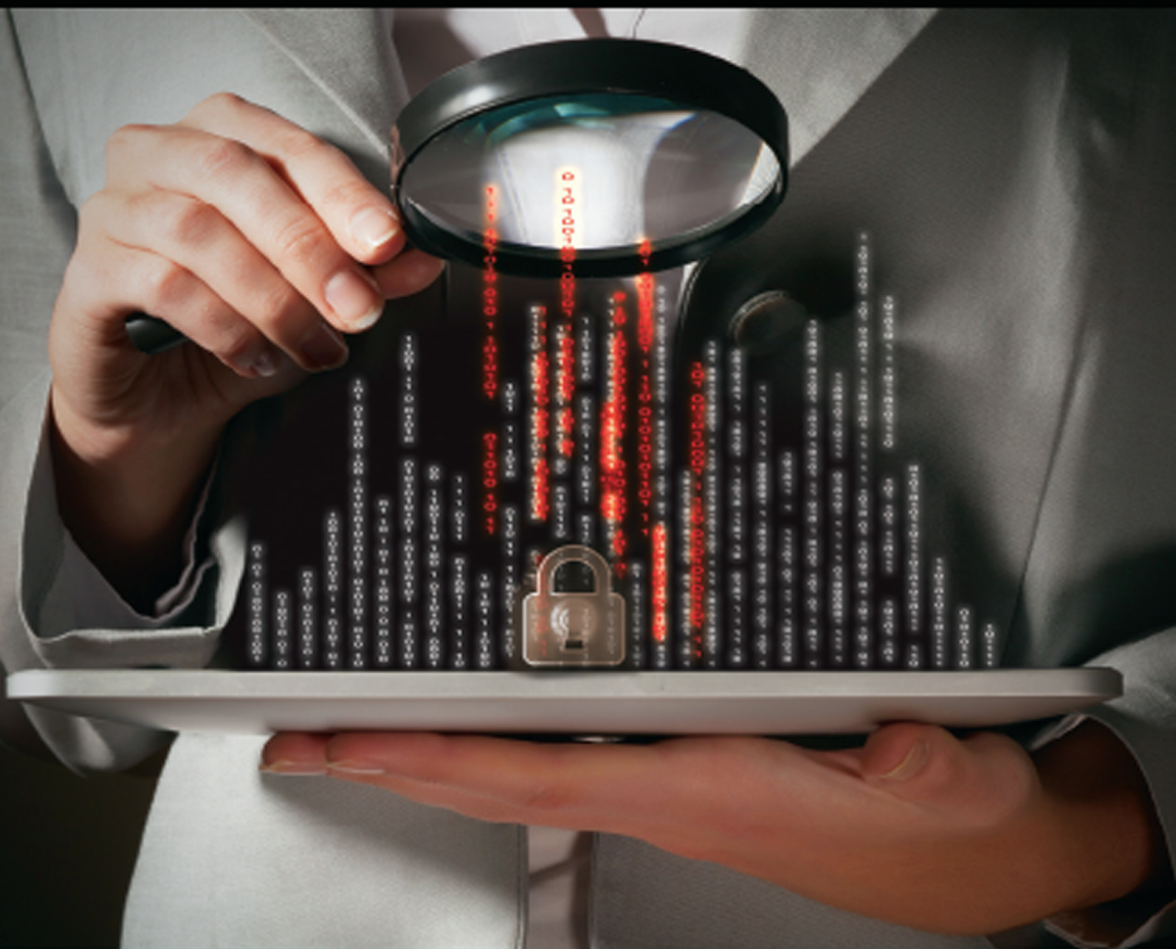


# Empirical Research for Software Security

## 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

## SERIES EDITORS

Jianying Zhou

Institute for Infocomm Research, Singapore  
jyzhou@i2r.a-star.edu.sg

Pierangela Samarati

Università degli Studi di Milano, Italy  
pierangela.samarati@unimi.it

## 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

© 2018 by Taylor & Francis Group, LLC  
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed on acid-free paper  
Version Date: 20171024

