgium

Hanno Langweg
HTWG Konstanz University of Applied Sciences
Konstanz, Germany

Santiago Matalonga
Universidad ORT Uruguay
Montevideo, Uruguay

René Noël
Universidad de Valparaíso
Valparaíso, Chile

Lotfi ben Othmane
Fraunhofer Institute for Secure Information Technology
Darmstadt, Germany

Gilberto Pedraza
Universidad de Los Andes and Universidad Piloto de Colombia
Bogotá, Colombia

Sandra Domenique Ringmann
HTWG Konstanz University of Applied Sciences
Konstanz, Germany

Riccardo Scandariato
Chalmers & University of Gothenburg
Gothenburg, Sweden

Natalia Stakhanova
University of New Brunswick
Fredericton, Canada

Hossain Shahriar
Kennesaw State University
Kennesaw, Georgia, USA

Peter Tsalovski
SAP SE
Walldorf, Germany

Edgar Weipp
SBA Research
Wien, Austria

Kim Wuyts
imec-DistriNet, KU Leuven
Leuven, Belgium

Koen Yskout
imec-DistriNet, KU Leuven
Leuven, Belgium

Chapter 1

Empirical Research on Security and Privacy by Design

What (not) to expect as a researcher or a reviewer

Koen Yskout, Kim Wuyts, Dimitri Van Landuyt, Riccardo Scandariato, and Wouter Joosen

CONTENTS

1.1 Introduction

Research on software security and privacy is very active, and new techniques and methods are proposed frequently. In practice, however, adoption is relatively slow, especially for techniques and methods in the early software engineering phases (requirements elicitation, architecture and design). Yet it is precisely in these early design phases that the security-by-design (and privacy-by-design) principles are expected to yield substantial returns on investment: a little extra early development effort may avoid lots of late re-engineering efforts.

Although these arguments are intuitively convincing, it is our belief that a lack of empirical evidence to support claims about the benefits of early security design is one of the main impediments to adoption. Empirical research is an essential technique to study whether a new method or technique has the promised effects, but also to validate whether it is feasible in practice. Despite their importance, such studies are not performed as often as they should be — there are many hurdles and roadblocks, especially for privacy and security engineering! Quantifying the level of security or privacy of a design is far from trivial. In addition, an attacker can use several approaches to breach a system. Determining whether a security objective was met, is thus challenging. Also, given its sensitive nature, obtaining the security documentation of an industrial software architecture is hard and therefore performing realistic studies can be difficult.

In this chapter, we share our experiences from the past five years with performing empirical studies specifically related to early security and privacy design activities (e.g., [27, 25, 35, 38, 37, 3, 24, 4]). Our empirical research experience mainly consists of controlled experiments and descriptive studies. We present approaches to perform such studies that have worked for us, discuss challenges that we have encountered along the way, and present remedies that we have effectuated to address these challenges. We provide **experience-driven recommendations** both for (beginning) empirical researchers, as well as for reviewers of empirical studies.

To sketch the context, table 1.1 (in the first and second column) illustrates (in a caricatural manner) some of the typical discrepancies that exist between how beginning researchers — or, for that matter, researchers performing pilot validations of their approach — undertake such studies, in comparison to more experienced empirical researchers. The second versus the third column also exemplifies the mismatch in expectations about empirical studies from the point of view of researchers on the one hand, and external stakeholders such as scientific reviewers on the other. The latter discrepancy is often, at least in part, caused by different expectations with respect to the internal and external validity of the study, and has already been described well by others (e.g., Siegmund et al. [28]).

In this chapter, we share our practical experiences with performing such studies. We provide concrete attention points for various aspects of designing, organizing, executing, processing and publishing about empirical studies. This chapter is therefore particularly useful **for security researchers** interested in conducting empirical research, as well as **for scientific reviewers**, as it gives some concrete pointers for assessing such studies. The chapter is in part anecdotal, by referring to concrete incidents and by citing reviewer comments that we have received. This necessarily takes these situations and quotes out of their context, at the risk of losing some of the nuances that were originally present.