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

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access [www.copyright.com](http://www.copyright.com) (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

**Trademark Notice:** Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

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

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

---

# Contents

---

**Preface** . . . . . ix

**List of Figures** . . . . . xiii

**List of Tables** . . . . . xv

**Contributors** . . . . . xix

**1 Empirical Research on Security and Privacy by Design** . . . . . 1

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

1.1 Introduction . . . . . 2

1.2 Empirical Research on Security and Privacy by Design . . . . . 5

1.3 Scoping . . . . . 9

1.4 Planning . . . . . 12

1.5 Operation . . . . . 30

1.6 Analysis and Interpretation . . . . . 35

1.7 Presentation and Packaging . . . . . 41

1.8 Conclusion . . . . . 43

**2 Guidelines for Systematic Mapping Studies in Security Engineering** . 47

*Michael Felderer and Jeffrey C. Carver*

2.1 Introduction . . . . . 48

2.2 Background on Systematic Mapping Studies in Software Engineering . . . . . 49

2.3 Overview of Available Mapping Studies in Security Engineering . . 55

2.4 Guidelines for Systematic Mapping Studies in Security Engineering 57

2.5 Summary . . . . . 65

<b>3</b>	<b>An Introduction to Data Analytics for Software Security . . . . .</b>	<b>69</b>
	<i>Lotfi ben Othmane, Achim D. Brucker, Stanislav Dashevskyi, and Peter Tsalovski</i>	
3.1	Introduction . . . . .	70
3.2	Secure Software Development . . . . .	71
3.3	Software Security Analytical Process . . . . .	74
3.4	Learning Methods Used in Software Security . . . . .	82
3.5	Evaluation of Model Performance . . . . .	86
3.6	More Lessons Learned . . . . .	89
3.7	Conclusion . . . . .	89
3.8	Acknowledgment . . . . .	90
<b>4</b>	<b>Generating Software Security Knowledge Through Empirical Methods</b>	<b>95</b>
	<i>René Noël, Santiago Matalonga, Gilberto Pedraza, Hernán Astudillo, and Eduardo B. Fernandez</i>	
4.1	Introduction and Motivation . . . . .	96
4.2	Empirical Methods for Knowledge Generation . . . . .	97
4.3	Example Application Domain: Secure Software Development Research Project . . . . .	99
4.4	Experiments . . . . .	100
4.5	Systematic Literature Mappings . . . . .	112
4.6	Case Studies . . . . .	122
4.7	Experimental Replications . . . . .	128
4.8	Conclusions . . . . .	132
4.9	Acknowledgment . . . . .	134
<b>5</b>	<b>Visual Analytics: Foundations and Experiences in Malware Analysis .</b>	<b>139</b>
	<i>Markus Wagner, Dominik Sacha, Alexander Rind, Fabian Fischer, Robert Luh, Sebastian Schrittwieser, Daniel A. Keim, and Wolfgang Aigner</i>	
5.1	Introduction . . . . .	140
5.2	Background in Malware Analysis . . . . .	140
5.3	Visual Analytics Foundations . . . . .	143
5.4	The Knowledge Generation Process . . . . .	150
5.5	Design and Evaluation for Visual Analytics Systems . . . . .	152
5.6	Experience in Malware Analysis . . . . .	154
5.7	Future Directions . . . . .	161
5.8	Conclusions . . . . .	164
<b>6</b>	<b>Analysis of Metrics for Classification Accuracy in Intrusion Detection</b>	<b>173</b>
	<i>Natalia Stakhanova and Alvaro A. Cardenas</i>	
6.1	Introduction . . . . .	174
6.2	Evaluation Metrics . . . . .	175
6.3	Literature Review . . . . .	185
6.4	What Hinders Adoption of Alternative Metrics . . . . .	191
6.5	Guidelines for Introducing New Evaluation Metrics . . . . .	194

6.6	Conclusions . . . . .	195
6.7	Acknowledgement . . . . .	196
<b>7</b>	<b>The Building Security in Maturity Model as a Research Tool . . . . .</b>	<b>201</b>
	<i>Martin Gilje Jaatun</i>	
7.1	Introduction . . . . .	201
7.2	Background . . . . .	202
7.3	Questionnaires in Software Security . . . . .	202
7.4	A Case Study . . . . .	204
7.5	Discussion . . . . .	205
7.6	Conclusion . . . . .	207
<b>8</b>	<b>Agile Test Automation for Web Applications — A Security Perspective</b>	<b>209</b>
	<i>Sandra Dominique Ringmann and Hanno Langweg</i>	
8.1	Introduction . . . . .	210
8.2	Methodology . . . . .	211
8.3	Risk Assessment . . . . .	212
8.4	Testing and Test Automation from the Security Perspective . . . . .	217
8.5	Static Analysis Tools . . . . .	222
8.6	Dynamic Analysis Tools and Frameworks . . . . .	229
8.7	Evaluating Static/Dynamic Analysis Tools and Frameworks . . . . .	238
8.8	Appraisal of the Tools . . . . .	239
8.9	Conclusion . . . . .	240
<b>9</b>	<b>Benchmark for Empirical Evaluation of Web Application Anomaly Detectors . . . . .</b>	<b>249</b>
	<i>Robert Bronte, Hossain Shahriar, and Hisham Haddad</i>	
9.1	Introduction . . . . .	250
9.2	Literature Review . . . . .	251
9.3	Benchmark Characteristics for Application-Layer Attack Detection Approaches . . . . .	256
9.4	An Example Environment for Generating Benchmark Data . . . . .	261
9.5	Using the Benchmark Dataset to Evaluate an IDS . . . . .	265
9.6	Conclusion . . . . .	271
<b>10</b>	<b>Threats to Validity in Empirical Software Security Research . . . . .</b>	<b>275</b>
	<i>Daniela S. Cruzes and Lotfi ben Othmane</i>	
10.1	Introduction . . . . .	276
10.2	Defining Validity . . . . .	277
10.3	Validity for Quantitative Research . . . . .	278
10.4	Threats to Validity for Qualitative Research . . . . .	289
10.5	Summary and Conclusions . . . . .	297
	<b>Index . . . . .</b>	<b>301</b>



# 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 dissemi-

nation. 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 Noél 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

---

- 1.1 The custom tool implemented for the SPAT2 study guides the participant through all the required steps. . . . . 28
  
- 2.1 Process Steps for Systematic Mapping Studies. . . . . 50
- 2.2 Classification Scheme for Model-Based Security Testing. . . . . 64
  
- 3.1 Overview of the S<sup>2</sup>DL . . . . . 71
- 3.2 Software security analytical process. . . . . 74
- 3.3 Plot that visualizes missing data. . . . . 78
- 3.4 Duration by vulnerability. . . . . 80
  
- 4.1 Running Example and Empirical Methods. . . . . 101
- 4.2 Box plot for effort and number of threats (QoT) variables. . . . . 109
- 4.3 Box plot for Experience groups . . . . . 110
- 4.4 Studies distribution plot . . . . . 120
- 4.5 Quantitative data analysis. . . . . 126
  
- 5.1 Components of a VA system: data, model, and visualization. . . . . 145
- 5.2 Sample images of the five visualization categories defined by Keim. 147
- 5.3 The knowledge generation model for visual analytics combines human and machine concepts. . . . . 150
- 5.4 Nested model for visualization design and evaluation by Munzner. . 153
- 5.5 MalwareVis [24] interactively visualizes network activity of an individual malware sample. . . . . 155
- 5.6 SEEM [22] : The system, which is built into Cynomix, provides a visual environment to support feature-based malware comparison for large attribute sets of malware samples. The highly interactive web-based system provides an overview about features of malware samples compared to other related or similar malware samples or families. . . . . 157