The common theme throughout the chapter is what makes these experiments highly challenging: empirical security researchers are continually forced to make difficult *trade-offs* between, on the one hand, the *scientific rigor* essential in scientific and empirical experimentation, and on the other hand, the required level of *pragmatism* to make such studies happen, especially when they involve human participants.

The chapter is structured according to the main phases of the process for conducting an empirical study, where we adopt the terminology used by Wohlin et al. [33]: **scoping** (Section 1.3), where the overall goals and objectives of the study are defined; **planning** (Section 1.4), which involves the careful design of the study; **operation** (Section 1.5), focusing on preparation of subjects, and the actual execution to collect data; **analysis and interpretation** (Section 1.6), i.e., exploring and sanitizing the data, and making scientifically sound conclusions; and **presentation and packaging** (Section 1.7), where the conclusions about the data and research materials

Table 1.1: An illustration of typical discrepancies between beginning researchers, experienced researchers, and reviewers.

To assess the value of a security- or privacy-by-design approach, ...

a beginning researcher performs	***an experienced researcher* performs**	*a reviewer* expects
▷ a small *validation exercise*,	**▷ a *controlled experiment*,**	▷ several *replicated controlled experiments*,
▷ involving a *handful of peers*,	**▷ involving *N representative participants*,**	▷ involving at least 100 ∗ *N experienced industrial developers, experts and practitioners*,
▷ who solve a *small example problem*,	**▷ who solve a *well-scoped design exercise*,**	▷ who perform a *large-scale industrial development project*,
▷ after receiving a short, *ad-hoc introduction* to the approach,	**▷ after receiving a *structured tutorial* about the approach,**	▷ after having *several months of practical experience* in applying the approach,
▷ where the approach introduces a specific *security or privacy design activity* technique,	**▷ where the approach supports *a specific security or privacy design activity*,**	▷ where the approach supports *the entire development life-cycle*,
▷ resulting in measures that are determined only *after the data was collected and processed.*	**▷ resulting in the quantitative evaluation of a *specific, well-defined hypothesis* concerning the approach.**	▷ resulting in the quantification of the influence of the approach on *security, privacy, productivity, compatibility with existing industrial practices, ...*

are wrapped up for publication. We do not intend to explain or even touch upon every activity in this process; elaborate descriptions can be found elsewhere (e.g., [33, Chapters 6–11]). But first, the following section sketches the context of our studies that are used as an example throughout the chapter.

1.2 Empirical Research on Security and Privacy by Design

There is an increasing awareness both in industry and in academic research that complex non-functional cross-cutting concerns such as security and privacy inherently require up-front attention, much in line with the principles of *(software) quality by design*. One example is the recent update of EU regulations which stipulate that software-intensive systems and services involving the processing of user data should be designed according to privacy-by-design principles [11]. In turn, many security vulnerabilities, bugs, and leaks find their roots at the level of the software architecture, because software is built with specific assumptions in mind, which — when invalidated by attackers — cause breakage of such systems (see [1] and [6], for example).

As part of our empirical research, we have studied a number of security and privacy by design methods, notations, and techniques that focus on the early stages of the software development and that aim at bringing the security and privacy by design principles into practice. From the many available techniques (see the surveys in [3, 7, 20, 30], for example), our efforts have focused primarily on STRIDE [15] for security threat elicitation and mitigation, LINDDUN [9, 34] as its counterpart for privacy, and architectural security patterns [36, 26], and security modeling notations [3].

A key question is whether this early development effort really pays off. Are these currently-existing security and privacy design techniques capable of identifying potential issues before they turn into actual problems, and do they effectively lead to software designs that are inherently less prone to security and privacy defects? These are by no means trivial questions to answer, and combined with the observation that empirical studies about these questions are performed far too infrequently, we conclude that *empirical evidence is lacking* to support such claims.

This book chapter crystallizes our main lessons learned, do's and don'ts, tips and tricks, and shares some of our experiences and war stories from over five years of empirical research on security and privacy in the early stages of the software development life cycle (requirements elicitation and analysis, and architectural design). Tables 1.2 and 1.3 below summarize our track record in conducting empirical studies on respectively security by design and privacy by design. We draw examples from these studies throughout the chapter, using the acronyms given in the top row of the tables when referring to them.