5.7	KAMAS [65]: a knowledge-assisted visualization system for behavior-based malware analysis. <i>Image by the authors.</i> . . . . .	159
5.8	A comparison of the three tools based on the knowledge generation model. Interactive components are illustrated with dashed borders. The strength of the loops (color and border) denote how well the loops are supported by the tools. <i>Image by the authors.</i> . . . . .	160
6.1	Authentication protocol V should output 1 if and only if P is who she claims to be. . . . .	175
6.2	(a) Evaluation of the correctness and security of an authentication algorithm. (b) Our problem: we can no longer achieve negligible error probabilities, i.e. $a$ cannot be made as small as possible without increasing $b$ . . . . .	176
6.3	B-ROC curve. . . . .	184
6.4	The summary of the reviewed publications. . . . .	187
6.5	The use of evaluation metrics over time. . . . .	188
6.6	The use of evaluation metrics among research groups and among the conferences. . . . .	190
7.1	Conservative maturity for the three most mature organizations . . .	205
8.1	OWASP Top 10 application security risks. . . . .	213
8.2	The cycle of continuous integration adapted from. . . . .	219
8.3	YASCA's architecture. . . . .	227
9.1	An Apache2 Attack Illustration . . . . .	257
9.2	User-to-Root Attack Diagram . . . . .	258
9.3	The Environment for Data Generation . . . . .	263
9.4	Content Management System Log Entries . . . . .	264
9.5	Blogging Platform Log Entries . . . . .	264
9.6	Bulletin Board System Log Entries . . . . .	264
9.7	Classifieds Marketplace Log Entries . . . . .	264
9.8	E-commerce Platform Log Entries . . . . .	264
9.9	Comparison between CEP, length and MD measures . . . . .	267
9.10	Comparison between CEV, length and MD measures . . . . .	267
9.11	Comparison between CET, length and MD measures . . . . .	267
9.12	Information-theoretic IDS Framework . . . . .	270
9.13	Log files created by web applications deployed in Apache and stored in MySQL . . . . .	271

---

# List of Tables

---

1.1	An illustration of typical discrepancies between beginning researchers, experienced researchers, and reviewers. . . . .	4
1.2	Empirical research for security by design: running examples . . . .	7
1.3	Empirical research for privacy by design: running examples . . . . .	8
2.1	Guidelines used for Systematic Mapping Studies in Security Engineering . . . . .	54
2.2	Security Engineering Mapping Studies . . . . .	55
2.3	Number of Retrieved and Included Papers in Available Mapping Studies in Security Engineering. . . . .	56
2.4	Main Security Journals Covering Software Security Engineering Topics. . . . .	58
2.5	Main Security-Related Conferences Covering Software Security Engineering Topics. . . . .	58
2.6	Main Software Engineering Venues Covered by Security Engineering Mapping Studies. . . . .	59
2.7	Common Databases Used for Software Security Engineering Mapping Studies . . . . .	60
3.1	Selected set of machine learning methods. . . . .	83
3.2	Confusion matrix for two-class outcome variables. . . . .	86
4.1	Levels of evidence, extended from Kitchenham et al. . . . .	97
4.2	Taxonomy of empirical methods . . . . .	98
4.3	Attainable evidence by method . . . . .	98
4.4	Scoping template adapted from . . . . .	102
4.5	Recommended descriptive statistics. . . . .	104
4.6	Scoping for the original experiment . . . . .	105
4.7	Experiment Design . . . . .	107
4.8	Experiment instruments . . . . .	107