In the remainder of this section, we briefly sketch the purpose of each of the studies, which may help to better understand the examples in this chapter. Note that the rest of this chapter intentionally does not discuss the topic or even the findings of our studies, but focuses exclusively on the aspects related to their planning and execution. We gladly refer the interested reader to the corresponding research publications for more information about the results.

In the **Security threat modeling (STM) [27]** study, we have investigated the cost and effectiveness of Microsoft's STRIDE [15] by assessing its correctness, completeness, and productivity. The study concluded that STRIDE is relatively time-consuming to execute, but fairly easy to learn and execute. Nevertheless, we have observed that many threats remained undetected during the analysis.

In the **Security threat modeling (STM2) [unpublished]** study, we are investigating how the correctness and completeness of a security threat analysis using STRIDE is affected by the level of detail in which the data flows of the system are described. The threats elicited by the participants are compared with a baseline that is independently defined by experts.

In the **Secure architectural design with patterns (SPAT1) [38]** study, we have investigated whether providing a fine-grained, systematic structure on top of a catalog of security patterns (as suggested by multiple researchers in the field) improves the performance of the software designer in terms of overall time spent, and the efficiency of finding and selecting a pattern. We concluded that adding more structure can be beneficial, but that this is not self-evident.

In the **Secure architectural design with patterns (SPAT2) [37]** study, a follow-up of the SPAT1 study, we have investigated whether the availability of security patterns increases the security of the resulting design and/or the performance of the designer. The study has lead to the observation that security patterns, in their current form, do not yet achieve their full potential, and that there exists a need for improving them.

In the **Privacy threat analysis at requirements level (PAR) [35]** study, we have investigated the cost and effectiveness of the LINDDUN privacy threat modeling framework [34] during the early stages of software development. Give the limited amount of participants, this study only had an exploratory nature and mainly focused on retrieving feedback from the participants regarding ease of use.

In the **Privacy threat analysis at architectural level (PAA) [35]** study, we have investigated the cost and effectiveness of the LINDDUN privacy threat modeling framework during the architectural design phase. We observed similar results compared to our STRIDE study STM. Although the completeness rate of LINDDUN turned out to be even beter than STRIDE's, we have found that the productivity was only half.

In the **Privacy methodology comparison with privacy experts (PCE) [35]** study, we have investigated the reliability of the LINDDUN privacy threat modeling framework, by comparing the analysis results of privacy experts with those of LINDDUN. We observed that LINDDUN was missing coverage in the areas of data minimization and data inference, which in turn allowed us to improve the LINDDUN methodology. However, LINDDUN did cover a number of threats that were overlooked by the privacy experts.

In the **Privacy threat modeling (PTM) [unpublished]** study, the privacy equivalent of the STM2 study mentioned above, we are investigating how the correctness and completeness of a privacy threat analysis using LINDDUN is affected by the level of detail in which the data flows of the system are described. Again, the threats elicited by the participants are compared with a baseline that is independently defined by experts.

Table 1.2: Empirical research for security by design: running examples

STM	STM2	SPAT1	SPAT2
Type of activity			
Security **T**hreat **M**odeling [27]	**S**ecurity **T**hreat **M**odeling [unpublished]	**S**ecure architectural design with **Pat**terns [38]	**S**ecure architectural design with **Pat**terns [37]
Study goals			
Productivity, correctness, completeness	Correctness, completeness, productivity	Performance and efficiency, impact of pattern catalog structure	Security (soundness and completeness of solutions), performance
Number of participants			
10 teams (41 master students)	93 participants (master students)	45 teams (90 master students)	32 teams (64 master students)
Quantitative vs. qualitative			
Quantitative	Quantitative, with exploration of ease of use	Quantitative (and some qualitative)	Quantitative (and some qualitative)
Output			
Templated threat report	Templated threat report, questionnaires	Report, tool measurements (pattern catalog browsing history, UML models, time, questionnaires)	Report, tool measurements (secured design models, time, questionnaires)
Environment			
Open (10 days offline + 1 lab session)	Restricted (2.5h lab session, no communication, all on paper)	Mixed (2 supervised lab sessions of 2.5h + work at home, at all times restricted to using provided tool)	Mixed (3 supervised lab sessions of 2.5h + work at home, at all times restricted to using provided tool)