4.9	Experiment Execution . . . . .	108
4.10	Medians for Experience groups . . . . .	110
4.11	Search string definition . . . . .	116
4.12	Source selection . . . . .	117
4.13	Inclusion/Exclusion Criteria . . . . .	118
4.14	Studies selection procedure . . . . .	119
4.15	Selected papers from first filtering . . . . .	120
4.16	Resume of number of papers selected in each phase . . . . .	120
4.17	Search string definition . . . . .	121
4.18	Qualitative Analysis: Constant Comparison Method application . . . . .	127
4.19	The family of experiments: Goal and Context . . . . .	133
6.1	<i>True positives (TP)</i> are known attack instances detected as abnormal, <i>false positives (FP)</i> are normal instances that are incorrectly classified as abnormal, <i>true negatives (TN)</i> are normal instances correctly identified as normal behavior, and <i>false negatives (FN)</i> present abnormal behavior incorrectly classified as normal. . . . .	177
6.2	Traditional metrics. . . . .	178
6.3	Summary of symbols. . . . .	181
6.4	Costs of IDS reports given a state of a system . . . . .	182
6.5	Costs model for detector outcome for a given event. $DCost(x) \leq RCost(x)$ is a degenerate case that will not apply in most cases. . . . .	182
6.6	The summary of the reviewed publications. . . . .	186
6.7	The evaluation metrics employed in the reviewed works. . . . .	186
6.8	Summary of alternative evaluation metrics. . . . .	191
7.1	The BSIMM Software Security Framework . . . . .	203
8.1	OWASP Top Ten 2013 . . . . .	214
8.2	Risk rating scores (adapted from. . . . .	215
8.3	Example risk calculation of A3-Cross-Site Scripting. . . . .	216
8.4	Likelihood and impact levels adapted from. . . . .	216
8.5	Overall risk severity. . . . .	217
8.6	Overview of static analysis tools referenced in. . . . .	223
8.7	Overview of dynamic analysis tools, i.e., vulnerability/web scanners and penetration testing tools. . . . .	229
8.8	Overview of dynamic analysis frameworks. . . . .	231
9.1	Summary of detailed Literature Review . . . . .	254
9.2	Measured Data Attributes from the Literature . . . . .	259
9.3	Related Works on Anomaly-based IDS . . . . .	268
10.1	Threats to conclusion validity in quantitative research. . . . .	280
10.2	Threats to internal validity in quantitative research. . . . .	281
10.3	Threats to construct validity in quantitative research. . . . .	282
10.4	Threats to external validity in quantitative research. . . . .	284

10.5	Validity criteria from different authors. . . . .	290
10.6	Techniques for addressing threats to validity in qualitative research.	292



# Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# 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 .....	2
1.2	Empirical Research on Security and Privacy by Design .....	5
1.3	Scoping .....	9
1.3.1	Setting Verifiable Goals .....	9
1.3.2	Process- or Result-Oriented .....	10
1.3.3	Quality or Quantity .....	11
1.4	Planning .....	12
1.4.1	Defining Research Questions .....	12
1.4.1.1	Determining appropriate measures .....	13
1.4.1.2	Defining success criteria .....	14

	1.4.1.3	Be prepared for surprises .....	15
1.4.2	Study Design .....		16
	1.4.2.1	Open or restricted environment .....	16
	1.4.2.2	Exploring the study potential .....	18
	1.4.2.3	Controlling variance .....	19
1.4.3	Students as Subjects .....		21
1.4.4	Instrumentation .....		23
	1.4.4.1	Study objects .....	23
	1.4.4.2	Guidelines and other material .....	25
	1.4.4.3	Measurement instruments .....	26
1.4.5	Evaluating Your Own Proposals .....		29
1.5	Operation .....		30
	1.5.1	Defining a Baseline .....	31
	1.5.2	Motivating and Training Participants .....	32
	1.5.3	Limited Time Frame .....	33
1.6	Analysis and Interpretation .....		35
	1.6.1	Preparing Data .....	35
		1.6.1.1 Anonymizing the dataset .....	35
		1.6.1.2 Data extraction .....	36
		1.6.1.3 Dealing with outliers .....	37
	1.6.2	Data Analysis .....	38
	1.6.3	Interpreting Results .....	39
1.7	Presentation and Packaging .....		41
	1.7.1	Replications .....	41
	1.7.2	Presentation .....	42
1.8	Conclusion .....		43
	References .....		44