Table 1.3: Empirical research for privacy by design: running examples

PAR	PAA	PCE	PTM
	Type of activity		
Privacy threat **A**nalysis at **R**equirements level [35]	**P**rivacy threat **A**nalysis at **A**rchitectural level [35]	**P**rivacy methodology **C**omparison with privacy **E**xperts [35]	**P**rivacy **T**hreat **M**odeling [unpublished]
	Study goals		
Correctness, completeness, productivity, ease of use	Correctness, completeness, productivity, ease of use	Reliability	Correctness, impact of domain knowledge
	Number of participants		
3 teams (8 professionals)	27 teams (54 master students)	5 participants (2 methodology experts + 3 privacy experts)	122 participants (master students)
	Quantitative vs. qualitative		
Qualitative, with exploration of quantitative goals	Quantitative, with exploration of ease of use	Quantitative	Quantitative, with exploration of ease of use
	Data		
Templated threat report, questionnaire, post-it discussion session	Templated threat report, questionnaires, self-reported time tracking, access logs of catalog	Templated threat report	Templated threat report, questionnaires
	Environment		
Mixed (13 hours of lab sessions divided over 3 days. No limitation of technology or communication.)	Open (2 weeks offline time + 2 lab sessions. Self-reported time tracking.)	Open (Offline, no time limitation)	Restricted (2.5h lab session, no communication, all on paper)

1.3 Scoping

> "The natural question for me is [something you didn't investigate]. Why not try to answer this question instead of a question about [what you actually did investigate]?"
>
> —Anonymous reviewer

Security and privacy by design are very broad topics. Therefore, the first decision that has to be made when setting up an empirical study about them is to determine the goals and attention points of the study. These need to be clearly specified before the details of the study can be worked out.

1.3.1 Setting Verifiable Goals

There is a wide range of goals that an empirical study about secure design can try to tackle. Usually, at the highest level, the goal of a study is to demonstrate that a (new) design approach is "good enough" in practice, or "better" than some other approach[1]. While such a goal is intuitively appealing, it needs to be made more concrete in order to evaluate whether it has been reached or not. Words like "better" thus need to be refined into more descriptive qualities, such as more secure, more performant, more efficient, or easier to use. Even then, directly evaluating such goals remains difficult. This is especially true in the context of secure design, where no consensus exists about how to measure the security of a software design [19]. Therefore, the definition of the goal will need to undergo several iterations, going back and forth between what is truly desired and what is verifiable.

Consider our SPAT2 study, for example. In this study, it was clear from the start that we wanted to answer the question whether using security patterns results in a more secure system design or not. Nevertheless, a significant amount of time was spent on devising a manner to translate this goal into a measurable quantity, taking into account that measuring security is far from trivial, especially on the design level, partly because the security of the system depends on (implicit) assumptions made by the designer [19, 13].

▷ **As a researcher:**

- Pay enough attention to the overall goal of your experiment.
- Don't start until you have refined this into a set of verifiable goals, and you are confident these sub-goals together sufficiently address the overall goal.

[1] It's also possible to investigate other questions, for example whether two approaches are "equally good" in practice, rather than one being "better" than another [10].

▷ **As a reviewer:**

- It's okay to assess the relevance and validity of the study's goals in a standalone manner, but keep in mind that the expectations or interests of the researchers may be different from yours.
- Verify whether the sub-goals and specific research questions addressed in the paper together sufficiently allow formulating an answer to the overall goal of the study.

1.3.2 Process- or Result-Oriented

As part of the definition of the goal, the researcher must determine whether the study will focus more on the *process* of the studied approach (i.e., the security design process followed by the participants) or on the *end result* obtained upon completion of the activities (i.e., the security of the resulting design). Determining this focus upfront is essential as it will greatly impact the setup of the study. Most of our studies are result-oriented, with the primary interest being the output produced by the participants (either the secured design or the identified threats). An exception is the SPAT1 study, which focused primarily on the performance and efficiency of the designer, and was therefore process-driven.