## 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, . . .		
<i>a beginning researcher performs</i>	<b><i>an experienced researcher performs</i></b>	<i>a reviewer expects</i>
<ul style="list-style-type: none"> <li>▷ a small <i>validation exercise</i>,</li> <li>▷ involving a <i>handful of peers</i>,</li> <li>▷ who solve a <i>small example problem</i>,</li> <li>▷ after receiving a short, <i>ad-hoc introduction</i> to the approach,</li> <li>▷ where the approach introduces a specific <i>security or privacy design activity</i> technique,</li> <li>▷ resulting in measures that are determined only <i>after the data was collected and processed</i>.</li> </ul>	<ul style="list-style-type: none"> <li>▷ <b>a controlled experiment</b>,</li> <li>▷ <b>involving <math>N</math> representative participants</b>,</li> <li>▷ <b>who solve a well-scoped design exercise</b>,</li> <li>▷ <b>after receiving a structured tutorial about the approach</b>,</li> <li>▷ <b>where the approach supports a specific security or privacy design activity</b>,</li> <li>▷ <b>resulting in the quantitative evaluation of a specific, well-defined hypothesis concerning the approach</b>.</li> </ul>	<ul style="list-style-type: none"> <li>▷ several <i>replicated controlled experiments</i>,</li> <li>▷ involving at least 100* <i><math>N</math> experienced industrial developers, experts and practitioners</i>,</li> <li>▷ who perform a <i>large-scale industrial development project</i>,</li> <li>▷ after having <i>several months of practical experience</i> in applying the approach,</li> <li>▷ where the approach supports <i>the entire development life-cycle</i>,</li> <li>▷ resulting in the quantification of the influence of the approach on <i>security, privacy, productivity, compatibility with existing industrial practices, . . .</i></li> </ul>

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 led 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. Given 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 better 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
<b>Type of activity</b>			
Security Threat Modeling [27]	Security Threat Modeling [unpublished]	Secure architectural design with Patterns [38]	Secure architectural design with Patterns [37]
<b>Study goals</b>			
Productivity, correctness, completeness	Correctness, completeness, productivity	Performance and efficiency, impact of pattern catalog structure	Security (soundness and completeness of solutions), performance
<b>Number of participants</b>			
10 teams (41 master students)	93 participants (master students)	45 teams (90 master students)	32 teams (64 master students)
<b>Quantitative vs. qualitative</b>			
Quantitative	Quantitative, with exploration of ease of use	Quantitative (and some qualitative)	Quantitative (and some qualitative)
<b>Output</b>			
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)
<b>Environment</b>			
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
<b>Type of activity</b>			
Privacy threat Analysis at Requirements level [35]	Privacy threat Analysis at Architectural level [35]	Privacy methodology Comparison with privacy Experts [35]	Privacy Threat Modeling [un-published]
<b>Study goals</b>			
Correctness, completeness, productivity, ease of use	Correctness, completeness, productivity, ease of use	Reliability	Correctness, impact of domain knowledge
<b>Number of participants</b>			
3 teams (8 professionals)	27 teams (54 master students)	5 participants (2 methodology experts + 3 privacy experts)	122 participants (master students)
<b>Quantitative vs. qualitative</b>			
Qualitative, with exploration of quantitative goals	Quantitative, with exploration of ease of use	Quantitative	Quantitative, with exploration of ease of use
<b>Data</b>			
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
<b>Environment</b>			
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<sup>1</sup>. 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.

<sup>1</sup>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.