Process-driven studies are mainly useful when the researcher wants to analyze detailed aspects of a security methodology, such as ease of use, execution time, action flow, etc. In this case, the activities of the participants will need to be registered more precisely during the execution of the study. This can happen in multiple ways, impacting the accuracy and the required resources, as described later in Section 1.4.4.3.

For result-oriented studies, the process followed by the participant is less important. However, in this case, the results delivered by the participants will be studied in depth. Therefore, it's important to clearly delineate the outcome that the participant is expected to produce. This is described in more detail in Section 1.4.4.2.

To get a more complete picture, combining both types into one study is of course also possible, but a conscious trade-off should be made between the added value of the other type of data with respect to the study's goals and the effort required to collect and process that data. For example, in our result-oriented studies, we have also collected some process measures (such as time) to investigate secondary hypotheses or simply gain more insight.

▷ **As a researcher:**

- Determine upfront what the main focus of the study will be: the process followed by the participants, or the results they produce.

- If your study is process-oriented, think early about what data you need to collect, and how you can do that.
- If your study is result-oriented, clearly define your expectations regarding the output that the participants have to produce. When possible, provide user-friendly templates to make it as easy as possible for them to comply with the expected format.

▷ **As a reviewer:**

- Check that the study setup and type of collected data match the goals of the study.

1.3.3 Quality or Quantity

In addition to the above, it is important for the researcher to determine in advance what kind of findings the study aims for: “hard numbers” (such as precision and recall, efficiency, or execution time) to statistically prove the security method’s soundness or superiority, or “softer” output related to the user experience of the method.

Process-oriented studies may involve several hard to quantify variables (i.e., ease of use, flow of actions) that probably require a more qualitative approach. Of course, certain aspects of them (such as time or efficiency) can still be measured quantitatively. Result-oriented studies are typically more suitable for the quantitative approach (i.e., correctness, completeness).

Quantitative studies can provide “hard” numbers that can help to improve the core of a methodology or technique. Once follow-up studies show an acceptable quantitative result, and a stable state has been reached, methodology designers can evaluate and optimize their methodologies by focusing on the more qualitative aspects of their approach.

Note that quantitative studies do not automatically imply objectivity. Security and privacy are not easy to quantify, hence calculating a quantity like number of true positives (e.g., the number of threats elicited by the participants that are considered correct according to a predefined baseline) is not a simple counting exercise. First, a proper baseline needs to be created, for example by reaching a consensus between experts (see Section 1.5.1). Second, the participants’ results need to be categorized according to this baseline (see Section 1.6.1). Both activities often require a certain amount of interpretation. In every (quantitative) study we have performed so far, we have observed that this turned out to be less straightforward (and more time-consuming) than expected.

▷ **As a researcher:**

- Determine whether your defined goals require a quantitative or rather qualitative approach.
- For quantitative studies, maximize the objectivity of the data collection process, but be aware that some subjectivity may still remain (e.g., expert opinions).

▷ **As a reviewer:**

- Check whether the choice between qualitative and quantitative is appropriate with respect to the stated study goals.
- Be aware that quantitative studies are not necessarily completely objective, and check whether the researchers have sufficiently explained the process by which their numbers have been obtained.

1.4 Planning

A common mistake when executing empirical studies is to underestimate the importance of a thorough design of the study. The planning phase is probably the most important phase of a study. This section zooms into some of the different steps one has to perform, and provides insights into common pitfalls.

1.4.1 Defining Research Questions

> "[This study] answers clear research questions using clearly defined metrics."
>
> —Anonymous reviewer

The iterative definition and refinement of the study goals will eventually lead to the research questions and precise hypotheses of the study, in accordance with an approach such as Goal-Question-Metric (GQM [2]). It should be obvious that these are crucial to the success of the study, but coming up with a clear and solid formulation, and a sound refinement into concrete hypotheses, is not straightforward, especially in a security or privacy context. Furthermore, the researcher should anticipate that the data that will be collected may not be as clear-cut as hoped for, and prepare some contingency plans in order to prepare for potential surprises.